Cisco ASR 9000 Series Aggregation Services Router Multicast Configuration Guide, Release 5.1.x

First Published: 2013-09-01

Last Modified: 2014-04-01

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

CONTENTS

Preface

From Release 6.1.2 onwards, Cisco introduces support for the 64-bit Linux-based IOS XR operating system. Extensive feature parity is maintained between the 32-bit and 64-bit environments. Unless explicitly marked otherwise, the contents of this document are applicable for both the environments. For more details on Cisco IOS XR 64 bit, refer to the Release Notes for Cisco ASR 9000 Series Routers, Release 6.1.2 document.

The preface contains these sections:

Changes to This Document

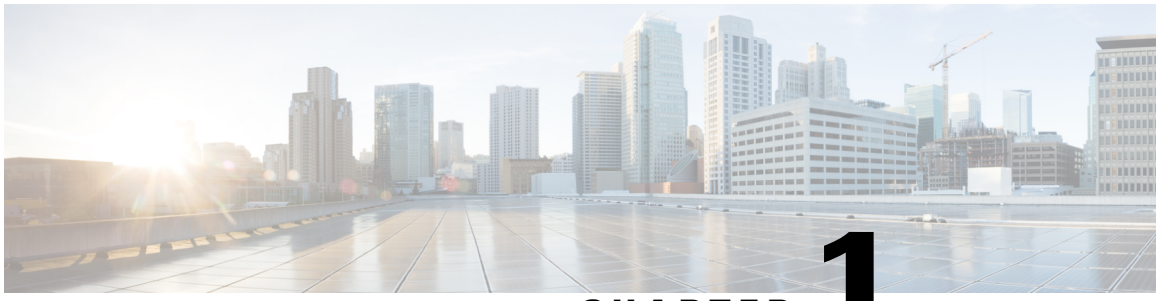
This table lists the technical changes made to this document since it was first printed.

Revision	Date	Summary
OL-30403-01	September 2013	Initial release of this document.
OL-30403-02	January 2014	Republished with documentation updates for Release 5.1.1
OL-30403-03	May 2014	Republished with documentation updates for Release 5.1.2

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see What's New in Cisco Product Documentation.

To receive new and revised Cisco technical content directly to your desktop, you can subscribe to the What's New in Cisco Product Documentation RSS feed. RSS feeds are a free service.



CHAPTER 1

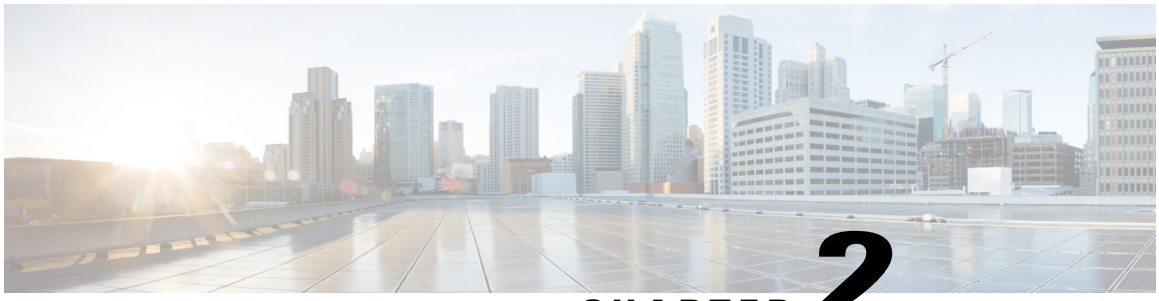
New and Changed Multicast Features

This chapter lists all the features that have been added or modified in this guide. The table also contains references to these feature documentation sections.

New and Changed Information Table

Table 1: New and Changed Features

Feature	Description	Changed in Release	Where Documented
Multicast IRB	This feature was introduced.	Release 5.1.1	Multicast IRB, on page 113
Multicast support for PWHE	This feature was introduced.	Release 5.1.1	Multicast support for PW-HE interfaces, on page 114
MVPN Profiles	This feature was updated with details for the new profiles. Support added for Ingress Replication P2MP-TE LSP.	Release 5.1.1	Characteristics of mLDP Profiles, on page 98
	No new features were added.	Release 5.1	-
MLDP Inband Signaling	This feature was introduced.	Release 5.1.2	MLDP inband signaling, on page 105
VRF support for MLD	This feature was introduced.	Release 5.1.2	VRF support for MLD, on page 120
P2MP MVPN enhancement	This feature was updated.	Release 5.1.2	Characteristics of mLDP Profiles, on page 98



CHAPTER 2

Implementing Layer-2 Multicast with IGMP Snooping

Internet Group Management Protocol (IGMP) snooping restricts multicast flows at Layer 2 to only those segments with at least one interested receiver. This module describes how to implement IGMP snooping on Cisco ASR 9000 Series Routers.

Feature History for IGMP Snooping

Release	Modification
Release 3.7.2	This feature was introduced.
Release 3.9.2	Support was added for the following features: • IGMP snooping group limits and access groups.
Release 4.0.0	Support was added for the following features: • Multicast redundancy using Multi-Chassis Link Aggregation (MC-LAG).

Prerequisites for IGMP Snooping

The following prerequisites must be satisfied before implementing IGMP snooping:

- The network must be configured with a Layer 2 VPN (L2VPN).
- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

  To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Restrictions for IGMP Snooping

- IGMP snooping is supported only under L2VPN bridge domains.
- Explicit host tracking (an IGMPv3 snooping feature) is not supported.
- IPv6 Multicast Listener Discovery (MLD) snooping is not supported.
- IGMPv1 is not supported.

Information About IGMP Snooping

IGMP Snooping Overview

Description of Basic Functions

IGMP snooping provides a way to constrain multicast traffic at Layer 2. By snooping the IGMP membership reports sent by hosts in the bridge domain, the IGMP snooping application can set up Layer 2 multicast forwarding tables to deliver traffic only to ports with at least one interested member, significantly reducing the volume of multicast traffic.

Configured at Layer 3, IGMP provides a means for hosts in an IPv4 multicast network to indicate which multicast traffic they are interested in and for routers to control and limit the flow of multicast traffic in the network at Layer 3.

IGMP snooping uses the information in IGMP membership report messages to build corresponding information in the forwarding tables to restrict IP multicast traffic at Layer 2. The forwarding table entries are in the form <Route, OIF List>, where:

- Route is a <*, G> route or <S, G> route.
- OIF List comprises all bridge ports that have sent IGMP membership reports for the specified route plus all multicast router (mrouter) ports in the bridge domain.

Implemented in a multicast network, IGMP snooping has the following attributes:

- In its basic form, it reduces bandwidth consumption by reducing multicast traffic that would otherwise flood an entire VPLS bridge domain.
- With the use of some optional configurations, it provides security between bridge domains by filtering the IGMP reports received from hosts on one bridge port and preventing leakage towards the hosts on other bridge ports.
- Using optional configurations, reduces the traffic impact on upstream IP multicast routers by suppressing IGMP membership reports (IGMPv2) or by acting as an IGMP proxy reporter (IGMPv3) to the upstream IP multicast router.

High Availability Features

All high availability features apply to the IGMP snooping processes with no additional configuration beyond enabling IGMP snooping. The following high availability features are supported:

- Process restarts
- RP Failover
- Stateful Switch-Over (SSO)
- Non-Stop Forwarding (NSF)—Forwarding continues unaffected while the control plane is restored following a process restart or route processor (RP) failover.
- Line card online insertion and removal (OIR)

Bridge Domain Support

IGMP snooping operates at the bridge domain level. When IGMP snooping is enabled on a bridge domain, the snooping functionality applies to all ports under the bridge domain, including:

- Physical ports under the bridge domain.
- Ethernet flow points (EFPs)—An EFP can be a VLAN, VLAN range, list of VLANs, or an entire interface port.
- Pseudowires (PWs) in VPLS bridge domains.
- Ethernet bundles—Ethernet bundles include IEEE 802.3ad link bundles and Cisco EtherChannel bundles. From the perspective of the IGMP snooping application, an Ethernet bundle is just another EFP. The forwarding application in the Cisco ASR 9000 Series Routers randomly nominates a single port from the bundle to carry the multicast traffic.

Multicast Router and Host Ports

IGMP snooping classifies each port (for example, EFPs, PWs, physical ports, or EFP bundles) as one of the following:

- Multicast router ports (mrouter ports)—These are ports to which a multicast-enabled router is connected. Mrouter ports are usually dynamically discovered, but may also be statically configured. Multicast traffic is always forwarded to all mrouter ports, except when an mrouter port is the ingress port.

- Host ports—Any port that is not an mrouter port is a host port.

Multicast Router Discovery and Static Configuration

IGMP snooping discovers mrouter ports dynamically. You can also explicitly configure a port as an mrouter port.

- Discovery—IGMP snooping identifies upstream mrouter ports in the bridge domain by snooping IGMP query messages and Protocol Independent Multicast Version 2 (PIMv2) hello messages. Snooping PIMv2 hello messages identifies IGMP nonqueriers in the bridge domain.
- Static configuration—You can statically configure a port as an mrouter port with the **mrouter** command in a profile attached to the port. Static configuration can help in situations when incompatibilities with non-Cisco equipment prevent dynamic discovery.

The **router-guard** command prevents a port from becoming a dynamically discovered mrouter port by filtering out multicast router messages, including IGMP queries and PIM messages. You can configure a port with the **router-guard** command and then configure it as a static mrouter. See the Router Guard and Static Mrouter, on page 17 for more information about configuring router-guard and mrouter commands on the same port.

Multicast Traffic Handling within a Bridge Domain with IGMP Snooping Enabled

The following tables describe traffic handling behaviors by IGMP snooping mrouter and host ports. Table 2: Multicast Traffic Handling for an IGMPv2 Querier, on page 6 describes traffic handling for an IGMPv2 querier. Table 3: Multicast Traffic Handling for an IGMPv3 Querier, on page 7 applies to an IGMPv3 querier.

By default, IGMP snooping supports IGMPv2 and IGMPv3. The version of the IGMP querier discovered in the bridge domain determines the operational version of the snooping processes. If you change the default, configuring IGMP snooping to support a minimum version of IGMPv3, IGMP snooping ignores any IGMPv2 queriers.

Table 2: Multicast Traffic Handling for an IGMPv2 Querier

Traffic Type	Received on MRouter Ports	Received on Host Ports
IP multicast source traffic	Forwards to all mrouter ports and to host ports that indicate interest.	Forwards to all mrouter ports and to host ports that indicate interest.
IGMP general queries	Forwards to all ports.	—
IGMP group-specific queries	Forwards to all other mrouter ports.	Dropped

Traffic Type	Received on MRouter Ports	Received on Host Ports
IGMPv2 joins	Examines (snoops) the reports. • If report suppression is enabled, forwards first join for a new group or first join following a general query for an existing group. • If report suppression is disabled, forwards on all mrouter ports.	Examines (snoops) the reports. • If report suppression is enabled, forwards first join for a new group or first join following a general query for an existing group. • If report suppression is disabled, forwards on all mrouter ports.
IGMPv3 reports	Ignores	Ignores
IGMPv2 leaves	Invokes last member query processing.	Invokes last member query processing.

Table 3: Multicast Traffic Handling for an IGMPv3 Querier

Traffic Type	Received on MRouter Ports	Received on Host Ports
IP multicast source traffic	Forwards to all mrouter ports and to host ports that indicate interest.	Forwards to all mrouter ports and to host ports that indicate interest.
IGMP general queries	Forwards to all ports.	—
IGMP group-specific queries	If received on the querier port floods on all ports.	—
IGMPv2 joins	Handles as IGMPv3 IS_EX{} reports.	Handles as IGMPv3 IS_EX{} reports.
IGMPv3 reports	• If proxy reporting is enabled—For state changes or source-list changes, generates a state change report on all mrouter ports. • If proxy reporting is disabled—Forwards on all mrouter ports.	• If proxy reporting is enabled—For state changes or source-list changes, generates a state change report on all mrouter ports. • If proxy reporting is disabled—Forwards on all mrouter ports.
IGMPv2 leaves	Handles as IGMPv3 IS_IN{} reports.	Handles as IGMPv3 IS_IN{} reports.

Multichassis Link Aggregation

The Multichassis Link Aggregation (MC-LAG) feature provides a simple redundancy mechanism for the Digital Subscriber Line Access Multiplexer (DSLAM) to access Cisco ASR 9000 Series Routers. The redundancy is achieved by allowing a dual-homed connection to two or more Cisco ASR 9000 Series Routers.

The DSLAM is known as a Dual-Homed Device (DHD) and the Cisco ASR 9000 Series Router is known as a Point of Attachment (PoA). An MC-LAG is assigned into a Redundancy Group (RG). The Cisco ASR 9000 Series Routers (PoAs) that manage a given MC-LAG are members of this RG. There may be multiple MC-LAGs in the RG. This indicates that the same RG may cover MC-LAG connections to other DSLAMs. Hence, the RG is uniquely identified on the PoAs by an Redundancy Group Identifier (RGID). The MC-LAG is identified on each PoA by a unique Redundancy Object Identifier, also termed as ROID. If VLAN sub-interfaces are configured on the MC-LAG, then each VLAN sub-interface has a unique ROID.

IGMP Snooping on the Cisco ASR 9000 Series Router supports MC-LAG configurations looking either downstream towards a DSLAM or upstream towards a multicast router.

Note Both the active and standby POAs must have the same configuration for the MC-LAG feature to work.

For more information on configuring link bundling and protocols used, see the Configuring Link Bundling chapter in Cisco ASR 9000 Series Aggregation Services Router Interface and Hardware Component Configuration Guide.

Information About IGMP Snooping Configuration Profiles

To enable IGMP snooping on a bridge domain, you must attach a profile to the bridge domain. The minimum configuration is an empty profile. An empty profile enables the default configuration options and settings for IGMP snooping, as listed in the Default IGMP Snooping Configuration Settings, on page 11.

You can attach IGMP snooping profiles to bridge domains or to ports under a bridge domain. The following guidelines explain the relationships between profiles attached to ports and bridge domains:

- Any IGMP profile attached to a bridge domain, even an empty profile, enables IGMP snooping. To disable IGMP snooping, detach the profile from the bridge domain.
- An empty profile configures IGMP snooping on the bridge domain and all ports under the bridge using default configuration settings.
- A bridge domain can have only one IGMP snooping profile attached to it (at the bridge domain level) at any time. Profiles can be attached to ports under the bridge, one profile per port.
- Port profiles are not in effect if the bridge domain does not have a profile attached to it.
- IGMP snooping must be enabled on the bridge domain for any port-specific configurations to be in effect.
- If a profile attached to a bridge domain contains port-specific configuration options, the values apply to all of the ports under the bridge, including all mrouter and host ports, unless another port-specific profile is attached to a port.
- When a profile is attached to a port, IGMP snooping reconfigures that port, disregarding any port configurations that may exist in the bridge-level profile.

Creating Profiles

To create a profile, use the **igmp snooping profile** command in global configuration mode.

Attaching and Detaching Profiles

To attach a profile to a bridge domain, use the **igmp snooping profile** command in l2vpn bridge group bridge domain configuration mode. To attach a profile to a port, use the **igmp snooping profile** command in the interface configuration mode under the bridge domain. To detach a profile, use the **no** form of the command in the appropriate configuration mode.

When you detach a profile from a bridge domain or a port, the profile still exists and is available for use at a later time. Detaching a profile has the following results:

- If you detach a profile from a bridge domain, IGMP snooping is deactivated in the bridge domain.
- If you detach a profile from a port, IGMP snooping configuration values for the port are instantiated from the bridge domain profile.

Changing Profiles

You cannot make changes to an active profile. An active profile is one that is currently attached.

If you need to change an active profile, you must detach it from all bridges or ports, change it, and reattach it.

Another way to do this is to create a new profile incorporating the desired changes and attach it to the bridges or ports, replacing the existing profile. This deactivates IGMP snooping and then reactivates it with parameters from the new profile.

Configuring Access Control

Access control configuration is the configuration of access groups and weighted group limits.

The role of access groups in IGMP v2/v3 message filteringis to permit or deny host membership requests for multicast groups (*,G) and multicast source groups (S,G). This is required to provide black-and-white list access to IPTV channel packages.

Weighted group limits restrict the number of IGMP v2/v3 groups, in which the maximum number of concurrently allowed multicast channels can be configured on a per EFP- and per PW-basis.

IGMP Snooping Access Groups

Although Layer-3 IGMP routing also uses the **igmp access-group** command in support of access groups, the support is not the same in Layer-2 IGMP, because the Layer-3 IGMP routing access group feature does not support source groups.

Access groups are specified using an extended IP access list referenced in an IGMP snooping profile that you attach to a bridge domain or a port.

Note A port-level access group overrides any bridge domain-level access group.

The **access-group** command instructs IGMP snooping to apply the specified access list filter to received membership reports. By default, no access list is applied.

Changes made to the access-list referenced in the profile (or a replacement of the access-list referenced in the igmp snooping profile) will immediately result in filtering the incoming igmp group reports and the existing group states accordingly, without the need for a detach-reattach of the igmp snooping profile in the bridge-domain, each time such a change is made.

IGMP Snooping Group Weighting

To limit the number of IGMP v2/v3 groups, in which the maximum number of concurrently allowed multicast channels must be configurable on a per EFP-basis and per PW-basis, configure group weighting.

IGMP snooping limits the membership on a bridge port to a configured maximum, but extends the feature to support IGMPv3 source groups and to allow different weights to be assigned to individual groups or source groups. This enables the IPTV provider, for example, to associate standard and high- definition IPTV streams, as appropriate, to specific subscribers.

This feature does not limit the actual multicast bandwidth that may be transmitted on a port. Rather, it limits the number of IGMP groups and source-groups, of which a port can be a member. It is the responsibility of the IPTV operator to configure subscriber membership requests to the appropriate multicast flows.

The **group policy** command, which is under igmp-snooping-profile configuration mode, instructs IGMP snooping to use the specified route policy to determine the weight contributed by a new <*,G> or <S,G> membership request. The default behavior is for there to be no group weight configured.

The **group limit** command specifies the group limit of the port. No new group or source group is accepted if its contributed weight would cause this limit to be exceeded. If a group limit is configured (without group policy configuration), a <S/*,G> group state will have a default weight of 1 attributed to it.

Note By default, each group or source-group contributes a weight of 1 towards the group limit. Different weights can be assigned to groups or source groups using the group policy command.

The group limit policy configuration is based on these conditions:

- Group weight values for <*,G> and <S,G> membership are configured in a Route Policy, that is included in an igmp snooping profile attached to a BD or port.
- Port level weight policy overrides any bridge domain level policy, if group-limit is set and route-policy is configured.
- If there is no policy configured, each group weight is counted equally and is equal to 1.
- If policy has been configured, all matching groups get weight of 1 and un-matched groups have 0 weight.

Default IGMP Snooping Configuration Settings

Table 4: IGMP Snooping Default Configuration Values

Scope	Feature	Default Value
Bridge Domain	IGMP snooping	Disabled on a bridge domain until an enabling IGMP profile is attached to the bridge domain.
	internal querier	None configured
	last-member-query-count	2
	last-member-query-interval	1000 (milliseconds)
	minimum-version	2 (supporting IGMPv2 and IGMPv3)
	querier query-interval	60 (seconds) **Note** This is a nonstandard default value.
	report-suppression	Enabled (enables report suppression for IGMPv2 and proxy-reporting for IGMPv3)
	querier robustness-variable	2
	router alert check	Enabled
	tcn query solicit	Disabled
	tcn flood	Enabled
	ttl-check	Enabled
	unsolicited-report-timer	1000 (milliseconds)
Port	immediate-leave	Disabled
	mrouter	No static mrouters configured; dynamic discovery occurs by default.
	router guard	Disabled
	static group	None configured

IGMP Snooping Configuration at the Bridge Domain Level

IGMP Minimum Version

The **minimum-version** command determines which IGMP versions are supported by IGMP snooping in the bridge domain:

- When minimum-version is 2, IGMP snooping intercepts IGMPv2 and IGMPv3 messages. This is the default value.
- When minimum-version is 3, IGMP snooping intercepts only IGMPv3 messages and drops all IGMPv2 messages.

IGMPv1 is not supported. The scope for this command is the bridge domain. The command is ignored in a profile attached to a port.

System IP Address

The **system-ip-address** command configures an IP address for IGMP snooping use. If not explicitly configured, the default address is 0.0.0.0. The default is adequate except in the following circumstances:

- If you are configuring an internal querier. The internal querier cannot use 0.0.0.0.
- If the bridge needs to communicate with an IGMP router that does not accept the 0.0.0.0 address.

The IGMP snooping system IP address is used in the following ways:

- The internal-querier sends queries from the system IP address. An address other than the default 0.0.0.0 must be configured.
- IGMPv3 sends proxy reports from the systemIP address. The default address 0.0.0.0 is preferred but may not be acceptable to some IGMP routers.
- In response to topology change notifications (TCNs) in the bridge domain, IGMP snooping sends global-leaves from the system IP address. The default address 0.0.0.0 is preferred but may not be acceptable to some IGMP routers.

Group Membership Interval, Robustness Variable, and Query Interval

The group membership interval (GMI) controls when IGMP snooping expires stale group membership states. The **show igmp snooping group** command shows groups with an expiry time of 0 until that stale state is cleaned up following the next query interval.

The GMI is calculated as:

GMI = (robustness-variable * query-interval) + maximum-response-time

where:

- maximum-response-time (MRT) is the amount of time during which receivers are required to report their membership state.
- robustness-variable is an integer used to influence the calculated GMI.

- query-interval is the amount of time between general queries.

Values for the components in the GMI are obtained as follows:

- MRT is advertised in the general query, for both IGMPv2 and IGMPv3.
- If the querier is running IGMPv2, IGMP snooping uses the IGMP-snooping-configured values for the robustness-variable and query-interval. These parameter values must match the configured values for the querier. In most cases, if you are interacting with other Cisco routers, you should not need to explicitly configure these values—the default values for IGMP snooping should match the default values of the querier. If they do not, use the **querier robustness-variable** and **querier query-interval** commands to configure matching values.
- IGMPv3 general queries convey values for robustness-variable and query-interval (QRV and QQI, respectively). IGMP snooping uses the values from the query, making the IGMP snooping GMI exactly match that of the querier.

Report Suppression (IGMPv2) and Proxy Reporting (IGMPv3)

The following IGMP snooping features reduce multicast traffic in a bridge domain. Both are enabled by default.

- IGMPv2 report suppression—If the bridge domain querier is running IGMPv2, IGMP snooping suppresses joins from a host if it has already forwarded the same join from another host during the current query interval. IGMP snooping forwards the last leave message to all mrouter ports.

  As insurance against lost reports when report suppression is enabled, IGMP snooping forwards IGMPv2 join reports the configured querier robustness-variable times for new groups. Configure the querier robustness-variable using the **querier robustness-variable** command.

- IGMPv3 proxy reporting—If the bridge domain querier is running IGMPv3, IGMP snooping acts as a proxy, generating reports from the proxy reporting address. Configure the proxy reporting address using the **system-ip-address** command. The default value is 0.0.0.0.

  As insurance against lost reports when proxy reporting is enabled, IGMP snooping generates and forwards state change reports robustness-variable times, where the robustness-variable is the QRV value in the querier's general query. The reports are forwarded at random intervals within the timeframe configured with the **unsolicited-report-timer** command.

To disable report suppression and proxy reporting, use the **report-suppression disable** command.

The scope for the commands mentioned in this section is the bridge domain. The commands are ignored in a profile attached to a port.

Group Leave Processing

Group Leave Options

When hosts want to leave a multicast group, they can either ignore the periodic general IGMP queries (called a silent leave), or they can send a group-specific leave message.

IGMP snooping can respond to group leaves in the following ways:

- Last member query processing—This is the default method for processing group leaves.

- Immediate leave—You can optionally configure individual ports for immediate leave.

Note IGMPv3 explicit host tracking, which provides per host immediate leave functionality on a multi-host LAN, is not supported.

Last Member Query Processing for IGMPv2 and IGMPv3

Last member query is the default group leave processing method used by IGMP snooping. With last member query processing, IGMP snooping processes leave messages as follows:

- IGMP snooping sends group-specific queries on the port that receives the leave message to determine if any other devices connected to that interface are interested in traffic for the specified multicast group. Using the following two configuration commands, you can control the latency between the request for a leave and the actual leave:
  - **last-member-query-count** command—Controls the number of group-specific queries IGMP snooping sends in response to a leave message.
  - **last-member-query-interval** command—Controls the amount of time between group-specific queries.
- If IGMP snooping does not receive an IGMP join message in response to group-specific queries, it assumes that no other devices connected to the port are interested in receiving traffic for this multicast group, and it removes the port from its Layer-2 forwarding table entry for that multicast group.
- If the leave message was from the only remaining port, IGMP snooping removes the group entry and generates an IGMP leave to the multicast routers.

Immediate-Leave Configuration

Immediate-leave is an optional port-level configuration parameter. Immediate-leave processing causes IGMP snooping to remove a Layer-2 interface from the forwarding table entry immediately, without first sending IGMP group-specific queries to the interface. After receiving an IGMP leave message, IGMP snooping immediately removes the interface from the Layer-2 forwarding table entry for that multicast group, unless a multicast router was learned on the port.

Immediate-leave processing improves leave latency, but is appropriate only when one receiver is configured on a port. For example, immediate-leave is appropriate in the following situations:

- Point-to-point configurations, such as an IPTV channel receiver
- Downstream DSLAMs with proxy reporting

Do not use immediate-leave on a port when the possibility exists for more than one receiver per port. Doing so could prevent an interested receiver from receiving traffic. For example, immediate-leave is not appropriate in a LAN.

Immediate-leave processing is a port-level option. You can configure this option explicitly per port in port profiles or in the bridge domain profile, in which case, it applies to all ports under the bridge.

Reaction to Topology Change Notifications

In a Spanning Tree Protocol (STP) topology, a Topology Change Notification (TCN) indicates that an STP topology change has occurred. As a result of a topology change, mrouters and hosts reporting group membership may migrate to other STP ports under the bridge domain. Mrouter and membership states must be relearned after a TCN.

IGMP snooping reacts to TCNs in the following ways:

1. IGMP snooping temporarily extends the flood set for all known multicast routes to include all ports participating in STP that are in forwarding state. The short-term flooding ensures that multicast delivery continues to all mrouters and all member hosts in the bridge domain while mrouter and membership states are relearned.

   But, as a result of this TCN flooding, downstream STP links may sometimes become over-subscribed by these extra multicast flows. This feature can in such cases be disabled by use of the **tcn flood disable** command.

2. The STP root bridge issues a global leave (for group 0.0.0.0) on all ports. This action triggers interoperable IGMP queriers to send general queries, expediting the relearning process.

Note Sending global leaves for query solicitation is a Cisco-specific implementation.

3. When the TCN refresh period ends, IGMP snooping withdraws the non-mrouter and non-member STP ports from the multicast route flood sets. You can control the amount of time that flooding occurs with the **tcn flood query count** command. This command sets the number of IGMP general queries for which the multicast traffic is flooded following a TCN, thereby influencing the refresh period.

IGMP snooping default behavior is that the STP root bridge always issues a global leave in response to a TCN, and that the non-root bridges do not issue global leaves.

With the **tcn query solicit** command, you can enable a bridge to always issue a global leave in response to TCNs, even when it is not the root bridge. In that case, the root bridge and the non-root bridge would issue the global leave and both would solicit general queries in response to a TCN. Use the **no** form of the command to turn off soliciting when the bridge is not the root.

Note One use for the **tcn query solicit** command is when Reverse Layer 2 Gateway Protocol (RL2GP) is configured to set up a MSTP Access Gateway. In this scenario, IGMP snooping is unaware of the root or non-root status of the bridge and, therefore, when a TCN occurs, no query is solicited in the domain unless IGMP snooping is explicity configured to do so on at least one bridge.

The root bridge always issues a global leave in response to a TCN. This behavior can not be disabled.

The internal querier has its own set of configuration options that control its reactions to TCNs.

The scope for all tcn related configuration option(s) is per bridge domain. If the command appears in profiles attached to ports, it has no effect.

IGMP Snooping Packet Checks

By default, IGMP snooping performs the following validations. If your network performs these validations elsewhere, you can disable the IGMP snooping validations.

- IGMP snooping checks the time-to-live (TTL) field in the IGMP header and drops packets where TTL is not equal to 1. The TTL field should always be set to 1 in the headers of IGMP reports and queries.

  You can disable this check using the **ttl-check disable** command, in which case IGMP snooping processes all packets without examining the TTL field in the IGMP header.

- IGMP snooping checks for the presence of the router alert option in the IP packet header of the IGMP message and drops packets that do not include this option.

  You can disable this check using the **router-alert-check disable** command, in which case IGMP snooping does not perform the validation before processing the message.

Startup Query Configuration

The startup query feature is configured using new igmp snooping profile parameters. You can configure the startup query processing in response to the following events:

- MC-LAG Port goes active
- Topology-change
- Port-up
- Process start

The above parameters are specific to MC-LAG feature. These are apart from the existing bridge domain level parameters such as count, MRT, and query interval. For more information about these CLI, refer the *Cisco ASR 9000 Series Aggregation Services Router Multicast Command Reference*.

Note

- For IGMP snooping to work on MC-LAG properly, the IGMP snooping configuration on both the POAs must be the same.
- In the case of downstream MC-LAG, when MC-LAG is configured and up and running, the MC-LAG port has to be added in IGMP Snooping enabled Bridge-domain.
- In the case of upstream MC-LAG, where POAs are attached to multicast router, the static mrouter port has to be configured on the multicast router that is towards both the POAs so that traffic is drawn to both the POAs.

IGMP Snooping Configuration at the Host Port Level

Router Guard and Static Mrouter

Router guard is a security feature that prevents malicious users from making a host port into an mrouter port. (This undesirable behavior is known as spoofing.) When a port is protected with the **router-guard** command, it cannot be dynamically discovered as an mrouter. When router guard is on a port, IGMP snooping filters protocol packets sent to the port and discards any that are multicast router control packets.

The **mrouter** command configures a port as a static mrouter.

You can use the **router-guard** and the **mrouter** commands on the same port to configure a guarded port as a static mrouter, for example, when:

- A large number of downstream host ports are present and you want to block dynamic mrouter discovery and configure static mrouters. In this case, configure the router guard feature at the domain level. By default, it will be applied to all ports, including the typically large number of downstream host ports. Then, use another profile without router guard configured for the relatively few upstream ports on which you want to permit dynamic mrouter discovery or configure static mrouters.
- Incompatibilities with non-Cisco equipment prevents correct dynamic discovery, you can disable all attempts for dynamic discovery using the router guard feature, and statically configure the mrouter.

  If you are using the router guard feature, because there is an incompatible IGMP router on the port, you should also configure the **mrouter** command on the port to ensure that the router receives IGMP reports and multicast flows.

Immediate-Leave

See the Group Leave Processing, on page 13.

Static Groups

IGMP snooping learns Layer-2 multicast groups dynamically. You can also statically configure Layer-2 multicast groups.

You can use the **static group** command in profiles intended for bridge domains or ports. I f you configure this option in a profile attached to a bridge domain, it applies to all ports under the bridge.

A profile can contain multiple static groups. You can define different source addresses for the same group address. Using the **source** keyword, you can configure IGMPv3 source groups.

Static group membership supersedes any dynamic manipulation by IGMP snooping. Multicast group membership lists can contain both static and dynamic group definitions.

When you configure a static group or source groups on a port, IGMP snooping adds the port as an outgoing port to the corresponding <S/*,G> forwarding entry and sends an IGMPv2 join or IGMPv3 report to all mrouter ports. IGMP snooping continues to send the membership report in response to general queries for as long as the static group remains configured on the port.

Internal Querier

When to Use an Internal Querier

In a network where IP multicast routing is configured, the IP multicast router acts as the IGMP querier. In situations when no external querier exists in the bridge domain (because the multicast traffic does not need to be routed), but local multicast sources exist, you must configure an internal querier to implement IGMP snooping. The internal querier solicits membership reports from hosts in the bridge domain so that IGMP snooping can build constrained multicast forwarding tables for the multicast traffic within the bridge domain.

An internal querier might also be useful when interoperability issues with non-Cisco equipment prevent IGMP snooping from working correctly with an external querier. In this case, you can:

1. Prevent the uncooperative external querier from being discovered by placing the **router-guard** command on that port.
2. Configure an internal querier to learn group membership interests from the ports in the bridge domain.
3. Configure static mrouter ports to receive multicast traffic.

Internal Querier Default Configuration

The minimum configuration for an internal querier is:

- Add the **internal-querier** command to a profile attached to the bridge domain. The default configuration is shown in Table 5: Internal Querier Default Configuration Values, on page 18.
- Add the **system-ip-address** command to a profile attached to the bridge domain to configure an address other than the default 0.0.0.0.

Table 5: Internal Querier Default Configuration Values

Configuration Command	Default Value
system-ip-address	0.0.0.0. The default address is invalid for the internal-querier.
internal-querier *max-response-time*	10
internal-querier *query-interval*	60 (seconds) **Note** This is a nonstandard default value.
internal-querier *robustness-variable*	2
internal-querier tcn query *count*	2
internal-querier tcn query *interval*	10 (seconds)

Configuration Command	Default Value
internal-querier timer *expiry* **Note** This is the Other Querier Present Interval as defined in RFC-3376, Section 8.5.	125 (seconds): robustness-variable * query-interval + ½(max-response-time) For example, using the default values for all components: (2 * 60) + ½ (10) = 125
internal-querier *version*	3

You can disable the internal querier (using the **no** form of the **internal-querier** command) without removing any other internal querier commands. The additional internal querier commands are ignored in that case.

The scope for the **internal-querier** command is per bridge domain. If the command appears in profiles attached to ports, it has no effect.

Internal Querier Processing

When the internal querier is the elected querier in the domain, it solicits membership reports by sending IGMP general queries at the interval specified by the **internal-querier query-interval** command on every active port in the bridge domain. The internal querier sends IGMPv3 queries by default. You can configure it to send IGMPv2 messages instead using the **internal-querier version** command.

The local IGMP snooping process responds to the internal querier's general queries. In particular, the IGMPv3 proxy (if enabled) generates a current-state report and forwards it to all mrouters. For IGMPv2 or when the IGMPv3 proxy is disabled, IGMP snooping generates current-state reports for static group state only.

The queries are sent from the address you configure for IGMP snooping using the **system-ip-address** command. The queries include the maximum response time configured with the **internal-querier max-response-time** command.

The **internal-querier robustness-variable** and **internal-querier query-interval** commands configure values for both IGMPv2 and IGMPv3 processing.

Querier Election for One Active Querier

A bridge domain can have only one active querier at a time. If the internal-querier receives queries from another querier in a bridge domain, it performs querier election. The lowest IP address wins. If the internal querier is the election loser, IGMP snooping starts a timer with the value set by the **internal-querier timer expiry** command. If this timer expires before another query is received from the election winner, the internal querier becomes the active querier.

Note The default **internal-querier timer expiry** command value is derived from the values of other configuration options, as described in Table 5: Internal Querier Default Configuration Values, on page 18. You can configure a different value to override the default calculation.

Internal Querier Reaction to TCNs

IGMP snooping generates group leaves in response to topology change notifications. For more information about how IGMP snooping reacts to TCNs, see the Reaction to Topology Change Notifications, on page 15.

If the internal querier receives a group leave while it is the elected querier in the domain, it reacts as follows:

- Generates an IGMP general query immediately.
- Waits the amount of time set by the **internal-querier tcn query interval** command and generates another IGMP general query.
- Continues to wait for the specified interval time and to send general queries until the query count reaches the value set with the **internal querier tcn query count** command.

Note You can configure the internal querier to ignore global leaves by setting the internal querier TCN query count to 0.

How to Configure IGMP Snooping

The first two tasks are required to configure basic IGMP snooping configuration. The optional tasks configure additional IGMP snooping features and provide a way to view statistics and counters:

- Adding Static Mrouter Configuration to a Profile, on page 26 (optional)
- Adding Router Guard to a Profile, on page 27 (optional)
- Configuring Immediate-Leave, on page 29 (optional)
- Configuring Static Groups, on page 30 (optional)
- Configuring an Internal Querier, on page 31 (optional)
- Verifying Multicast Forwarding, on page 32 (optional)

Creating an IGMP Snooping Profile

SUMMARY STEPS

1. **configure**
2. **igmp snooping profile** *profile-name*
3. Optionally, add commands to override default configuration values.
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**igmp snooping profile** *profile-name* **Example:** `RP/0/RSP0/CPU0:router(config)# igmp snooping profile default-bd-profile`	Enters IGMP snooping profile configuration mode and creates a named profile. The default profile enables IGMP snooping. You can commit the new profile without any additional configurations, or you can include additional configuration options to the profile. You can also return to the profile later to add configurations, as described in other tasks in this module.
Step 3	Optionally, add commands to override default configuration values.	If you are creating a bridge domain profile, consider the following: • An empty profile is appropriate for attaching to a bridge domain. An empty profile enables IGMP snooping with default configuration values. • You can optionally add more commands to the profile to override default configuration values. • If you include port-specific configurations in a bridge domain profile, the configurations apply to all ports under the bridge, unless another profile is attached to a port. If you are creating a port-specific profile, consider the following: • While an empty profile could be attached to a port, it would have no effect on the port configuration. • When you attach a profile to a port, IGMP snooping reconfigures that port, overriding any inheritance of configuration values from the bridge-domain profile. You must repeat the commands in the port profile if you want to retain those configurations. You can detach a profile, change it, and reattach it to add commands to a profile at a later time.
Step 4	**commit**	

Where to Go Next

You must attach a profile to a bridge domain or to a port to have it take effect. See one of the following tasks:

Attaching a Profile and Activating IGMP Snooping on a Bridge Domain

To activate IGMP snooping on a bridge domain, attach an IGMP snooping profile to the bridge domain, as described in the following steps.

SUMMARY STEPS

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *bridge-domain-name*
5. **igmp snooping profile** *profile-name*
6. **commit**
7. **show igmp snooping bridge-domain detail**
8. **show l2vpn bridge-domain detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**l2vpn** **Example:** `RP/0/RSP0/CPU0:router(config)# l2vpn`	Enters Layer 2 VPN configuration mode.
Step 3	**bridge group** *bridge-group-name* **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group GRP1`	Enters Layer 2 VPN VPLS bridge group configuration mode for the named bridge group.
Step 4	**bridge-domain** *bridge-domain-name* **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain ISP1`	Enters Layer 2 VPN VPLS bridge group bridge domain configuration mode for the named bridge domain.
Step 5	**igmp snooping profile** *profile-name* **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# igmp snooping profile default-bd-profile`	Attaches the named IGMP snooping profile to the bridge domain, enabling IGMP snooping on the bridge domain.
Step 6	**commit**	
Step 7	**show igmp snooping bridge-domain detail** **Example:** `RP/0/RSP0/CPU0:router# show igmp snooping`	(Optional) Verifies that IGMP snooping is enabled on a bridge domain and shows the IGMP snooping profile names attached to bridge domains and ports.

	Command or Action	Purpose
	`bridge-domain detail`	
Step 8	**show l2vpn bridge-domain detail** **Example:** `RP/0/RSP0/CPU0:router# show l2vpn bridge-domain`	(Optional) Verifies that IGMP snooping is implemented in the forwarding plane (Layer 2) on a bridge domain.

Detaching a Profile and Deactivating IGMP Snooping on a Bridge Domain

To deactivate IGMP snooping on a bridge domain, remove the profile from the bridge domain using the following steps.

Note A bridge domain can have only one profile attached to it at a time.

SUMMARY STEPS

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *bridge-domain-name*
5. **no igmp snooping**
6. **commit**
7. **show igmp snooping bridge-domain detail**
8. **show l2vpn bridge-domain detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**l2vpn** **Example:** `RP/0/RSP0/CPU0:router(config)# l2vpn`	Enters Layer 2 VPN configuration mode.

	Command or Action	Purpose
Step 3	**bridge group** *bridge-group-name* **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group GRP1`	Enters Layer 2 VPN VPLS bridge group configuration mode for the named bridge group.
Step 4	**bridge-domain** *bridge-domain-name* **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain ISP1`	Enters Layer 2 VPN VPLS bridge group bridge domain configuration mode for the named bridge domain.
Step 5	**no igmp snooping** **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# no igmp snooping`	Detaches the IGMP snooping profile from the bridge domain, disabling IGMP snooping on that bridge domain. **Note** Only one profile can be attached to a bridge domain at a time. If a profile is attached, IGMP snooping is enabled. If a profile is not attached, IGMP snooping is disabled.
Step 6	**commit**	
Step 7	**show igmp snooping bridge-domain detail** **Example:** `RP/0/RSP0/CPU0:router# show igmp snooping bridge-domain detail`	(Optional) Verifies that IGMP snooping is disabled on a bridge domain.
Step 8	**show l2vpn bridge-domain detail** **Example:** `RP/0/RSP0/CPU0:router# show l2vpn bridge-domain`	(Optional) Verifies that IGMP snooping is disabled in the forwarding plane (Layer 2) on a bridge domain.

Attaching and Detaching Profiles to Ports Under a Bridge

Before You Begin

IGMP snooping must be enabled on the bridge domain for port-specific profiles to affect IGMP snooping behavior.

SUMMARY STEPS

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *bridge-domain-name*
5. **interface** *interface-type interface-number*
6. Do one of the following:
   - **igmp snooping profile** *profile-name*
   - **no igmp snooping**
7. **commit**
8. **show igmp snooping bridge-domain detail**
9. **show l2vpn bridge-domain detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**l2vpn** **Example:** `RP/0/RSP0/CPU0:router(config)# l2vpn`	Enters Layer 2 VPN configuration mode.
Step 3	**bridge group** *bridge-group-name* **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group GRP1`	Enters Layer 2 VPN bridge group configuration mode for the named bridge group.
Step 4	**bridge-domain** *bridge-domain-name* **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain ISP1`	Enters Layer 2 VPN bridge group bridge domain configuration mode for the named bridge domain.
Step 5	**interface** *interface-type interface-number* **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface gig 1/1/1/1`	Enters Layer 2 VPN VPLS bridge group bridge domain interface configuration mode for the named interface or PW.
Step 6	Do one of the following:	Attaches the named IGMP snooping profile to the port.

	Command or Action	Purpose
	• **igmp snooping profile** *profile-name* • **no igmp snooping** **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-if)# igmp snooping profile mrouter-port-profile`	**Note** A profile on a port has no effect unless there is also a profile attached to the bridge. The **no** form of the command detaches a profile from the port. Only one profile can be attached to a port.
Step 7	**commit**	
Step 8	**show igmp snooping bridge-domain detail** **Example:** `RP/0/RSP0/CPU0:router# show igmp snooping bridge-domain detail`	(Optional) Verifies that IGMP snooping is enabled on a bridge domain and shows the IGMP snooping profile names attached to bridge domains and ports.
Step 9	**show l2vpn bridge-domain detail** **Example:** `RP/0/RSP0/CPU0:router# show l2vpn bridge-domain`	(Optional) Verifies that IGMP snooping is implemented in the forwarding plane (Layer 2) on a bridge domain.

Adding Static Mrouter Configuration to a Profile

Before You Begin

IGMP snooping must be enabled on the bridge domain for port-specific profiles to affect IGMP snooping behavior.

Note Static mrouter port configuration is a port-level option and should be added to profiles intended for ports. It is not recommended to add mrouter port configuration to a profile intended for bridge domains.

SUMMARY STEPS

1. **configure**
2. **igmp snooping profile** *profile-name*
3. **mrouter**
4. **commit**
5. **show igmp snooping profile** *profile-name* **detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**igmp snooping profile** *profile-name* **Example:** `RP/0/RSP0/CPU0:router(config)# igmp snooping profile mrouter-port-profile`	Enters IGMP snooping profile configuration mode and creates a new profile or accesses an existing profile.
Step 3	**mrouter** **Example:** `RP/0/RSP0/CPU0:router(config-igmp-snooping-profile)# mrouter`	Configures a port as a static mrouter port.
Step 4	**commit**	
Step 5	**show igmp snooping profile** *profile-name* **detail** **Example:** `RP/0/RSP0/CPU0:router# show igmp snooping profile mrouter-port-profile detail`	(Optional) Displays the configuration settings in the named profile.

Where to Go Next

Attach the profile to ports to complete static mrouter configuration. See the Attaching and Detaching Profiles to Ports Under a Bridge, on page 24.

Adding Router Guard to a Profile

To prevent multicast routing protocol messages from being received on a port and, therefore, prevent a port from being a dynamic mrouter port, follow these steps. Note that both router guard and static mrouter commands may be configured on the same port. See the Router Guard and Static Mrouter, on page 17 for information.

Before You Begin

IGMP snooping must be enabled on the bridge domain for port-specific profiles to affect IGMP snooping behavior.

Note Router guard configuration is a port-level option and should be added to profiles intended for ports. It is not recommended to add router guard configuration to a profile intended for bridge domains. To do so would prevent all mrouters, including IGMP queriers, from being discovered in the bridge domain.

SUMMARY STEPS

1. **configure**
2. **igmp snooping profile** *profile-name*
3. **router-guard**
4. **commit**
5. **show igmp snooping profile** *profile-name* **detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**igmp snooping profile** *profile-name* **Example:** `RP/0/RSP0/CPU0:router(config)# igmp snooping profile host-port-profile`	Enters IGMP snooping profile configuration mode and creates a new profile or accesses an existing profile.
Step 3	**router-guard** **Example:** `RP/0/RSP0/CPU0:router(config-igmp-snooping-profile)# router-guard`	Protects the port from dynamic discovery.
Step 4	**commit**	
Step 5	**show igmp snooping profile** *profile-name* **detail** **Example:** `RP/0/RSP0/CPU0:router# show igmp snooping profile host-port-profile detail`	(Optional) Displays the configuration settings in the named profile.

Where to Go Next

Attach the profile to ports to complete router guard configuration. See the Attaching and Detaching Profiles to Ports Under a Bridge, on page 24.

Configuring Immediate-Leave

To add the IGMP snooping immediate-leave option to an IGMP snooping profile, follow these steps.

Before You Begin

IGMP snooping must be enabled on the bridge domain for port-specific profiles to affect IGMP snooping behavior.

SUMMARY STEPS

1. **configure**
2. **igmp snooping profile** *profile-name*
3. **immediate-leave**
4. **commit**
5. **show igmp snooping profile** *profile-name* **detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**igmp snooping profile** *profile-name* **Example:** `RP/0/RSP0/CPU0:router(config)# igmp snooping profile host-port-profile`	Enters IGMP snooping profile configuration mode and creates a new profile or accesses an existing profile.
Step 3	**immediate-leave** **Example:** `RP/0/RSP0/CPU0:router(config-igmp-snooping-profile)# immediate-leave`	Enables the immediate-leave option. • If you add this option to a profile attached to a bridge domain, it applies to all ports under the bridge. • If you add this option to a profile attached to a port, it applies to the port.
Step 4	**commit**	
Step 5	**show igmp snooping profile** *profile-name* **detail** **Example:** `RP/0/RSP0/CPU0:router# show igmp snooping profile host-port-profile detail`	(Optional) Displays the configuration settings in the named profile.

Where to Go Next

Attach the profile to bridge domains or ports to complete immediate-leave configuration. See one of the following sections:

Configuring Static Groups

To add one or more static groups or IGMPv3 source groups to an IGMP snooping profile, follow these steps.

Before You Begin

IGMP snooping must be enabled on the bridge domain for port-specific profiles to affect IGMP snooping behavior.

SUMMARY STEPS

1. **configure**
2. **igmp snooping profile** *profile-name*
3. **static-group** *group-addr* [**source** *source-addr*]
4. Repeat the previous step, as needed, to add more static groups.
5. **commit**
6. **show igmp snooping profile** *profile-name* **detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**igmp snooping profile** *profile-name* **Example:** `RP/0/RSP0/CPU0:router(config)# igmp snooping profile host-port-profile`	Enters IGMP snooping profile configuration mode and creates a new profile or accesses an existing profile.
Step 3	**static-group** *group-addr* [**source** *source-addr*] **Example:** `RP/0/RSP0/CPU0:router(config-igmp-snooping-profile)# static-group 239.1.1.1 source 10.0.1.1`	Configures a static group. • If you add this option to a profile attached to a bridge domain, it applies to all ports under the bridge. • If you add this option to a profile attached to a port, it applies to the port.
Step 4	Repeat the previous step, as needed, to add more static groups.	(Optional) Adds additional static groups.
Step 5	**commit**	

	Command or Action	Purpose
Step 6	**show igmp snooping profile** *profile-name* **detail** **Example:** `RP/0/RSP0/CPU0:router# show igmp snooping profile host-port-profile detail`	(Optional) Displays the configuration settings in the named profile.

Where to Go Next

Attach the profile to bridge domains or ports to complete static-group configuration. See one of the following sections:

Configuring an Internal Querier

Before You Begin

IGMP snooping must be enabled on the bridge domain for this procedure to take effect.

SUMMARY STEPS

1. **configure**
2. **igmp snooping profile** *profile-name*
3. **system-ip-address** *ip-addr*
4. **internal-querier**
5. **commit**
6. **show igmp snooping profile** *profile-name* **detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**igmp snooping profile** *profile-name* **Example:** `RP/0/RSP0/CPU0:router(config)# igmp snooping profile internal-querier-profile`	Enters IGMP snooping profile configuration mode and creates a new profile or accesses an existing profile.

	Command or Action	Purpose
Step 3	**system-ip-address** *ip-addr* **Example:** `RP/0/RSP0/CPU0:router(config-igmp-snooping-profile)# system-ip-address 10.1.1.1`	Configures an IP address for internal querier use. The default system-ip-address value (0.0.0.0) is not valid for the internal querier. You must explicitly configure an IP address.
Step 4	**internal-querier** **Example:** `RP/0/RSP0/CPU0:router(config-igmp-snooping-profile)# internal-querier`	Enables an internal querier with default values for all options.
Step 5	**commit**	
Step 6	**show igmp snooping profile** *profile-name* **detail** **Example:** `RP/0/RSP0/CPU0:router# show igmp snooping profile internal-querier-profile detail`	(Optional) Displays the configuration settings in the named profile.

Where to Go Next

Attach the profile to a bridge domain to complete internal querier configuration.

See Attaching a Profile and Activating IGMP Snooping on a Bridge Domain, on page 21.

Verifying Multicast Forwarding

SUMMARY STEPS

1. **configure**
2. **show l2vpn forwarding bridge-domain** [*bridge-group-name*:*bridge-domain-name*] **mroute ipv4** [**detail**] [**hardware** {**ingress** | **egress**}] **location** *node-id*
3. **show l2vpn forwarding bridge-domain** [*bridge-group-name*:*bridge-domain-name*] **mroute ipv4 summary location** *node-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

	Command or Action	Purpose
Step 2	**show l2vpn forwarding bridge-domain** [*bridge-group-name*:*bridge-domain-name*] **mroute ipv4** [**detail**] [**hardware** {**ingress** | **egress**}] **location** *node-id* **Example:** `RP/0/RSP0/CPU0:router# show l2vpn forwarding bridge-domain bridgeGroup1:ABC mroute ipv4 detail location 0/3/CPU0`	Displays multicast routes as they are converted into the forwarding plane forwarding tables. Use optional arguments to limit the display to specific bridge groups or bridge domains. If these routes are not as expected, check the control plane configuration and correct the corresponding IGMP snooping profiles.
Step 3	**show l2vpn forwarding bridge-domain** [*bridge-group-name*:*bridge-domain-name*] **mroute ipv4 summary location** *node-id* **Example:** `RP/0/RSP0/CPU0:router# show l2vpn forwarding bridge-domain bridgeGroup1:ABC mroute ipv4 summary location 0/3/CPU0`	Displays summary-level information about multicast routes as stored in the forwarding plane forwarding tables. Use optional arguments to limit the display to specific bridge domains.

Configuring Group Limits

This procedure consists the following tasks:

Configuring route-policy

SUMMARY STEPS

1. **configure**
2. **route-policy** *policy-name*
3. **end-policy**
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

	Command or Action	Purpose
Step 2	**route-policy** *policy-name* **Example:** `RP/0/RSP0/CPU0:router(config)# route-policy sky`	Configures route policy with the defined name.
Step 3	**end-policy** **Example:** `RP/0/RSP0/CPU0:router(config-rpl)# end-policy`	Ends the route-policy configuration.
Step 4	**commit**	

Configuring group limit

SUMMARY STEPS

1. **configure**
2. **igmp snooping profile** *profile-name*
3. **group policy** *policy-name*
4. **group limit** *range*
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**igmp snooping profile** *profile-name* **Example:** `RP/0/RSP0/CPU0:router(config)# igmp snooping profile name1`	Enters IGMP snooping profile configuration mode and creates a new profile or accesses an existing profile.
Step 3	**group policy** *policy-name* **Example:** `RP/0/RSP0/CPU0:router(config-igmp-snooping-profile)# group policy policy1`	Specifies the configured route-policy to set the group weight.

	Command or Action	Purpose
Step 4	**group limit** *range* **Example:** `RP/0/RSP0/CPU0:router(config-igmp-snooping-profile)# group limit 100`	Limits the number of groups (or source-groups) allowed on a port.
Step 5	**commit**	

Configuring access-groups

This task instructs IGMP Snoop to apply the specified access-list filter(s) to receive membership reports.

The user needs to create and configure access-lists before configuring the access-groups. For detailed configuration procedures, for creating and configuring standard and extended access-lists, refer to the Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Configuration Guide.

SUMMARY STEPS

1. **configure**
2. **igmp snooping profile** *profile-name*
3. **access-group** *acl-name*
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**igmp snooping profile** *profile-name* **Example:** `RP/0/RSP0/CPU0:router(config)# igmp snooping profile name1`	Enters IGMP snooping profile configuration mode and creates a new profile or accesses an existing profile.
Step 3	**access-group** *acl-name* **Example:** `RP/0/RSP0/CPU0:router(config-igmp-snooping-profile)# access-group acl1`	Configures group membership filter(s).
Step 4	**commit**	

Configuration Examples for IGMP Snooping

The following examples show how to enable IGMP snooping on Layer 2 VPLS bridge domains on Cisco ASR 9000 Series Routers:

Configuring IGMP Snooping on Physical Interfaces Under a Bridge: Example

1. Create two profiles.

```
igmp snooping profile bridge_profile
!
igmp snooping profile port_profile
   mrouter
!
```

2. Configure two physical interfaces for L2 transport.

```
interface GigabitEthernet0/8/0/38
   negotiation auto
   l2transport
   no shut
   !
!
interface GigabitEthernet0/8/0/39
   negotiation auto
   l2transport
   no shut
   !
!
```

3. Add interfaces to the bridge domain. Attach bridge_profile to the bridge domain and port_profile to one of the Ethernet interfaces. The second Ethernet interface inherits IGMP snooping configuration attributes from the bridge domain profile.

```
l2vpn
  bridge group bg1
     bridge-domain bd1
     igmp snooping profile bridge_profile
     interface GigabitEthernet0/8/0/38
       igmp snooping profile port_profile
     interface GigabitEthernet0/8/0/39
     !
  !
!
```

4. Verify the configured bridge ports.

```
show igmp snooping port
```

Configuring IGMP Snooping on VLAN Interfaces Under a Bridge: Example

1 Configure two profiles.

```
igmp snooping profile bridge_profile
igmp snooping profile port_profile
   mrouter
!
```

2 Configure VLAN interfaces for L2 transport.

```
interface GigabitEthernet0/8/0/8
   negotiation auto
   no shut
   !
!
interface GigabitEthernet0/8/0/8.1 l2transport
   encapsulation dot1q 1001
   mtu 1514
   !
!
interface GigabitEthernet0/8/0/8.2 l2transport
   encapsulation dot1q 1002
   mtu 1514
   !
!
```

3 Attach a profile and add interfaces to the bridge domain. Attach a profile to one of the interfaces. The other interface inherits IGMP snooping configuration attributes from the bridge domain profile.

```
l2vpn
  bridge group bg1
     bridge-domain bd1
     igmp snooping profile bridge_profile
     interface GigabitEthernet0/8/0/8.1
       igmp snooping profile port_profile
     interface GigabitEthernet0/8/0/8.2
     !
  !
!
```

4 Verify the configured bridge ports.

```
show igmp snooping port
```

Configuring IGMP Snooping on Ethernet Bundles Under a Bridge: Example

1 This example assumes that the front-ends of the bundles are preconfigured. For example, a bundle configuration might consist of three switch interfaces, as follows:

```
    interface Port-channel1
    !
interface GigabitEthernet0/0/0/0
    !
interface GigabitEthernet0/0/0/1
!
    interface GigabitEthernet0/0/0/2
       channel-group 1 mode on
    !
    interface GigabitEthernet0/0/0/3
```

```
      channel-group 1 mode on
   !
```

2 Configure two IGMP snooping profiles.

```
igmp snooping profile bridge_profile
!
igmp snooping profile port_profile
   mrouter
!
```

3 Configure interfaces as bundle member links.

```
interface GigabitEthernet0/0/0/0
  bundle id 1 mode on
  negotiation auto
!
interface GigabitEthernet0/0/0/1
  bundle id 1 mode on
  negotiation auto
!
interface GigabitEthernet0/0/0/2
  bundle id 2 mode on
  negotiation auto
!
interface GigabitEthernet0/0/0/3
  bundle id 2 mode on
  negotiation auto
!
```

4 Configure the bundle interfaces for L2 transport.

```
interface Bundle-Ether 1
          l2transport
          !
       !
       interface Bundle-Ether 2
          l2transport
          !
       !
```

5 Add the interfaces to the bridge domain and attach IGMP snooping profiles.

```
l2vpn
  bridge group bg1
     bridge-domain bd1
     igmp snooping profile bridge_profile
     interface bundle-Ether 1
       igmp snooping profile port_profile
     interface bundle-Ether 2
     !
   !
 !
```

6 Verify the configured bridge ports.

```
show igmp snooping port
```

Configuring IGMP Snooping on VFIs Under a Bridge: Example

This example configures IGMP snooping on a virtual forwarding instance (VFI) under a bridge domain. The topology consists of two routers, PE1 and PE2, each with an access circuit (AC) and pseudowire (PW) as bridge ports.

PE1 Configuration

1 Configure IGMP snooping profiles.

```
igmp snooping profile prof1
!
igmp snooping profile prof2
   mrouter
!
```

2 Configure interfaces.

```
interface Loopback0
  ipv4 address 10.1.1.1 255.255.255.255
!
interface GigabitEthernet0/2/0/9
   ipv4 address 10.10.10.1 255.255.255.0
   negotiation auto
!
interface GigabitEthernet0/2/0/39
   negotiation auto
   l2transport
!
```

3 Configure Open Shortest Path First (OSPF).

```
router ospf 1
  log adjacency changes
  router-id 10.1.1.1
  area 0
    interface Loopback0
    !
    interface GigabitEthernet0/2/0/9
    !
  !
!
```

4 Configure Label Distribution Protocol (LDP).

```
mpls ldp
   router-id 10.1.1.1
   log neighbor
   !
   interface GigabitEthernet0/2/0/9
   !
!
```

5 Configure a bridge domain, enable IGMP snooping on the bridge, and add the interfaces to the bridge domain.

```
l2vpn
  pw-class atom-dyn
  encapsulation mpls
     protocol ldp
  !
!
```

```
    bridge group bg1
        bridge-domain bd1
        igmp snooping profile prof1
        interface GigabitEthernet0/2/0/39
           igmp snooping profile prof2
        vfi mplscore
           neighbor 10.2.2.2 pw-id 101
           pw-class atom-dyn
           !
        !
      !
    !
```

6 Verify the configured bridge ports.

```
show igmp snooping port
```

PE2 Configuration

1 Configure the IGMP profiles.

```
igmp snooping profile bridge_profile
!
igmp snooping profile port_profile
   mrouter
!
```

2 Configure interfaces.

```
interface Loopback0
  ipv4 address 10.2.2.2 255.255.255.255
!
interface GigabitEthernet0/2/0/9
   ipv4 address 10.10.10.1 255.255.255.0
   negotiation auto
!
interface GigabitEthernet0/2/0/39
   negotiation auto
   l2transport
!
```

3 Configure OSPF.

```
router ospf 1
  log adjacency changes
  router-id 10.2.2.2
  area 0
    interface Loopback0
    !
    interface GigabitEthernet0/2/0/9
    !
  !
!
```

4 Configure LDP.

```
mpls ldp
   router-id 10.2.2.2
   log neighbor
   !
   interface GigabitEthernet0/2/0/9
   !
!
```

5 Add interfaces to the bridge domain and attach IGMP snooping profiles.

```
l2vpn
   pw-class atom-dyn
   encapsulation mpls
      protocol ldp
   !
 !

 bridge group bg1
     bridge-domain bd1
     igmp snooping profile bridge_profile
     interface GigabitEthernet0/2/0/39
        igmp snooping profile port_profile
     vfi mplscore
        neighbor 10.1.1.1 pw-id 101
        pw-class atom-dyn
        !
     !
   !
 !
```

6 Verify the configured bridge ports.

```
show igmp snooping port
```

Configuring IGMP access-groups

In the example below, a list is configured and attached to an L2VPN bridge port that allows user membership of <*,G> groups 225.0.0.0/24 and 228.0.0.0/24, only. A second access-list is defined that permits <S,G> membership. This access-list is attached to a bridge-port.

```
interface gig 0/2/0/1.1 l2transport
...
!
ipv4 access-list iptv-basic-white-list
 10 permit ipv4 any 225.0.0.0/24
 20 permit ipv4 any 228.0.0.0/24
!
ipv4 access-list iptv-premium-white-list
 10 permit ipv4 192.168.0.1 232.0.1.0/24
 20 permit ipv4 192.168.0.1 232.0.2.0/24
!
igmp snooping profile iptv
 access-group iptv-white-list
!
igmp snooping profile iptv2
 access-group iptv-premium-white-list
!
l2vpn
 bridge group vz
bridge domain vz-iptv
 igmp snooping profile iptv
 interface gig 0/2/0/1.1
 interface gig 0/2/0/1.2
  igmp snooping profile iptv2
 interface gig 0/2/0/1.3
 ...
!
```

IGMP routing also supports access-groups using the igmp **access-group** command. It uses simple IP access-groups to specify group address filters. In order to support source-group filters as well as group filters, IGMP Snooping requires extended IP access-lists.

Note Access-groups are not applied to static groups and source-groups.

Configuring IGMP Snooping over MCLAG: Example

Case 1: Downstream MCLAG

Topology : DHD connected to 2 POAs which in turn is connected to PE.

DHD:

1 Configure a bundle towards POA1 and POA2. This device will be masked from the existence of 2 POAs. The bundle considers that it is connected to a single POA.

```
interface Bundle-Ether10
 description interface towards POAs
 lacp switchover suppress-flaps 100
 bundle maximum-active links 1
l2transport
 !

!

interface GigabitEthernet0/0/0/28
 description interface towards POA1
 bundle id 10 mode active
!

interface GigabitEthernet0/0/0/29
 description interface towards POA2
 bundle id 10 mode active
!
```

2 Joins coming to this must be forwarded to POAs over bundle. So, configuring the incoming port (host port) and bundle in L2VPN BD (without snooping).

```
RP/0/RSP0/CPU0:router:DHD# show running-config l2vpn

l2vpn
 bridge group bg1
  bridge-domain bg1_bd1
   interface Bundle-Ether10
   !
interface GigabitEthernet0/0/0/10
   !

  !

 !

!
```

POA1:

1 Configure interfaces (for OSPF and MPLS LDP)

```
interface Loopback0

ipv4 address 20.20.20.20 255.255.255.255

!

interface GigabitEthernet0/2/0/1
description interface towards POA2
ipv4 address 10.0.0.1 255.255.255.0

 negotiation auto

!

interface GigabitEthernet0/2/0/8

 description interface towards PE

 ipv4 address 10.0.1.1 255.255.255.0

 negotiation auto

!
```

2 Configure OSPF and MPLS LDP:

```
router ospf 1
router-id 20.20.20.20
nsf cisco
area 0
interface Loopback0

  !
interface GigabitEthernet0/2/0/1
  !
interface GigabitEthernet0/2/0/8

  !

 !

!
mpls ldp
router-id 20.20.20.20
graceful-restart
interface GigabitEthernet0/2/0/1
 !
interface GigabitEthernet0/2/0/8

 !

!
```

3 Configure an MCLAG bundle towards DHD:

```
interface Bundle-Ether10
description interface towards DHD
lacp switchover suppress-flaps 100
mlacp iccp-group 1
mlacp switchover recovery-delay 60
mlacp port-priority 1
mac-address 0.aaaa.1111
bundle wait-while 0
l2transport
```

```
 !

!
interface GigabitEthernet0/2/0/29
bundle id 10 mode active
!
```

4 Configure redundancy group for MCLAG:

```
redundancy

 iccp
group 1
mlacp node 1
mlacp system mac 0000.aaaa.0000
mlacp system priority 1
member
neighbor 30.30.30.30
!
backbone
interface GigabitEthernet0/2/0/8
  !

  !

 !

!
```

5 Configure IGMP Snooping profile:

```
igmp snooping profile p1
ttl-check disable
router-alert-check disable

!
```

6 Enable IGMP Snooping in the L2VPN BD which includes MCLAG bundle towards DHD and PW towards PE:

```
l2vpn
bridge group bg1
bridge-domain bg1_bd1
igmp snooping profile p1
interface Bundle-Ether10
!
vfi bg1_bd1_vfi
neighbor 40.40.40.40 pw-id 1

    !

   !

  !

 !

!
```

POA2:

1 Configure interfaces (for OSPF and MPLS LDP)

```
interface Loopback0

ipv4 address 30.30.30.30 255.255.255.255
!
```

```
interface GigabitEthernet0/0/0/1
description interface towards POA1
ipv4 address 10.0.0.2 255.255.255.0
negotiation auto
!
interface GigabitEthernet0/0/0/8
description interface towards PE
ipv4 address 10.0.2.1 255.255.255.0
negotiation auto
!
```

2 Configure OSPF and MPLS LDP:

```
router ospf 1
router-id 30.30.30.30
nsf cisco
area 0
interface Loopback0
!
interface GigabitEthernet0/0/0/1
!
interface GigabitEthernet0/0/0/8
!
!

!
mpls ldp
router-id 30.30.30.30
graceful-restart
interface GigabitEthernet0/0/0/1
!
interface GigabitEthernet0/0/0/8

 !

!
```

3 Configure an MCLAG bundle towards DHD:

```
interface Bundle-Ether10
description interface towards DHD
lacp switchover suppress-flaps 100
mlacp iccp-group 1
mlacp switchover recovery-delay 60
mlacp port-priority 2
mac-address 0.aaaa.1111
bundle wait-while 0
l2transport

 !

!
interface GigabitEthernet0/0/0/28
bundle id 10 mode active

!
```

4 Configure redundancy group for MCLAG:

```
redundancy
iccp
group 1
mlacp node 2
mlacp system mac 0000.aaaa.0000
mlacp system priority 1
member
neighbor 20.20.20.20
!
backbone
```

```
interface GigabitEthernet0/0/0/8
   !
  !
 !
!
```

5 Configure IGMP Snooping profile:

```
igmp snooping profile p1
ttl-check disable
router-alert-check disable
!
```

6 Enable IGMP Snooping in the L2VPN BD which includes MCLAG bundle towards DHD and PW towards PE:

```
l2vpn
bridge group bg1
bridge-domain bg1_bd1
igmp snooping profile p1
interface Bundle-Ether10
!
vfi bg1_bd1_vfi
neighbor 40.40.40.40 pw-id 1

    !

   !

  !

 !

!
```

PE:

1 Configure Interfaces :

```
interface Loopback0
ipv4 address 40.40.40.40 255.255.255.255
!
interface GigabitEthernet0/0/0/8
description interface towards POA1
ipv4 address 10.0.1.2 255.255.255.0
negotiation auto
!
interface GigabitEthernet0/0/0/9
description interface towards POA2
ipv4 address 10.0.2.2 255.255.255.0
negotiation auto
!
interface GigabitEthernet0/0/0/20
description interface towards Multicast Router
l2transport
!
!
```

2 Configure OSPF and MPLS LDP:

```
router ospf 1
router-id 40.40.40.40
nsf cisco
area 0
interface Loopback0
!
```

```
interface GigabitEthernet0/0/0/8
!
interface GigabitEthernet0/0/0/9
!
!
!
mpls ldp
router-id 40.40.40.40
graceful-restart

interface GigabitEthernet0/0/0/8
!
interface GigabitEthernet0/0/0/9
!
!
```

3 Configure IGMP Snooping profile:

```
igmp snooping profile p1
ttl-check disable
router-alert-check disable
!
```

4 Enable IGMP Snooping in the L2VPN BD which includes PWs towards both the POAs and a port towards Multicast Router:

```
l2vpn
bridge group bg1
bridge-domain bg1_bd1
igmp snooping profile p1
interface GigabitEthernet0/0/0/20
!
vfi bg1_bd1_vfi
neighbor 20.20.20.20 pw-id 1
!
neighbor 30.30.30.30 pw-id 1
!
!
!
```

Case 2 : Upstream MCLAG

Topology: The multicast router is connected to 2 POAs and which is in turn connected to PE multicast Router.

1 Configure bundle towards POAs.

```
interface Bundle-Ether10
description interface towards POAs
ipv4 address 100.0.0.1 255.255.255.0
lacp switchover suppress-flaps 100
bundle maximum-active links 1
!
interface GigabitEthernet0/0/0/28
description interface towards POA1
bundle id 10 mode active
!
interface GigabitEthernet0/0/0/29
description interface towards POA2
bundle id 10 mode active
!
```

2 Enable multicast routing on the bundle interface:

```
multicast-routing
address-family ipv4
interface Bundle-Ether10
```

```
enable
!
!
!
```

POA1:

1 Configure interfaces (for OSPF and MPLS LDP).

```
interface Loopback0
ipv4 address 20.20.20.20 255.255.255.255
!
interface GigabitEthernet0/2/0/1
description interface towards POA2
ipv4 address 10.0.0.1 255.255.255.0
negotiation auto
!
interface GigabitEthernet0/2/0/8
description interface towards PE
ipv4 address 10.0.1.1 255.255.255.0
negotiation auto
!
```

2 Configure OSPF and MPLS LDP:

```
router ospf 1
router-id 20.20.20.20
nsf cisco
area 0
interface Loopback0
!
interface GigabitEthernet0/2/0/1
!
interface GigabitEthernet0/2/0/8
!
!
!
mpls ldp
router-id 20.20.20.20
graceful-restart
interface GigabitEthernet0/2/0/1
!
interface GigabitEthernet0/2/0/8
!
!
```

3 Configure an MCLAG bundle towards DHD:

```
interface Bundle-Ether10
description interface towards DHD
lacp switchover suppress-flaps 100
mlacp iccp-group 1
mlacp switchover recovery-delay 60
mlacp port-priority 1
mac-address 0.aaaa.1111
bundle wait-while 0
l2transport
!
!
interface GigabitEthernet0/2/0/29
bundle id 10 mode active
!
```

4 Configure redundancy group for MCLAG:

```
redundancy
```

```
iccp
group 1
mlacp node 1
mlacp system mac 0000.aaaa.0000
mlacp system priority 1
member
neighbor 30.30.30.30
!
backbone
interface GigabitEthernet0/2/0/8
!
!
!
!
```

5 Configure IGMP Snooping profile:

```
igmp snooping profile p1
ttl-check disable
router-alert-check disable
!
```

6 Enable IGMP Snooping in the L2VPN BD which includes MCLAG bundle towards DHD and PW towards PE:

```
l2vpn
bridge group bg1
bridge-domain bg1_bd1
igmp snooping profile p1
interface Bundle-Ether10
!
vfi bg1_bd1_vfi
neighbor 40.40.40.40 pw-id 1
!
!
!
!
!
```

POA2:

1 Configure interfaces (for OSPF and MPLS LDP).

```
interface Loopback0
ipv4 address 30.30.30.30 255.255.255.255
!
interface GigabitEthernet0/0/0/1
description interface towards POA1
ipv4 address 10.0.0.2 255.255.255.0
negotiation auto
!
interface GigabitEthernet0/0/0/8
description interface towards PE
ipv4 address 10.0.2.1 255.255.255.0
negotiation auto
!
```

2 Configure OSPF and MPLS LDP:

```
router ospf 1
router-id 30.30.30.30
nsf cisco
area 0
interface Loopback0
!
interface GigabitEthernet0/0/0/1
!
interface GigabitEthernet0/0/0/8
```

```
!
!
!
mpls ldp
router-id 30.30.30.30
graceful-restart
interface GigabitEthernet0/0/0/1
!
interface GigabitEthernet0/0/0/8
!
!
```

3 Configure an MCLAG bundle towards DHD:

```
interface Bundle-Ether10
description interface towards DHD
lacp switchover suppress-flaps 100
mlacp iccp-group 1
mlacp switchover recovery-delay 60
mlacp port-priority 2
mac-address 0.aaaa.1111
bundle wait-while 0
l2transport
!
!
interface GigabitEthernet0/0/0/28
bundle id 10 mode active
!
```

4 Configure redundancy group for MCLAG:

```
redundancy
iccp
group 1
mlacp node 2
mlacp system mac 0000.aaaa.0000
mlacp system priority 1
member
neighbor 20.20.20.20
!
backbone
interface GigabitEthernet0/0/0/8
!
!
!
```

5 Configure IGMP Snooping profile:

```
igmp snooping profile p1
ttl-check disable
router-alert-check disable
!
```

6 Enable IGMP Snooping in the L2VPN BD which includes MCLAG bundle towards DHD and PW towards PE:

```
l2vpn
bridge group bg1
bridge-domain bg1_bd1
igmp snooping profile p1
interface Bundle-Ether10
!
vfi bg1_bd1_vfi
neighbor 40.40.40.40 pw-id 1
!
!
!
!
```

PE:

1 Configure interfaces:

```
interface Loopback0
ipv4 address 40.40.40.40 255.255.255.255
!
interface GigabitEthernet0/0/0/8
description interface towards POA1
ipv4 address 10.0.1.2 255.255.255.0
negotiation auto
!
interface GigabitEthernet0/0/0/9
description interface towards POA2
ipv4 address 10.0.2.2 255.255.255.0
negotiation auto
!
interface GigabitEthernet0/0/0/20
description interface towards Host
l2transport
!
!
```

2 Configure OSPF and MPLS LDP:

```
router ospf 1
router-id 40.40.40.40
nsf cisco
area 0
interface Loopback0
!
interface GigabitEthernet0/0/0/8
!
interface GigabitEthernet0/0/0/9
!
!
!
mpls ldp
router-id 40.40.40.40
graceful-restart
interface GigabitEthernet0/0/0/8
!
interface GigabitEthernet0/0/0/9
!
!
```

3 Configure IGMP Snooping profile:

```
igmp snooping profile p1
ttl-check disable
router-alert-check disable
!
igmp snooping profile p2
mrouter
!
```

4 Enable IGMP Snooping in the L2VPN BD which includes PWs towards both the POAs and a port towards the Host. Configure static mrouter port on the PWs towards both the POAs.

```
l2vpn
bridge group bg1
bridge-domain bg1_bd1
igmp snooping profile p1
interface GigabitEthernet0/0/0/20
!
vfi bg1_bd1_vfi
neighbor 20.20.20.20 pw-id 1
igmp snooping profile p2
!
```

```
neighbor 30.30.30.30 pw-id 1
igmp snooping profile p2
!
!
!
!
```

Additional References

Related Documents

Related Topic	Document Title
Configuring MPLS VPLS bridges	Implementing Virtual Private LAN Services on Cisco IOS XR Software module in the *Cisco ASR 9000 Series Aggregation Services Router MPLS Configuration Guide*
Getting started information	*Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide*
Configuring EFPs and EFP bundles	*Cisco ASR 9000 Series Aggregation Services Router Interface and Hardware Component Configuration Guide*

Standards

Standards[1]	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

1 Not all supported standards are listed.

MIBs

MIBs	MIBs Link
No MIBs support IGMP snooping.	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
RFC-4541	Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

CHAPTER 3

Implementing MLD Snooping

This module describes how to implement MLD snooping on the Cisco ASR 9000 Series Router.

Feature History for MLD Snooping

Release	Modification
Release 4.3.0	This feature was introduced.

MLD Snooping

Multicast Listener Discovery (MLD) snooping provides a way to constrain multicast traffic at Layer 2. By snooping the MLD membership reports sent by hosts in the bridge domain, the MLD snooping application can set up Layer 2 multicast forwarding tables to deliver traffic only to ports with at least one interested member, significantly reducing the volume of multicast traffic.

MLD snooping uses the information in MLD membership report messages to build corresponding information in the forwarding tables to restrict IPv6 multicast traffic at Layer 2. The forwarding table entries are in the form <Route, OIF List>, where:

- Route is a <*, G> route or <S, G> route.
- OIF List comprises all bridge ports that have sent MLD membership reports for the specified route plus all multicast router (mrouter) ports in the bridge domain.

For more information regarding MLD snooping, refer the *Cisco ASR 9000 Series Aggregation Services Router Multicast Configuration Guide*.

Prerequisites for MLD Snooping

- The network must be configured with a layer2 VPN.
- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Restrictions for MLD Snooping

Following are the restrictions (features that are not supported):

- MLD Snooping is supported only on L2VPN bridge domains.
- Explicit host tracking.
- Multicast Admission Control.
- Security filtering.
- Report rate limiting.
- Multicast router discovery.

Advantages of MLD Snooping

Advantages of MLD Snooping

- In its basic form, it reduces bandwidth consumption by reducing multicast traffic that would otherwise flood an entire VPLS bridge domain.

- With the use of some optional configurations, it provides security between bridge domains by filtering the MLD reports received from hosts on one bridge port and preventing leakage towards the hosts on other bridge ports.

High Availability (HA) features for MLD

MLD supports the following HA features:

- Process restarts
- RP Failover
- Stateful Switch-Over (SSO)
- Non-Stop Forwarding (NSF)—Forwarding continues unaffected while the control plane is restored following a process restart or route processor (RP) failover.
- Line card online insertion and removal (OIR)

Bridge Domain Support for MLD

MLD snooping operates at the bridge domain level. When MLD snooping is enabled on a bridge domain, the snooping functionality applies to all ports under the bridge domain, including:

- Physical ports under the bridge domain.
- Ethernet flow points (EFPs)—An EFP can be a VLAN, VLAN range, list of VLANs, or an entire interface port.
- Pseudowires (PWs) in VPLS bridge domains.
- Ethernet bundles—Ethernet bundles include IEEE 802.3ad link bundles and Cisco EtherChannel bundles. From the perspective of the MLD snooping application, an Ethernet bundle is just another EFP. The forwarding application in the Cisco ASR 9000 Series Routers randomly nominates a single port from the bundle to carry the multicast traffic.

Multicast Router and Host Ports

MLD snooping classifies each port as one of the following:

- Multicast router ports (mrouter ports)—These are ports to which a multicast-enabled router is connected. Mrouter ports are usually dynamically discovered, but may also be statically configured. Multicast traffic is always forwarded to all mrouter ports, except when an mrouter port is the ingress port.
- Host ports—Any port that is not an mrouter port is a host port.

Multicast Router Discovery for MLD

MLD snooping discovers mrouter ports dynamically. You can also explicitly configure a port as an emrouter port.

- Discovery- MLD snooping identifies upstream mrouter ports in the bridge domain by snooping mld query messages and Protocol Independent Multicast Version 2 (PIMv2) hello messages. Snooping PIMv2 hello messages identifies mld nonqueriers in the bridge domain.
- Static configuration—You can statically configure a port as an mrouter port with the **mrouter** command in a profile attached to the port. Static configuration can help in situations when incompatibilities with non-Cisco equipment prevent dynamic discovery.

Multicast Traffic Handling for MLD

The following tables describe the traffic handling behavior by MLD mrouters and host ports.

Table 6: Multicast Traffic Handling for a MLDv1 Querier

Traffic Type	Received on MRouter Ports	Received on Host Ports
IP multicast source traffic	Forwards to all mrouter ports and to host ports that indicate interest.	Forwards to all mrouter ports and to host ports that indicate interest.
MLD general queries	Forwards to all ports.	—
MLD group-specific queries	Forwards to all other mrouter ports.	Dropped
MLDv1 joins	Examines (snoops) the reports. • If report suppression is enabled, forwards first join for a new group or first join following a general query for an existing group. • If report suppression is disabled, forwards on all mrouter ports.	Examines (snoops) the reports. • If report suppression is enabled, forwards first join for a new group or first join following a general query for an existing group. • If report suppression is disabled, forwards on all mrouter ports.
MLDv2 reports	Ignores	Ignores
MLDv1 leaves	Invokes last member query processing.	Invokes last member query processing.

Table 7: Multicast Traffic Handling for a MLDv2 Querier

Traffic Type	Received on MRouter Ports	Received on Host Ports
IP multicast source traffic	Forwards to all mrouter ports and to host ports that indicate interest.	Forwards to all mrouter ports and to host ports that indicate interest.
MLD general queries	Forwards to all ports.	—
MLD group-specific queries	If received on the querier port floods on all ports.	—
MLDv1 joins	Handles as MLDv2 IS_EX{} reports.	Handles as MLDv2 IS_EX{} reports.
MLDv2 reports	• If proxy reporting is enabled—For state changes or source-list changes, generates a state change report on all mrouter ports. • If proxy reporting is disabled—Forwards on all mrouter ports.	• If proxy reporting is enabled—For state changes or source-list changes, generates a state change report on all mrouter ports. • If proxy reporting is disabled—Forwards on all mrouter ports.
MLDv1 leaves	Handles as MLDv2 IS_IN{} reports.	Handles as MLDv2 IS_IN{} reports.

Creating a MLD Snooping Profile

SUMMARY STEPS

1. **configure**
2. **mld snooping profile** *profile-name*
3. Optionally, add commands to override default configuration values.
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**mld snooping profile** *profile-name*	Enters MLD snooping profile configuration mode and creates a named profile.

	Command or Action	Purpose
	Example: `RP/0/RSP0/CPU0:router(config)# mld snooping profile default-bd-profile`	The default profile enables MLD snooping. You can commit the new profile without any additional configurations, or you can include additional configuration options to the profile. You can also return to the profile later to add configurations, as described in other tasks in this module.
Step 3	Optionally, add commands to override default configuration values.	If you are creating a bridge domain profile, consider the following: • An empty profile is appropriate for attaching to a bridge domain. An empty profile enables MLD snooping with default configuration values. • You can optionally add more commands to the profile to override default configuration values. • If you include port-specific configurations in a bridge domain profile, the configurations apply to all ports under the bridge, unless another profile is attached to a port. If you are creating a port-specific profile, consider the following: • While an empty profile could be attached to a port, it would have no effect on the port configuration. • When you attach a profile to a port, MLD snooping reconfigures that port, overriding any inheritance of configuration values from the bridge-domain profile. You must repeat the commands in the port profile if you want to retain those configurations. You can detach a profile, change it, and reattach it to add commands to a profile at a later time.
Step 4	**commit**	

Activating MLD Snooping on a Bridge Domain

To activate MLD snooping on a bridge domain, attach a MLD snooping profile to the desired bridge domain as explained here.

SUMMARY STEPS

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *bridge-domain-name*
5. **mld snooping profile** *profile-name*
6. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**l2vpn** **Example:** `RP/0/RSP0/CPU0:router(config)# l2vpn`	Enters Layer 2 VPN configuration mode.
Step 3	**bridge group** *bridge-group-name* **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group GRP1`	Enters Layer 2 VPN VPLS bridge group configuration mode for the named bridge group.
Step 4	**bridge-domain** *bridge-domain-name* **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain ISP1`	Enters Layer 2 VPN VPLS bridge group bridge domain configuration mode for the named bridge domain.
Step 5	**mld snooping profile** *profile-name* **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# mld snooping profile default-bd-profile`	Attaches the named MLD snooping profile to the bridge domain, enabling MLD snooping on the bridge domain.
Step 6	**commit**	

Deactivating MLD Snooping on a Bridge Domain

To deactivate MLD snooping from a bridge domain, remove the profile from the bridge domain using the following steps:

Note A bridge domain can have only one profile attached to it at a time.

SUMMARY STEPS

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *bridge-domain-name*
5. **no mld snooping**
6. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**l2vpn** **Example:** `RP/0/RSP0/CPU0:router(config)# l2vpn`	Enters Layer 2 VPN configuration mode.
Step 3	**bridge group** *bridge-group-name* **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group GRP1`	Enters Layer 2 VPN VPLS bridge group configuration mode for the named bridge group.
Step 4	**bridge-domain** *bridge-domain-name* **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain ISP1`	Enters Layer 2 VPN VPLS bridge group bridge domain configuration mode for the named bridge domain.
Step 5	**no mld snooping** **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# no mld snooping`	Detaches the MLD snooping profile from the bridge domain, disabling MLD snooping on that bridge domain. **Note** Only one profile can be attached to a bridge domain at a time. If a profile is attached, MLD snooping is enabled. If a profile is not attached, MLD snooping is disabled.
Step 6	**commit**	

Configuring Static Mrouter Ports (MLD)

Before You Begin

MLD snooping must be enabled on the bridge domain for port-specific profiles to affect MLD snooping behavior.

Note Static mrouter port configuration is a port-level option and should be added to profiles intended for ports. It is not recommended to add mrouter port configuration to a profile intended for bridge domains.

SUMMARY STEPS

1. **configure**
2. **mld snooping profile** *profile-name*
3. **mrouter**
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**mld snooping profile** *profile-name* **Example:** `RP/0/RSP0/CPU0:router(config)# mld snooping profile mrouter-port-profile`	Enters MLD snooping profile configuration mode and creates a new profile or accesses an existing profile.
Step 3	**mrouter** **Example:** `RP/0/RSP0/CPU0:router(config-mld-snooping-profile)# mrouter`	Configures a port as a static mrouter port.
Step 4	**commit**	

Configuring Router Guard (MLD)

To prevent multicast routing protocol messages from being received on a port and, therefore, prevent a port from being a dynamic mrouter port, follow these steps. Note that both router guard and static mrouter commands may be configured on the same port.

Before You Begin

MLD snooping must be enabled on the bridge domain for port-specific profiles to affect MLD snooping behavior.

Note Router guard configuration is a port-level option and should be added to profiles intended for ports. It is not recommended to add router guard configuration to a profile intended for bridge domains. To do so would prevent all mrouters, including MLD queriers, from being discovered in the bridge domain.

SUMMARY STEPS

1. **configure**
2. **mld snooping profile** *profile-name*
3. **router-guard**
4. **commit**
5. **show mld snooping profile** *profile-name* **detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**mld snooping profile** *profile-name* **Example:** `RP/0/RSP0/CPU0:router(config)# mld snooping profile host-port-profile`	Enters MLD snooping profile configuration mode and creates a new profile or accesses an existing profile.
Step 3	**router-guard** **Example:** `RP/0/RSP0/CPU0:router(config-mld-snooping-profile)# router-guard`	Protects the port from dynamic discovery.
Step 4	**commit**	
Step 5	**show mld snooping profile** *profile-name* **detail** **Example:** `RP/0/RSP0/CPU0:router# show mld snooping profile host-port-profile detail`	(Optional) Displays the configuration settings in the named profile.

Configuring Immediate-leave for MLD

To add the MLD snooping immediate-leave option to an MLD snooping profile, follow these steps.

SUMMARY STEPS

1. **configure**
2. **mld snooping profile** *profile-name*
3. **immediate-leave**
4. **commit**
5. **show mld snooping profile** *profile-name* **detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**mld snooping profile** *profile-name* **Example:** `RP/0/RSP0/CPU0:router(config)# mld snooping profile host-port-profile`	Enters MLD snooping profile configuration mode and creates a new profile or accesses an existing profile.
Step 3	**immediate-leave** **Example:** `RP/0/RSP0/CPU0:router(config-mld-snooping-profile)# immediate-leave`	Enables the immediate-leave option. • If you add this option to a profile attached to a bridge domain, it applies to all ports under the bridge. • If you add this option to a profile attached to a port, it applies to the port.
Step 4	**commit**	
Step 5	**show mld snooping profile** *profile-name* **detail** **Example:** `RP/0/RSP0/CPU0:router# show mld snooping profile host-port-profile detail`	(Optional) Displays the configuration settings in the named profile.

Configuring Internal Querier for MLD

Before You Begin

MLD snooping must be enabled on the bridge domain for this procedure to take effect.

SUMMARY STEPS

1. **configure**
2. **mld snooping profile** *profile-name*
3. **system-ip-address** *ip-addr*
4. **internal-querier**
5. **commit**
6. **show mld snooping profile** *profile-name* **detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**mld snooping profile** *profile-name* **Example:** `RP/0/RSP0/CPU0:router(config)# mld snooping profile internal-querier-profile`	Enters MLD snooping profile configuration mode and creates a new profile or accesses an existing profile.
Step 3	**system-ip-address** *ip-addr* **Example:** `RP/0/RSP0/CPU0:router(config-mld-snooping-profile)# system-ip-address 10.1.1.1`	Configures an IP address for internal querier use. The default system-ip-address value (0.0.0.0) is not valid for the internal querier. You must explicitly configure an IP address.
Step 4	**internal-querier** **Example:** `RP/0/RSP0/CPU0:router(config-mld-snooping-profile)# internal-querier`	Enables an internal querier with default values for all options.
Step 5	**commit**	
Step 6	**show mld snooping profile** *profile-name* **detail** **Example:** `RP/0/RSP0/CPU0:router# show mld snooping profile internal-querier-profile detail`	(Optional) Displays the configuration settings in the named profile.

Configuring Static Groups for MLD

To add one or more static groups or MLDv2 source groups to an MLD snooping profile, follow these steps.

Before You Begin

MLD snooping must be enabled on the bridge domain for port-specific profiles to affect MLD snooping behavior.

SUMMARY STEPS

1. **configure**
2. **mld snooping profile** *profile-name*
3. **static-group** *group-addr* [**source** *source-addr*]
4. Repeat the previous step, as needed, to add more static groups.
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**mld snooping profile** *profile-name* **Example:** `RP/0/RSP0/CPU0:router(config)# mld snooping profile host-port-profile`	Enters MLD snooping profile configuration mode and creates a new profile or accesses an existing profile.
Step 3	**static-group** *group-addr* [**source** *source-addr*] **Example:** `RP/0/RSP0/CPU0:router(config-mld-snooping-profile)# static-group 239.1.1.1 source 10.0.1.1`	Configures a static group. • If you add this option to a profile attached to a bridge domain, it applies to all ports under the bridge. • If you add this option to a profile attached to a port, it applies to the port.
Step 4	Repeat the previous step, as needed, to add more static groups.	(Optional) Adds additional static groups.
Step 5	**commit**	

Configuring MLD Snooping

1. Create two profiles:

```
mld snooping profile bridge_profile
!
mld snooping profile port_profile
   mrouter
!
```

2. Configure two physical interfaces for L2 support.

```
interface GigabitEthernet0/8/0/38
   negotiation auto
   l2transport
   no shut
   !
!
interface GigabitEthernet0/8/0/39
   negotiation auto
   l2transport
   no shut
   !
!
```

3. Add interfaces to the bridge domain. Attach bridge_profile to the bridge domain and port_profile to one of the Ethernet interfaces. The second Ethernet interface inherits MLD snooping configuration attributes from the bridge domain profile.

```
l2vpn
  bridge group bg1
     bridge-domain bd1
     mld snooping profile bridge_profile
     interface GigabitEthernet0/8/0/38
       mld snooping profile port_profile
     interface GigabitEthernet0/8/0/39
     !
  !
!
```

4. Verify the configured bridge ports.

```
show mld snooping port
```

Configuring MLD Snooping on Ethernet Bundles

1. This example assumes that the front-ends of the bundles are preconfigured. For example, a bundle configuration might consist of three switch interfaces, as follows:

```
    interface Port-channel1
    !
interface GigabitEthernet0/0/0/0
    !
interface GigabitEthernet0/0/0/1
!
    interface GigabitEthernet0/0/0/2
       channel-group 1 mode on
    !
```

```
interface GigabitEthernet0/0/0/3
   channel-group 1 mode on
!
```

2 Configure two MLD snooping profiles.

```
mld snooping profile bridge_profile
!
mld snooping profile port_profile
   mrouter
!
```

3 Configure interfaces as bundle member links.

```
interface GigabitEthernet0/0/0/0
  bundle id 1 mode on
  negotiation auto
!
interface GigabitEthernet0/0/0/1
  bundle id 1 mode on
  negotiation auto
!
interface GigabitEthernet0/0/0/2
  bundle id 2 mode on
  negotiation auto
!
interface GigabitEthernet0/0/0/3
  bundle id 2 mode on
  negotiation auto
!
```

4 Configure the bundle interfaces for L2 transport.

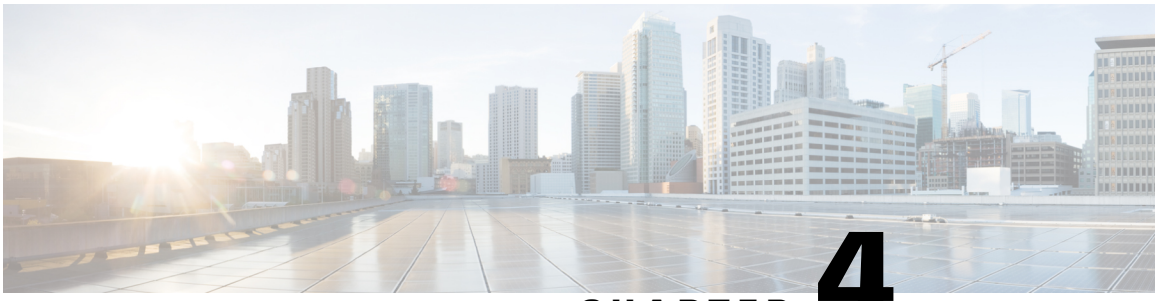
```
interface Bundle-Ether 1
          l2transport
          !
       !
       interface Bundle-Ether 2
          l2transport
          !
       !
```

5 Add the interfaces to the bridge domain and attach MLD snooping profiles.

```
l2vpn
  bridge group bg1
     bridge-domain bd1
     mld snooping profile bridge_profile
     interface bundle-Ether 1
       mld snooping profile port_profile
     interface bundle-Ether 2
     !
  !
!
```

6 Verify the configured bridge ports.

```
show mld snooping port
```



CHAPTER 4

Implementing Layer-3 Multicast Routing on Cisco IOS XR Software

This module describes how to implement Layer 3 multicast routing on Cisco ASR 9000 Series Routers running Cisco IOS XR Software.

Multicast routing is a bandwidth-conserving technology that reduces traffic by simultaneously delivering a single stream of information to potentially thousands of corporate recipients and homes. Applications that take advantage of multicast routing include video conferencing, corporate communications, distance learning, and distribution of software, stock quotes, and news.

This document assumes that you are familiar with IPv4 and IPv6 multicast routing configuration tasks and concepts for Cisco IOS XR Software .

Multicast routing allows a host to send packets to a subset of all hosts as a group transmission rather than to a single host, as in unicast transmission, or to all hosts, as in broadcast transmission. The subset of hosts is known as **group members** and are identified by a single multicast group address that falls under the IP Class D address range from 224.0.0.0 through 239.255.255.255.

For detailed conceptual information about multicast routing and complete descriptions of the multicast routing commands listed in this module, you can refer to the Related Documents, on page 235.

Feature History for Configuring Multicast Routing on the Cisco ASR 9000 Series Routers

Release	Modification
Release 3.7.2	This feature was introduced.
Release 3.9.0	Support was added for these features: • Flow-based multicast only fast reroute (MoFRR). • IGMP VRF override.
Release 3.9.1	Support was added for the Multicast VPN feature. (For IPv4 address family)

Release	Modification
Release 4.0.0	Support was added for IPv4 Multicast routing, Multicast VPN basic and InterAS option A on Cisco ASR 9000 Series SPA Interface Processor-700 linecard and MVPN Hub and Spoke Topology.
Release 4.0.1	Support was added for IPv6 Multicast routing.
Release 4.1.0	Support was added for Label Switched Multicast using Point-to-Multipoint Traffic Engineering in global context only (not in VRF).
Release 4.2.1	Support was added for these features: • Label Switched Multicast using MLDP (Multicast Label Distribution Protocol). • Multicast VPN for IPv6 address family. • Support for Satellite nV. • InterAS Support on Multicast VPN.
Release 4.3.2	Support was added for these features: • Support for IPv4 traffic on Multicast over unicast GRE was introduced. • Support was added for TI (Topology Independent) MoFRR.
Release 5.2.0	Support was introduced for Bidirectional Global Protocol Independent Multicast.
Release 6.1.2	Layer 3 Multicast Bundle Subinterface Load Balancing feature was introduced.
Release 6.1.2	Segmented Multicast Stitching with Inter AS and MLDP Carrier Supporting Carrier based MVPN feature support was extended to support Cisco IOS XR 64 bit.
Release 6.1.2	MVPN, MoGRE, MoFRR and Global Table Multicast feature support was extended to support Cisco IOS XR 64 bit.

- Prerequisites for Implementing Multicast Routing, page 73
- Information About Implementing Multicast Routing, page 73
- How to Implement Multicast Routing, page 120
- Multicast only fast reroute (MoFRR), page 162
- Enabling multicast on PW-HE interfaces, page 169

Prerequisites for Implementing Multicast Routing

- You must install and activate the multicast pie.
- For detailed information about optional PIE installation, see *Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide*
- For MLDP, an MPLS PIE has to be installed.
- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- You must be familiar with IPv4 and IPv6 multicast routing configuration tasks and concepts.
- Unicast routing must be operational.
- To enable multicast VPN, you must configure a VPN routing and forwarding (VRF) instance.

Information About Implementing Multicast Routing

Key Protocols and Features Supported in the Cisco IOS XR Software Multicast Routing Implementation

Table 8: Supported Features for IPv4 and IPv6

Feature	IPv4 Support	IPv6 Support
Dynamic host registration	Yes (IGMP v1/2/3)	Yes
Explicit tracking of hosts, groups, and channels	Yes (IGMP v3)	Yes
PIM-SM[2]	Yes	Yes
PIM-SSM	Yes	Yes
PIM-SSM Mapping	Yes	Yes

Feature	IPv4 Support	IPv6 Support
Auto-RP	Yes	No
Multicast VPN	Yes	Yes
InterAS Option A	Yes	Yes
BSR	Yes	Yes
BGP	Yes	Yes
MSDP	Yes	No
Multicast NSF	Yes	Yes
OOR handling	Yes	Yes
Global Protocol Independent Multicast Bidirectional support	Yes	Yes

2

Multicast Routing Functional Overview

Traditional IP communication allows a host to send packets to a single host (*unicast transmission*) or to all hosts (*broadcast transmission*). Multicast provides a third scheme, allowing a host to send a single data stream to a subset of all hosts (*group transmission*) at about the same time. IP hosts are known as group members.

Packets delivered to group members are identified by a single multicast group address. Multicast packets are delivered to a group using best-effort reliability, just like IP unicast packets.

The multicast environment consists of senders and receivers. Any host, regardless of whether it is a member of a group, can send to a group. However, only the members of a group receive the message.

A multicast address is chosen for the receivers in a multicast group. Senders use that group address as the destination address of a datagram to reach all members of the group.

Membership in a multicast group is dynamic; hosts can join and leave at any time. There is no restriction on the location or number of members in a multicast group. A host can be a member of more than one multicast group at a time.

How active a multicast group is and what members it has can vary from group to group and from time to time. A multicast group can be active for a long time, or it may be very short-lived. Membership in a group can change constantly. A group that has members may have no activity.

Routers use the Internet Group Management Protocol (IGMP) (IPv4) and Multicast Listener Discovery (MLD) (IPv6) to learn whether members of a group are present on their directly attached subnets. Hosts join multicast groups by sending IGMP or MLD report messages.

Many multimedia applications involve multiple participants. Multicast is naturally suitable for this communication paradigm.

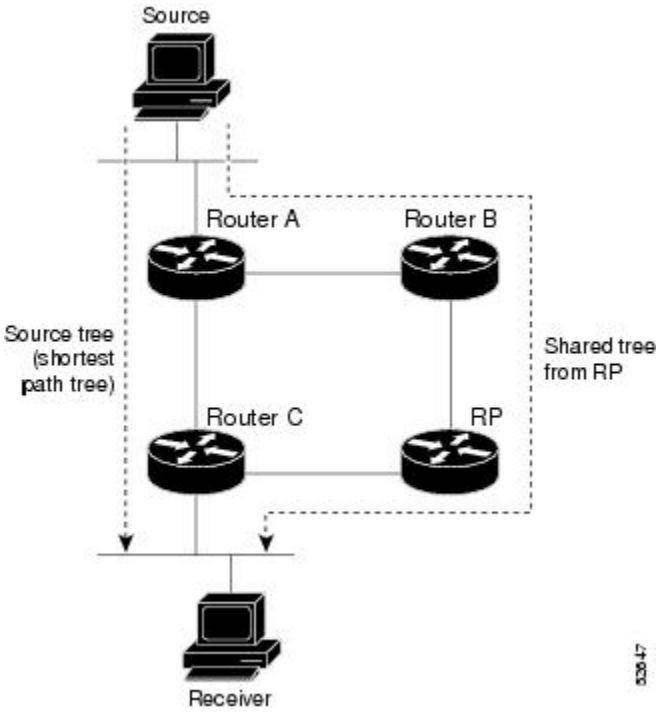
Multicast Routing Implementation

Cisco IOS XR Software supports the following protocols to implement multicast routing:

- IGMP is used between hosts on a LAN and the routers on that LAN to track the multicast groups of which hosts are members.
- Protocol Independent Multicast in sparse mode (PIM-SM) is used between routers so that they can track which multicast packets to forward to each other and to their directly connected LANs.
- Protocol Independent Multicast in Source-Specific Multicast (PIM-SSM) is similar to PIM-SM with the additional ability to report interest in receiving packets from specific source addresses (or from all but the specific source addresses), to an IP multicast address.
- PIM-SSM is made possible by IGMPv3 and MLDv2. Hosts can now indicate interest in specific sources using IGMPv3 and MLDv2. SSM does not require a rendezvous point (RP) to operate.
- PIM Bidirectional is a variant of the Protocol Independent Multicast suit of routing protocols for IP multicast. PIM-BIDIR is designed to be used for many-to-many applications within individual PIM domains.

This image shows IGMP and PIM-SM operating in a multicast environment.

Figure 1: Multicast Routing Protocols



PIM-SM, PIM-SSM, and PIM-BIDIR

Protocl Independent Multicast (PIM) is a multicast routing protocol used to create multicast distribution trees, which are used to forward multicast data packets. PIM is an efficient IP routing protocol that is “independent” of a routing table, unlike other multicast protocols such as Multicast Open Shortest Path First (MOSPF) or Distance Vector Multicast Routing Protocol (DVMRP).

Cisco IOS XR Software supports Protocol Independent Multicast in sparse mode (PIM-SM), Protocol Independent Multicast in Source-Specific Multicast (PIM-SSM), and Protocol Independent Multicast in Bi-directional mode (BIDIR) permitting these modes to operate on your router at the same time.

PIM-SM and PIM-SSM supports one-to-many applications by greatly simplifying the protocol mechanics for deployment ease. Bidir PIM helps deploy emerging communication and financial applications that rely on a many-to-many applications model. BIDIR PIM enables these applications by allowing them to easily scale to a very large number of groups and sources by eliminating the maintenance of source state.

PIM-SM Operations

PIM in sparse mode operation is used in a multicast network when relatively few routers are involved in each multicast and these routers do not forward multicast packets for a group, unless there is an explicit request for the traffic.

For more information about PIM-SM, see the PIM-Sparse Mode, on page 79.

PIM-SSM Operations

PIM in Source-Specific Multicast operation uses information found on source addresses for a multicast group provided by receivers and performs source filtering on traffic.

- By default, PIM-SSM operates in the 232.0.0.0/8 multicast group range for IPv4 and ff3x::/32 (where x is any valid scope) in IPv6. To configure these values, use the **ssm range** command.
- If SSM is deployed in a network already configured for PIM-SM, only the last-hop routers must be upgraded with Cisco IOS XR Software that supports the SSM feature.
- No MSDP SA messages within the SSM range are accepted, generated, or forwarded.

PIM-Bidirectional Operations

PIM Bidirectional (BIDIR) has one shared tree from sources to RP and from RP to receivers. This is unlike the PIM-SM, which is unidirectional by nature with multiple source trees - one per (S,G) or a shared tree from receiver to RP and multiple SG trees from RP to sources.

Benefits of PIM BIDIR are as follows:

- As many sources for the same group use one and only state (*, G), only minimal states are required in each router.
- No data triggered events.
- Rendezvous Point (RP) router not required. The RP address only needs to be a routable address and need not exist on a physical device.

Restrictions for PIM-SM and PIM-SSM, and PIM BIDIR

Interoperability with SSM

PIM-SM operations within the SSM range of addresses change to PIM-SSM. In this mode, only PIM (S,G) join and prune messages are generated by the router, and no (S,G) RP shared tree or (*,G) shared tree messages are generated.

IGMP Version

To report multicast memberships to neighboring multicast routers, hosts use IGMP, and all routers on the subnet must be configured with the same version of IGMP.

A router running Cisco IOS XR Software does not automatically detect Version 1 systems. You must use the **version** command in router IGMP configuration submode to configure the IGMP version.

PIM-Bidir Restrictions

PIM-Bidir is not supported on MVPN.

Internet Group Management Protocol

Cisco IOS XR Software provides support for Internet Group Management Protocol (IGMP) over IPv4.

IGMP provides a means for hosts to indicate which multicast traffic they are interested in and for routers to control and limit the flow of multicast traffic throughout the network. Routers build state by means of IGMP and MLD messages; that is, router queries and host reports.

A set of queries and hosts that receive multicast data streams from the same source is called a *multicast group*. Hosts use IGMP and MLD messages to join and leave multicast groups.

Note IGMP messages use group addresses, which are Class D IP addresses. The high-order four bits of a Class D address are 1110. Host group addresses can be in the range 224.0.0.0 to 239.255.255.255. The address 224.0.0.0 is guaranteed not to be assigned to any group. The address 224.0.0.1 is assigned to all systems on a subnet. The address 224.0.0.2 is assigned to all routers on a subnet.

IGMP Versions

The following points describe IGMP versions 1, 2, and 3:

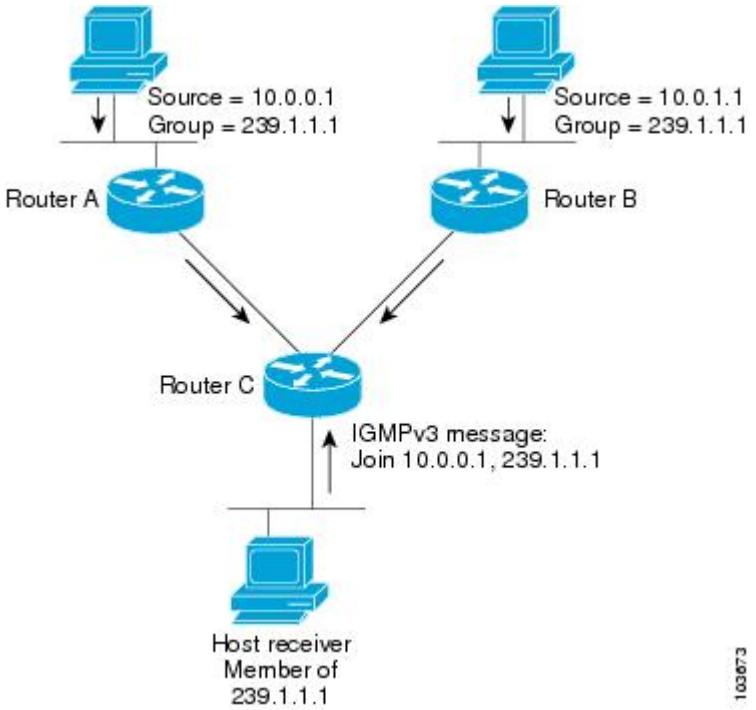
- IGMP Version 1 provides for the basic query-response mechanism that allows the multicast router to determine which multicast groups are active and for other processes that enable hosts to join and leave a multicast group.
- IGMP Version 2 extends IGMP allowing such features as the IGMP query timeout and the maximum query-response time. See RFC 2236.
- IGMP Version 3 permits joins and leaves for certain source and group pairs instead of requesting traffic from all sources in the multicast group.

IGMP Routing Example

Figure 2: IGMPv3 Signaling, on page 78 illustrates two sources, 10.0.0.1 and 10.0.1.1, that are multicasting to group 239.1.1.1. The receiver wants to receive traffic addressed to group 239.1.1.1 from source 10.0.0.1 but not from source 10.0.1.1. The host must send an IGMPv3 message containing a list of sources and groups (S, G) that it wants to join and a list of sources and groups (S, G) that it wants to leave. Router C can now use this information to prune traffic from Source 10.0.1.1 so that only Source 10.0.0.1 traffic is being delivered to

Router C.

Figure 2: IGMPv3 Signaling



Note When configuring IGMP, ensure that all systems on the subnet support the same IGMP version. The router does not automatically detect Version 1 systems. Configure the router for Version 2 if your hosts do not support Version 3.

Protocol Independent Multicast

Protocol Independent Multicast (PIM) is a routing protocol designed to send and receive multicast routing updates. Proper operation of multicast depends on knowing the unicast paths towards a source or an RP. PIM relies on unicast routing protocols to derive this reverse-path forwarding (RPF) information. As the name PIM implies, it functions independently of the unicast protocols being used. PIM relies on the Routing Information Base (RIB) for RPF information.

If the multicast subsequent address family identifier (SAFI) is configured for Border Gateway Protocol (BGP), or if multicast intact is configured, a separate multicast unicast RIB is created and populated with the BGP multicast SAFI routes, the intact information, and any IGP information in the unicast RIB. Otherwise, PIM gets information directly from the unicast SAFI RIB. Both multicast unicast and unicast databases are outside of the scope of PIM.

The Cisco IOS XR implementation of PIM is based on RFC 4601 Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification. For more information, see RFC 4601 and the Protocol Independent Multicast (PIM): Motivation and Architecture Internet Engineering Task Force (IETF) Internet draft.

Note Cisco IOS XR Software supports PIM-SM, PIM-SSM, and PIM Version 2 only. PIM Version 1 hello messages that arrive from neighbors are rejected.

PIM-Sparse Mode

Typically, PIM in sparse mode (PIM-SM) operation is used in a multicast network when relatively few routers are involved in each multicast. Routers do not forward multicast packets for a group, unless there is an explicit request for traffic. Requests are accomplished using PIM join messages, which are sent hop by hop toward the root node of the tree. The root node of a tree in PIM-SM is the rendezvous point (RP) in the case of a shared tree or the first-hop router that is directly connected to the multicast source in the case of a shortest path tree (SPT). The RP keeps track of multicast groups, and the sources that send multicast packets are registered with the RP by the first-hop router of the source.

As a PIM join travels up the tree, routers along the path set up the multicast forwarding state so that the requested multicast traffic is forwarded back down the tree. When multicast traffic is no longer needed, a router sends a PIM prune message up the tree toward the root node to prune (or remove) the unnecessary traffic. As this PIM prune travels hop by hop up the tree, each router updates its forwarding state appropriately. Ultimately, the forwarding state associated with a multicast group or source is removed. Additionally, if prunes are not explicitly sent, the PIM state will timeout and be removed in the absence of any further join messages.

PIM-SM is the best choice for multicast networks that have potential members at the end of WAN links.

PIM-Source Specific Multicast

In many multicast deployments where the source is known, protocol-independent multicast-source-specific multicast (PIM-SSM) mapping is the obvious multicast routing protocol choice to use because of its simplicity. Typical multicast deployments that benefit from PIM-SSM consist of entertainment-type solutions like the ETTH space, or financial deployments that completely rely on static forwarding.

PIM-SSM is derived from PIM-SM. However, whereas PIM-SM allows for the data transmission of all sources sending to a particular group in response to PIM join messages, the SSM feature forwards traffic to receivers only from those sources that the receivers have explicitly joined. Because PIM joins and prunes are sent directly towards the source sending traffic, an RP and shared trees are unnecessary and are disallowed. SSM is used to optimize bandwidth utilization and deny unwanted Internet broadcast traffic. The source is provided by interested receivers through IGMPv3 membership reports.

In SSM, delivery of datagrams is based on (S,G) channels. Traffic for one (S,G) channel consists of datagrams with an IP unicast source address S and the multicast group address G as the IP destination address. Systems receive traffic by becoming members of the (S,G) channel. Signaling is not required, but receivers must subscribe or unsubscribe to (S,G) channels to receive or not receive traffic from specific sources. Channel

subscription signaling uses IGMP to include mode membership reports, which are supported only in Version 3 of IGMP (IGMPv3).

To run SSM with IGMPv3, SSM must be supported on the multicast router, the host where the application is running, and the application itself. Cisco IOS XR Software allows SSM configuration for an arbitrary subset of the IP multicast address range 224.0.0.0 through 239.255.255.255. When an SSM range is defined, existing IP multicast receiver applications do not receive any traffic when they try to use addresses in the SSM range, unless the application is modified to use explicit (S,G) channel subscription.

PIM-Bidirectional Mode

PIM BIDIR is a variant of the Protocol Independent Multicast (PIM) suite of routing protocols for IP multicast. In PIM, packet traffic for a multicast group is routed according to the rules of the mode configured for that multicast group. In bidirectional mode, traffic is only routed along a bidirectional shared tree that is rooted at the rendezvous point (RP) for the group. In PIM-BIDIR, the IP address of the RP acts as the key to having all routers establish a loop-free spanning tree topology rooted in that IP address. This IP address does not need to be a router, but can be any unassigned IP address on a network that is reachable throughout the PIM domain. Using this technique is the preferred configuration for establishing a redundant RP configuration for PIM-BIDIR.

Note In Cisco IOS XR Release 4.2.1, Anycast RP is not supported on PIM Bidirectional mode.

PIM-BIDIR is designed to be used for many-to-many applications within individual PIM domains. Multicast groups in bidirectional mode can scale to an arbitrary number of sources without incurring overhead due to the number of sources. PIM-BIDIR is derived from the mechanisms of PIM-sparse mode (PIM-SM) and shares many SPT operations. PIM-BIDIR also has unconditional forwarding of source traffic toward the RP upstream on the shared tree, but no registering process for sources as in PIM-SM. These modifications are necessary and sufficient to allow forwarding of traffic in all routers solely based on the (*, G) multicast routing entries. This feature eliminates any source-specific state and allows scaling capability to an arbitrary number of sources.

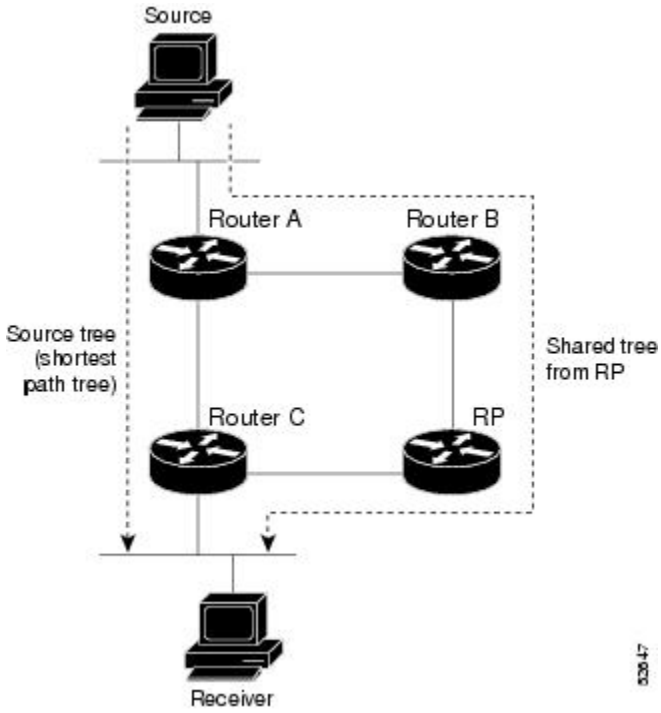
The traditional PIM protocols (dense-mode and sparse-mode) provided two models for forwarding multicast packets, source trees and shared trees. Source trees are rooted at the source of the traffic while shared trees are rooted at the rendezvous point. Source trees achieve the optimum path between each receiver and the source at the expense of additional routing information: an (S,G) routing entry per source in the multicast routing table. The shared tree provides a single distribution tree for all of the active sources. This means that traffic from different sources traverse the same distribution tree to reach the interested receivers, therefore reducing the amount of routing state in the network. This shared tree needs to be rooted somewhere, and the location of this root is the rendezvous point. PIM BIDIR uses shared trees as their main forwarding mechanism.

The algorithm to elect the designated forwarder is straightforward, all the PIM neighbors in a subnet advertise their unicast route to the rendezvous point and the router with the best route is elected. This effectively builds a shortest path between every subnet and the rendezvous point without consuming any multicast routing state (no (S,G) entries are generated). The designated forwarder election mechanism expects all of the PIM neighbors to be BIDIR enabled. In the case where one of more of the neighbors is not a BIDIR capable router, the election fails and BIDIR is disabled in that subnet.

PIM Shared Tree and Source Tree (Shortest Path Tree)

In PIM-SM, the rendezvous point (RP) is used to bridge sources sending data to a particular group with receivers sending joins for that group. In the initial setup of state, interested receivers receive data from senders to the group across a single data distribution tree rooted at the RP. This type of distribution tree is called a shared tree or rendezvous point tree (RPT) as illustrated in Figure 3: Shared Tree and Source Tree (Shortest Path Tree), on page 81 . Data from senders is delivered to the RP for distribution to group members joined to the shared tree.

Figure 3: Shared Tree and Source Tree (Shortest Path Tree)



Unless the **spt-threshold infinity** command is configured, this initial state gives way as soon as traffic is received on the leaf routers (designated router closest to the host receivers). When the leaf router receives traffic from the RP on the RPT, the router initiates a switch to a data distribution tree rooted at the source sending traffic. This type of distribution tree is called a **shortest path tree** or **source tree**. By default, the Cisco IOS XR Software switches to a source tree when it receives the first data packet from a source.

The following process describes the move from shared tree to source tree in more detail:

1. Receiver joins a group; leaf Router C sends a join message toward RP.
2. RP puts link to Router C in its outgoing interface list.
3. Source sends data; Router A encapsulates data in Register and sends it to RP.
4. RP forwards data down the shared tree to Router C and sends a join message toward Source. At this point, data may arrive twice at the RP, once encapsulated and once natively.
5. When data arrives natively (unencapsulated) at RP, RP sends a register-stop message to Router A.

6 By default, receipt of the first data packet prompts Router C to send a join message toward Source.

7 When Router C receives data on (S,G), it sends a prune message for Source up the shared tree.

8 RP deletes the link to Router C from outgoing interface of (S,G). RP triggers a prune message toward Source.

Join and prune messages are sent for sources and RPs. They are sent hop by hop and are processed by each PIM router along the path to the source or RP. Register and register-stop messages are not sent hop by hop. They are exchanged using direct unicast communication between the designated router that is directly connected to a source and the RP for the group.

Tip The **spt-threshold infinity** command lets you configure the router so that it never switches to the shortest path tree (SPT).

Multicast-Intact

The multicast-intact feature provides the ability to run multicast routing (PIM) when Interior Gateway Protocol (IGP) shortcuts are configured and active on the router. Both Open Shortest Path First, version 2 (OSPFv2), and Intermediate System-to-Intermediate System (IS-IS) support the multicast-intact feature. Multiprotocol Label Switching Traffic Engineering (MPLS-TE) and IP multicast coexistence is supported in Cisco IOS XR Software by using the **mpls traffic-eng multicast-intact** IS-IS or OSPF router command. See *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide* for information on configuring multicast intact using IS-IS and OSPF commands.

You can enable multicast-intact in the IGP when multicast routing protocols (PIM) are configured and IGP shortcuts are configured on the router. IGP shortcuts are MPLS tunnels that are exposed to IGP. The IGPs route the IP traffic over these tunnels to destinations that are downstream from the egress router of the tunnel (from an SPF perspective). PIM cannot use IGP shortcuts for propagating PIM joins because reverse path forwarding (RPF) cannot work across a unidirectional tunnel.

When you enable multicast-intact on an IGP, the IGP publishes a parallel or alternate set of equal-cost next-hops for use by PIM. These next-hops are called **mcast-intact next-hops**. The mcast-intact next-hops have the following attributes:

- They are guaranteed not to contain any IGP shortcuts.
- They are not used for unicast routing but are used only by PIM to look up an IPv4 next hop to a PIM source.
- They are not published to the Forwarding Information Base (FIB).
- When multicast-intact is enabled on an IGP, all IPv4 destinations that were learned through link-state advertisements are published with a set equal-cost mcast-intact next-hops to the RIB. This attribute applies even when the native next-hops have no IGP shortcuts.
- In IS-IS, the max-paths limit is applied by counting both the native and mcast-intact next-hops together. (In OSPFv2, the behavior is slightly different.)

Designated Routers

Cisco routers use PIM-SM to forward multicast traffic and follow an election process to select a designated router (DR) when there is more than one router on a LAN segment.

The designated router is responsible for sending PIM register and PIM join and prune messages toward the RP to inform it about host group membership.

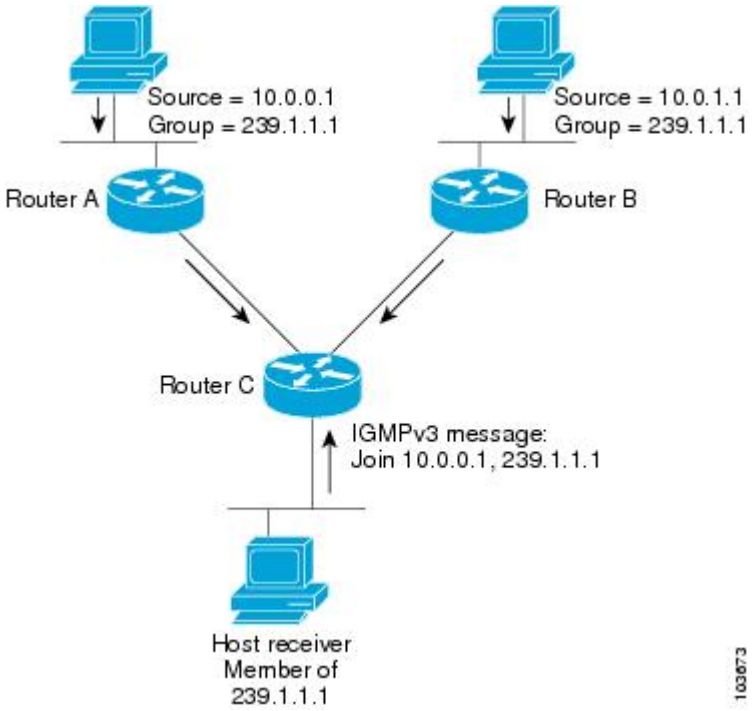
If there are multiple PIM-SM routers on a LAN, a designated router must be elected to avoid duplicating multicast traffic for connected hosts. The PIM router with the highest IP address becomes the DR for the LAN unless you choose to force the DR election by use of the **dr-priority** command. The DR priority option allows you to specify the DR priority of each router on the LAN segment (default priority = 1) so that the router with the highest priority is elected as the DR. If all routers on the LAN segment have the same priority, the highest IP address is again used as the tiebreaker.

Figure 4: Designated Router Election on a Multiaccess Segment, on page 84 illustrates what happens on a multiaccess segment. Router A (10.0.0.253) and Router B (10.0.0.251) are connected to a common multiaccess Ethernet segment with Host A (10.0.0.1) as an active receiver for Group A. As the Explicit Join model is used, only Router A, operating as the DR, sends joins to the RP to construct the shared tree for Group A. If Router B were also permitted to send (*, G) joins to the RP, parallel paths would be created and Host A would receive duplicate multicast traffic. When Host A begins to source multicast traffic to the group, the DR's responsibility is to send register messages to the RP. Again, if both routers were assigned the responsibility, the RP would receive duplicate multicast packets.

If the DR fails, the PIM-SM provides a way to detect the failure of Router A and to elect a failover DR. If the DR (Router A) were to become inoperable, Router B would detect this situation when its neighbor adjacency with Router A timed out. Because Router B has been hearing IGMP membership reports from Host A, it already has IGMP state for Group A on this interface and immediately sends a join to the RP when it becomes the new DR. This step reestablishes traffic flow down a new branch of the shared tree using Router B. Additionally, if Host A were sourcing traffic, Router B would initiate a new register process immediately after receiving the next multicast packet from Host A. This action would trigger the RP to join the SPT to Host A, using a new branch through Router B.

Tip Two PIM routers are neighbors if there is a direct connection between them. To display your PIM neighbors, use the **show pim neighbor** command in EXEC mode.

Figure 4: Designated Router Election on a Multiaccess Segment



Note DR election process is required only on multiaccess LANs. The last-hop router directly connected to the host is the DR.

Rendezvous Points

When PIM is configured in sparse mode, you must choose one or more routers to operate as a rendezvous point (RP). A rendezvous point is a single common root placed at a chosen point of a shared distribution tree, as illustrated in Figure 3: Shared Tree and Source Tree (Shortest Path Tree), on page 81. A rendezvous point can be either configured statically in each box or learned through a dynamic mechanism.

PIM DRs forward data from directly connected multicast sources to the rendezvous point for distribution down the shared tree. Data is forwarded to the rendezvous point in one of two ways:

- Encapsulated in register packets and unicast directly to the rendezvous point by the first-hop router operating as the DR
- Multicast forwarded by the RPF forwarding algorithm, described in the Reverse-Path Forwarding, on page 86, if the rendezvous point has itself joined the source tree.

The rendezvous point address is used by first-hop routers to send PIM register messages on behalf of a host sending a packet to the group. The rendezvous point address is also used by last-hop routers to send PIM join and prune messages to the rendezvous point to inform it about group membership. You must configure the rendezvous point address on all routers (including the rendezvous point router).

A PIM router can be a rendezvous point for more than one group. Only one rendezvous point address can be used at a time within a PIM domain. The conditions specified by the access list determine for which groups the router is a rendezvous point.

You can either manually configure a PIM router to function as a rendezvous point or allow the rendezvous point to learn group-to-RP mappings automatically by configuring Auto-RP or BSR. (For more information, see the Auto-RP, on page 85 section that follows and PIM Bootstrap Router, on page 86.)

Auto-RP

Automatic route processing (Auto-RP) is a feature that automates the distribution of group-to-RP mappings in a PIM network. This feature has these benefits:

- It is easy to use multiple RPs within a network to serve different group ranges.
- It allows load splitting among different RPs.
- It facilitates the arrangement of RPs according to the location of group participants.
- It avoids inconsistent, manual RP configurations that might cause connectivity problems.

Multiple RPs can be used to serve different group ranges or to serve as hot backups for each other. To ensure that Auto-RP functions, configure routers as candidate RPs so that they can announce their interest in operating as an RP for certain group ranges. Additionally, a router must be designated as an RP-mapping agent that receives the RP-announcement messages from the candidate RPs, and arbitrates conflicts. The RP-mapping agent sends the consistent group-to-RP mappings to all remaining routers. Thus, all routers automatically determine which RP to use for the groups they support.

Tip By default, if a given group address is covered by group-to-RP mappings from both static RP configuration, and is discovered using Auto-RP or PIM BSR, the Auto-RP or PIM BSR range is preferred. To override the default, and use only the RP mapping, use the **rp-address override** keyword.

Note If you configure PIM in sparse mode and do not configure Auto-RP, you must statically configure an RP as described in the Configuring a Static RP and Allowing Backward Compatibility, on page 124. When router interfaces are configured in sparse mode, Auto-RP can still be used if all routers are configured with a static RP address for the Auto-RP groups.

Note Auto-RP is not supported on VRF interfaces. Auto-RP Lite allows you to configure auto-RP on the CE router. It allows the PE router that has the VRF interface to relay auto-RP discovery, and announce messages across the core and eventually to the remote CE. Auto-RP is supported in only the IPv4 address family.

PIM Bootstrap Router

The PIM bootstrap router (BSR) provides a fault-tolerant, automated RP discovery and distribution mechanism that simplifies the Auto-RP process. This feature is enabled by default allowing routers to dynamically learn the group-to-RP mappings.

PIM uses the BSR to discover and announce RP-set information for each group prefix to all the routers in a PIM domain. This is the same function accomplished by Auto-RP, but the BSR is part of the PIM Version 2 specification. The BSR mechanism interoperates with Auto-RP on Cisco routers.

To avoid a single point of failure, you can configure several candidate BSRs in a PIM domain. A BSR is elected among the candidate BSRs automatically. Candidates use bootstrap messages to discover which BSR has the highest priority. The candidate with the highest priority sends an announcement to all PIM routers in the PIM domain that it is the BSR.

Routers that are configured as candidate RPs unicast to the BSR the group range for which they are responsible. The BSR includes this information in its bootstrap messages and disseminates it to all PIM routers in the domain. Based on this information, all routers are able to map multicast groups to specific RPs. As long as a router is receiving the bootstrap message, it has a current RP map.

Reverse-Path Forwarding

Reverse-path forwarding (RPF) is an algorithm used for forwarding multicast datagrams. It functions as follows:

- If a router receives a datagram on an interface it uses to send unicast packets to the source, the packet has arrived on the RPF interface.
- If the packet arrives on the RPF interface, a router forwards the packet out the interfaces present in the outgoing interface list of a multicast routing table entry.
- If the packet does not arrive on the RPF interface, the packet is silently discarded to prevent loops.

PIM uses both source trees and RP-rooted shared trees to forward datagrams; the RPF check is performed differently for each, as follows:

- If a PIM router has an (S,G) entry present in the multicast routing table (a source-tree state), the router performs the RPF check against the IP address of the source for the multicast packet.
- If a PIM router has no explicit source-tree state, this is considered a shared-tree state. The router performs the RPF check on the address of the RP, which is known when members join the group.

Sparse-mode PIM uses the RPF lookup function to determine where it needs to send joins and prunes. (S,G) joins (which are source-tree states) are sent toward the source. (*,G) joins (which are shared-tree states) are sent toward the RP.

Multicast VPN

Multicast VPN (MVPN) provides the ability to dynamically provide multicast support over MPLS networks. MVPN introduces an additional set of protocols and procedures that help enable a provider to support multicast traffic in a VPN.

Note PIM-Bidir is not supported on MVPN.

There are two ways MCAST VPN traffic can be transported over the core network:

- Rosen GRE (native): MVPN uses GRE with unique multicast distribution tree (MDT) forwarding to enable scalability of native IP Multicast in the core network. MVPN introduces multicast routing information to the VPN routing and forwarding table (VRF), creating a Multicast VRF. In Rosen GRE, the MCAST customer packets (c-packets) are encapsulated into the provider MCAST packets (p-packets), so that the PIM protocol is enabled in the provider core, and mrib/mfib is used for forwarding p-packets in the core.
- MLDP ones (Rosen, partition): MVPN allows a service provider to configure and support multicast traffic in an MPLS VPN environment. This type supports routing and forwarding of multicast packets for each individual VPN routing and forwarding (VRF) instance, and it also provides a mechanism to transport VPN multicast packets across the service provider backbone. In the MLDP case, the regular label switch path forwarding is used, so core does not need to run PIM protocol. In this scenario, the c-packets are encapsulated in the MPLS labels and forwarding is based on the MPLS Label Switched Paths (LSPs) ,similar to the unicast case.

In both the above types, the MVPN service allows you to build a Protocol Independent Multicast (PIM) domain that has sources and receivers located in different sites.

To provide Layer 3 multicast services to customers with multiple distributed sites, service providers look for a secure and scalable mechanism to transmit customer multicast traffic across the provider network. Multicast VPN (MVPN) provides such services over a shared service provider backbone, using native multicast technology similar to BGP/MPLS VPN.

In addition to all the ethernet based line cards, Multicast VPN is also supported on the Cisco ASR 9000 Series SPA Interface Processor-700 card from the Cisco IOS XR Software Release 4.0 onwards. Cisco ASR 9000 Series SPA Interface Processor-700 enables the Cisco ASR 9000 Series Routers to support multiple legacy services (such as TDM and ATM) on a router that is primarily designed for Ethernet networks. Cisco ASR 9000 Series SPA Interface Processor-700 is QFP-based and therefore has the flexibility and service scale offered by Cisco ASIC and the reliability of Cisco IOS XR Software.

MVPN emulates MPLS VPN technology in its adoption of the multicast domain (MD) concept, in which provider edge (PE) routers establish virtual PIM neighbor connections with other PE routers that are connected to the same customer VPN. These PE routers thereby form a secure, virtual multicast domain over the provider network. Multicast traffic is then transmitted across the core network from one site to another, as if the traffic were going through a dedicated provider network.

Multi-instance BGP is supported on multicast and MVPN. Multicast-related SAFIs can be configured on multiple BGP instances.

Multicast VPN Routing and Forwarding

Dedicated multicast routing and forwarding tables are created for each VPN to separate traffic in one VPN from traffic in another.

The VPN-specific multicast routing and forwarding database is referred to as **MVRF**. On a PE router, an MVRF is created when multicast is enabled for a VRF. Protocol Independent Multicast (PIM), and Internet Group Management Protocol (IGMP) protocols run in the context of MVRF, and all routes created by an MVRF protocol instance are associated with the corresponding MVRF. In addition to VRFs, which hold

VPN-specific protocol states, a PE router always has a global VRF instance, containing all routing and forwarding information for the provider network.

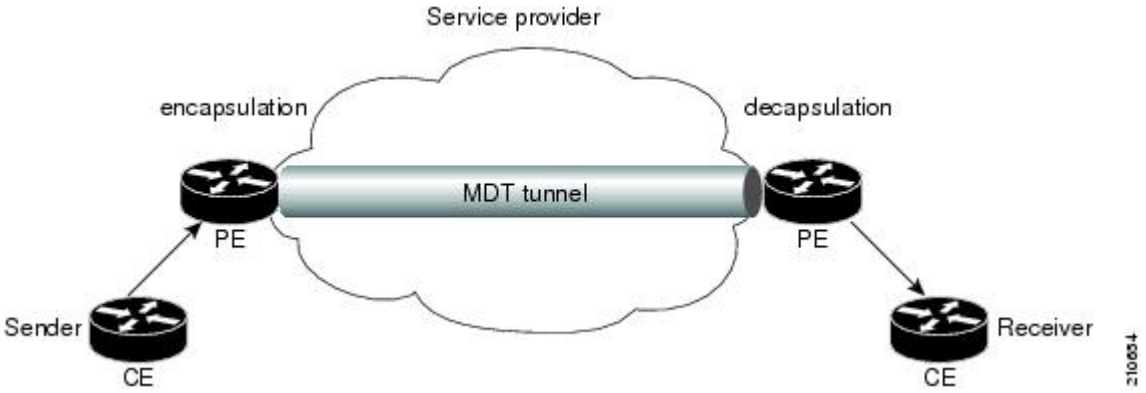
Multicast Distribution Tree Tunnels

The multicast distribution tree (MDT) can span multiple customer sites through provider networks, allowing traffic to flow from one source to multiple receivers. For MLDP, the MDT tunnel trees are called as Labeled MDT (LMDT).

Secure data transmission of multicast packets sent from the customer edge (CE) router at the ingress PE router is achieved by encapsulating the packets in a provider header and transmitting the packets across the core. At the egress PE router, the encapsulated packets are decapsulated and then sent to the CE receiving routers.

Multicast distribution tree (MDT) tunnels are point-to-multipoint. A MDT tunnel interface is an interface that MVRF uses to access the multicast domain. It can be deemed as a passage that connects an MVRF and the global MVRF. Packets sent to an MDT tunnel interface are received by multiple receiving routers. Packets sent to an MDT tunnel interface are encapsulated, and packets received from a MDT tunnel interface are decapsulated.

Figure 5: Virtual PIM Peer Connection over an MDT Tunnel Interface



Encapsulating multicast packets in a provider header allows PE routers to be kept unaware of the packets' origin—all VPN packets passing through the provider network are viewed as native multicast packets and are routed based on the routing information in the core network. To support MVPN, PE routers only need to support native multicast routing.

MVPN also supports optimized VPN traffic forwarding for high-bandwidth applications that have sparsely distributed receivers. A dedicated multicast group can be used to encapsulate packets from a specific source, and an optimized MDT can be created to send traffic only to PE routers connected to interested receivers. This is referred to as **data MDT**.

InterAS Support on Multicast VPN

The Multicast VPN Inter-AS Support feature enables service providers to provide multicast connectivity to VPN sites that span across multiple autonomous systems. This feature was added to MLDP profile that enables Multicast Distribution Trees (MDTs), used for Multicast VPNs (MVPNs), to 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.
- Intra-Provider Inter-AS—Multiple service providers that need to coordinate their networks to provide inter-AS support.

To establish a Multicast VPN between two autonomous systems, a MDT-default tunnel must be setup between the two PE routers. The PE routers accomplish this by joining the configured MDT-default group. This MDT-default group is configured on the PE router and is unique for each VPN. The PIM sends the join based on the mode of the groups, which can be PIM SSM, or sparse mode.

Note PIM-Bidir is not supported on MVPN.

Benefits of MVPN Inter-AS Support

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

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

InterAS Option A

InterAS Option A is the basic Multicast VPN configuration option. In this option, the PE router partially plays the Autonomous System Border Router (ASBR) role in each Autonomous System (AS). Such a PE router in each AS is directly connected through multiple VRF bearing subinterfaces. MPLS label distribution protocol need not run between these InterAS peering PE routers. However, an IGP or BGP protocol can be used for route distribution under the VRF.

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. 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. This provides better control over routing exchanges and security between the two networks. However, Option A is considered the least scalable of all the inter-AS connectivity options.

IPv6 Connectivity over MVPN

On the Cisco ASR 9000 Series Routers, in Cisco IOS XR Software starting Release 4.2.1, IPv6 connectivity is supported between customer sites over an IPv4-only core network with a default VRF. VPN PE routers interoperate between the two address families, with control and forwarding actions between IPv4-encapsulated

MDTs and IPv6 customer routes. IPv6 users can configure IPv6-over-IPv4 multicast VPN support through BGP.

In Cisco IOS XR Software, MVPNv6 can have a separate data mdt group configured, which can be different from MVPNv4. But both MVPNv6 and MVPNv4 must have the same default mdt group configured.

The configuration example below shows MVPNv6 data mdt :

```
vrf cisco-sjc1
  address-family ipv4
   mdt data 226.8.3.0/24 threshold 5
   mdt default ipv4 226.8.0.1
  !
  address-family ipv6
   mdt data 226.8.4.0/24 threshold 5
   mdt default ipv4 226.8.0.1
  !
```

BGP Requirements

PE routers are the only routers that need to be MVPN-aware and able to signal remote PEs with information regarding the MVPN. It is fundamental that all PE routers have a BGP relationship with each other, either directly or through a route reflector, because the PE routers use the BGP peering address information to derive the RPF PE peer within a given VRF.

PIM-SSM MDT tunnels cannot be set up without a configured BGP MDT address-family, because you establish the tunnels, using the BGP connector attribute.

See the Implementing BGP on Cisco IOS XR Software module of the *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide* for information on BGP support for Multicast VPN.

Multitopology Routing

Multitopology routing allows you to manipulate network traffic flow when desirable (for example, to broadcast duplicate video streams) to flow over non-overlapping paths.

At the core of multitopology routing technology is router space infrastructure (RSI). RSI manages the global configuration of routing tables. These tables are hierarchically organized into VRF tables under logical routers. By default, RSI creates tables for unicast and multicast for both IPv4 and IPv6 under the default VRF. Using multitopology routing, you can configure named topologies for the default VRF.

PIM uses a routing policy that supports matching on source or group address to select the topology in which to look up the reverse-path forwarding (RPF) path to the source. If you do not configure a policy, the existing behavior (to select a default table) remains in force.

Currently, IS-IS and PIM routing protocols alone support multitopology-enabled network.

Multicast VPN Extranet Routing

Multicast VPN (MVPN) extranet routing lets service providers distribute IP multicast content from one enterprise site to another across a multicast VRF. In other words, this feature provides capability to seamlessly hop VRF boundaries to distribute multicast content end to end.

Unicast extranet can be achieved simply by configuring matching route targets across VRFs. However, multicast extranet requires such configuration to resolve route lookups across VRFs in addition to the following:

- Maintain multicast topology maps across VRFs.
- Maintain multicast distribution trees to forward traffic across VRFs.

Information About Extranets

An extranet can be viewed as part of an enterprise intranet that is extended to users outside the enterprise. A VPN is used as a way to do business with other enterprises and with customers, such as selling products and maintaining strong business partnerships. An extranet is a VPN that connects to one or more corporate sites to external business partners or suppliers to securely share a designated part of the enterprise's business information or operations.

MVPN extranet routing can be used to solve such business problems as:

- Inefficient content distribution between enterprises.
- Inefficient content distribution from service providers or content providers to their enterprise VPN customers.

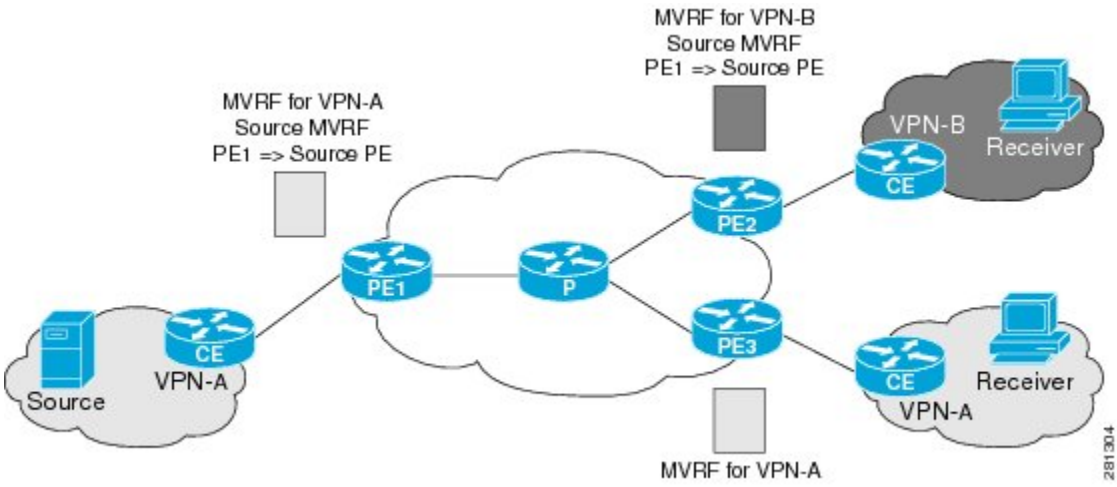
MVPN extranet routing provides support for IPv4 and IPv6 address family.

An extranet network requires the PE routers to pass traffic across VRFs (labeled "P" in Figure 6: Components of an Extranet MVPN, on page 91). Extranet networks can run either IPv4 or IPv6, but the core network always runs only IPv4 active multicast.

Note Multicast extranet routing is not supported on BVI interfaces.

Extranet Components

Figure 6: Components of 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.

Information About the Extranet MVPN Routing Topology

In unicast routing of peer-to-peer VPNs, BGP routing protocol is used to advertise VPN IPv4 and IPv6 customer routes between provider edge (PE) routers. However, in an MVPN extranet peer-to-peer network, PIM RPF is used to determine whether the RPF next hop is in the same or a different VRF and whether that source VRF is local or remote to the PE.

Source MVRF on a Receiver PE Router

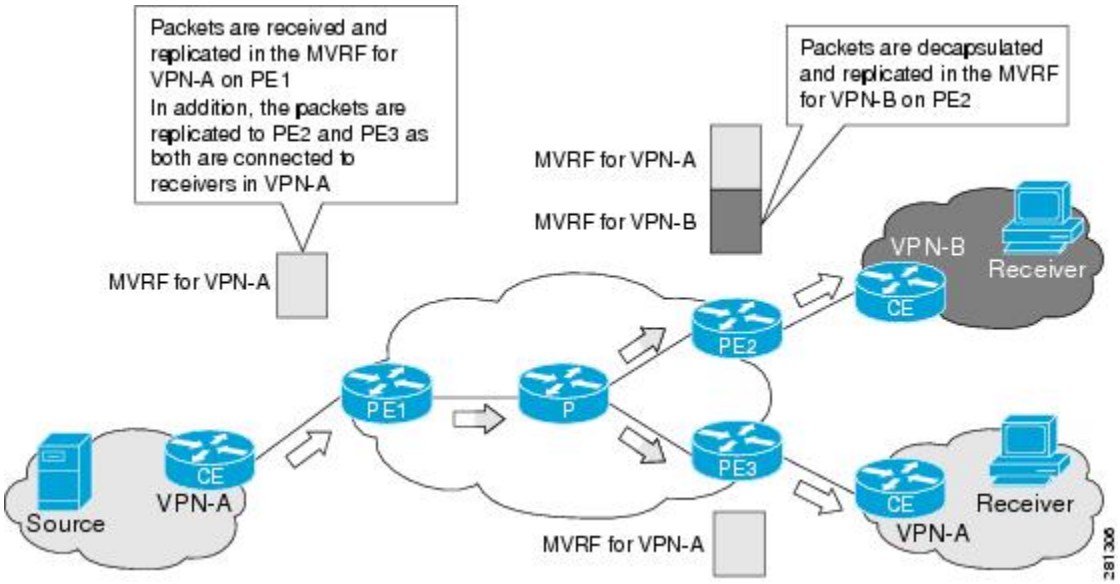
To provide extranet MVPN services to enterprise VPN customers by configuring a source MVRF on a receiver PE router, you would complete the following procedure:

- On a receiver PE router that has one or more interested receivers in an extranet site behind a directly connected CE router, configure an MVRF that has the same default MDT group as the site connected to the multicast source.
- On the receiver PE router, configure the same unicast routing policy to import routes from the source MVRF to the receiver MVRF.

If the originating MVRF of the RPF next hop is local (source MVRF at receiver PE router), the join state of the receiver VRFs propagates over the core by using the default multicast distribution tree (MDT) of the source VRF. Figure 7: Source MVRF at the Receiver PE Router, on page 93 illustrates the flow of multicast traffic in an extranet MVPN topology where the source MVRF is configured on a receiver PE router (source at receiver MVRF topology). An MVRF is configured for VPN-A and VPN-B 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-A, and there are interested receivers behind PE2, the receiver PE router for VPN-B, and also behind PE3, the receiver PE router for VPN-A. After PE1 receives the packets from the source in the MVRF for

VPN-A, it replicates and forwards the packets to PE2 and PE3. The packets received at PE2 in VPN-A are decapsulated and replicated to receivers in VPN-B.

Figure 7: Source MVRF at the Receiver PE Router



Receiver MVRF on the Source PE Router

To provide extranet MVPN services to enterprise VPN customers by configuring the receiver MVRF on the source PE router, complete the following procedure:

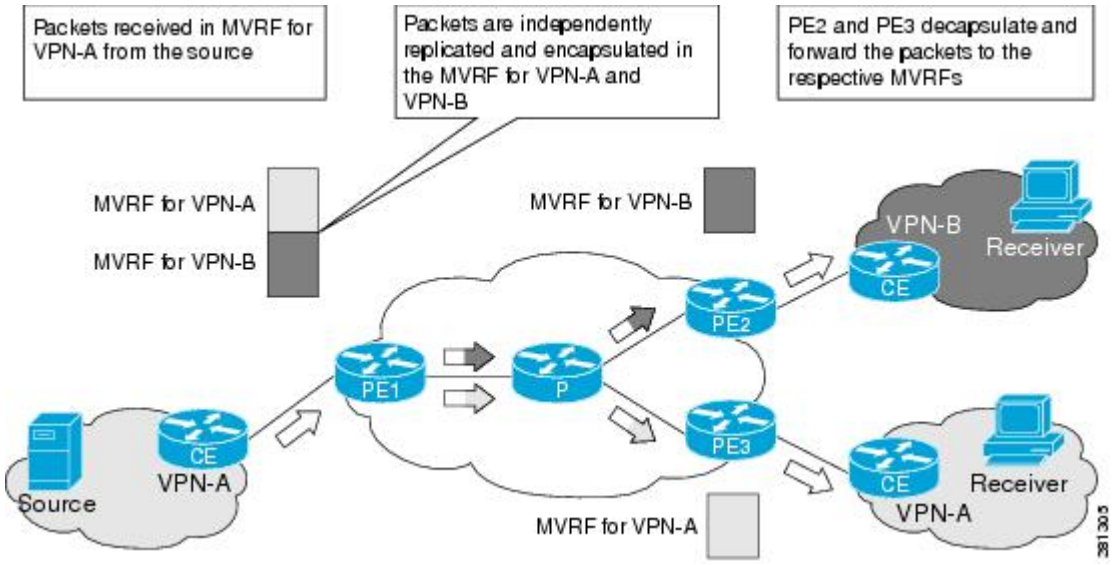
- For each extranet site, you would configure an additional MVRF on the source PE router, which has the same default MDT group as the receiver MVRF, if the MVRF is not already configured on the source PE.
- In the receiver MVRF configuration, you would configure the same unicast routing policy on the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

If the originating MVRF of the RPF next-hop is remote (receiver MVRF on the source PE router), then the join state of receiver VRFs propagates over the core through the MDT of each receiver.

Figure 8: Receiver MVRF at the Source PE Router Receiver, on page 94 illustrates the flow of multicast traffic in an extranet MVPN topology where a receiver MVRF is configured on the source PE router. An MVRF is configured for VPN-A and VPN-B on PE1, the source PE router. A multicast source behind PE1 is sending out a multicast stream to the MVRF for VPN-A, and there are interested receivers behind PE2 and PE3, the receiver PE routers for VPN-B and VPN-A, respectively. After PE1 receives the packets from the source in the MVRF for VPN-A, it independently replicates and encapsulates the packets in the MVRF for

VPN-A and VPN-B and forwards the packets. After receiving the packets from this source, PE2 and PE3 decapsulate and forward the packets to the respective MVRFs.

Figure 8: Receiver MVRF at the Source PE Router Receiver



For more information, see also Configuring MVPN Extranet Routing, on page 153 and Configuring MVPN Extranet Routing: Example, on page 204.

RPF Policies in an Extranet

RPF policies can be configured in receiver VRFs to bypass RPF lookup in receiver VRFs and statically propagate join states to specified source VRF. Such policies can be configured to pick a source VRF based on either multicast group range, multicast source range, or RP address.

For more information about configuration of RFP policies in extranets, see Configuring RPL Policies in Receiver VRFs to Propagate Joins to a Source VRF: Example, on page 206 and Configuring RPL Policies in Receiver VRFs on Source PE Routers to Propagate Joins to a Source VRF: Example, on page 209.

Multicast VPN Hub and Spoke Topology

Hub and spoke topology is an interconnection of two categories of sites — Hub sites and Spoke sites. The routes advertised across sites are such that they achieve connectivity in a restricted hub and spoke fashion. A spoke can interact only with its hub because the rest of the network (that is, other hubs and spokes) appears hidden behind the hub.

The hub and spoke topology can be adopted for these reasons:

- Spoke sites of a VPN customer receives all their traffic from a central (or Hub) site hosting services such as server farms.
- Spoke sites of a VPN customer requires all the connectivity between its spoke sites through a central site. This means that the hub site becomes a transit point for interspoke connectivity.

- Spoke sites of a VPN customer do not need any connectivity between spoke sites. Hubs can send and receive traffic from all sites but spoke sites can send or receive traffic only to or from Hub sites.

Realizing the Hub and Spoke Topology

Hub and Spoke implementation leverages the infrastructure built for MVPN Extranet. The regular MVPN follows the model in which packets can flow from any site to the other sites. But Hub and Spoke MVPN will restrict traffic flows based on their subscription.

A site can be considered to be a geographic location with a group of CE routers and other devices, such as server farms, connected to PE routers by PE-CE links for VPN access. Either every site can be placed in a separate VRF, or multiple sites can be combined in one VRF on the PE router.

By provisioning every site in a separate VRF, you can simplify the unicast and multicast Hub and Spoke implementation. Such a configuration brings natural protection from traffic leakage - from one spoke site to another. Cisco IOS XR Software implementation of hub and spoke follows the one- site-to-one VRF model. Any site can be designated as either a hub or spoke site, based on how the import or export of routes is setup. Multiple hub and spoke sites can be collated on a given PE router.

Unicast Hub and Spoke connectivity is achieved by the spoke sites importing routes from only Hub sites, and Hub sites importing routes from all sites. As the spoke sites do not exchange routes, spoke to spoke site traffic cannot flow. If interspoke connectivity is required, hubs can choose to re-inject routes learned from one spoke site into other spoke site.

MVPN Hub and Spoke is achieved by separating core tunnels, for traffic sourced from hub sites, and spoke sites. MDT hub is the tunnel carrying traffic sourced from all Hub sites, and MDT spoke carries traffic sourced from all spoke sites. Such tunnel end-points are configured on all PEs participating in hub and spoke topology. If spoke sites do not host any multicast sources or RPs, provisioning of MDT Spoke can be completely avoided at all such routers.

Once these tunnels are provisioned, multicast traffic path will be policy routed in this manner:

1. Hub sites will send traffic to only MDT Hub.
2. Spoke sites will send traffic to only MDT Spoke.
3. Hub sites will receive traffic from both tunnels.
4. Spoke sites will receive traffic from only MDT Hub.

These rules ensure that hubs and spokes can send and receive traffic to or from each other, but direct spoke to spoke communication does not exist. If required, interspoke multicast can flow by turning around the traffic at Hub sites.

These enhancements are made to the Multicast Hub and Spoke topology in Cisco IOS XR Software Release 4.0:

- Auto-RP and BSR are supported across VRFs that are connected through extranet. It is no longer restricted to using static RP only.
- MP-BGP can publish matching import route-targets while passing prefix nexthop information to RIB.
- Route policies can use extended community route targets instead of IP address ranges.
- Support for extranet v4 data mdt was included so that data mdt in hub and spoke can be implemented.

Label Switched Multicast (LSM) Multicast Label Distribution Protocol (mLDP) based Multicast VPN (mVPN) Support

Label Switch Multicast (LSM) is MPLS technology extensions to support multicast using label encapsulation. Next-generation MVPN is based on Multicast Label Distribution Protocol (mLDP), which can be used to build P2MP and MP2MP LSPs through a MPLS network. These LSPs can be used for transporting both IPv4 and IPv6 multicast packets, either in the global table or VPN context.

Benefits of LSM MLDP based MVPN

LSM provides these benefits when compared to GRE core tunnels that are currently used to transport customer traffic in the core:

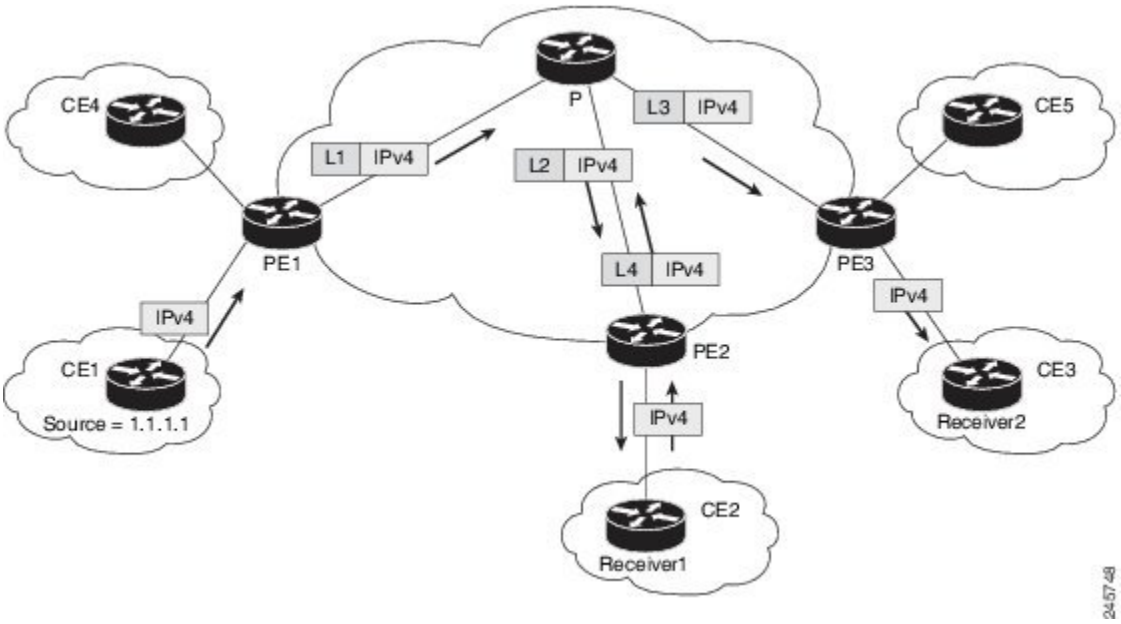
- It leverages the MPLS infrastructure for transporting IP multicast packets, providing a common data plane for unicast and multicast.
- It applies the benefits of MPLS to IP multicast such as Fast ReRoute (FRR) and
- It eliminates the complexity associated PIM.

Configuring MLDP MVPN

The MLDP MVPN configuration enables IPv4 multicast packet delivery using MPLS. This configuration uses MPLS labels to construct default and data Multicast Distribution Trees (MDTs). The MPLS replication is used as a forwarding mechanism in the core network. For MLDP MVPN configuration to work, ensure that the global MPLS MLDP configuration is enabled. To configure MVPN extranet support, configure the source

multicast VPN Routing and Forwarding (mVRF) on the receiver Provider Edge (PE) router or configure the receiver mVRF on the source PE. MLDP MVPN is supported for both intranet and extranet.

Figure 9: MLDP based MPLS Network



P2MP and MP2MP Label Switched Paths

mLDP is an application that sets up Multipoint Label Switched Paths (MP LSPs) in MPLS networks without requiring multicast routing protocols in the MPLS core. mLDP constructs the P2MP or MP2MP LSPs without interacting with or relying upon any other multicast tree construction protocol. Using LDP extensions for MP LSPs and Unicast IP routing, mLDP can setup MP LSPs. The two types of MP LSPs that can be setup are Point-to-Multipoint (P2MP) and Multipoint-to-Multipoint (MP2MP) type LSPs.

A P2MP LSP allows traffic from a single root (ingress node) to be delivered to a number of leaves (egress nodes), where each P2MP tree is uniquely identified with a 2-tuple (root node address, P2MP LSP identifier). A P2MP LSP consists of a single root node, zero or more transit nodes, and one or more leaf nodes, where typically root and leaf nodes are PEs and transit nodes are P routers. A P2MP LSP setup is receiver-driven and is signaled using mLDP P2MP FEC, where LSP identifier is represented by the MP Opaque Value element. MP Opaque Value carries information that is known to ingress LSRs and Leaf LSRs, but need not be interpreted by transit LSRs. There can be several MP LSPs rooted at a given ingress node, each with its own identifier.

A MP2MP LSP allows traffic from multiple ingress nodes to be delivered to multiple egress nodes, where a MP2MP tree is uniquely identified with a 2-tuple (root node address, MP2MP LSP identifier). For a MP2MP LSP, all egress nodes, except the sending node, receive a packet sent from an ingress node.

A MP2MP LSP is similar to a P2MP LSP, but each leaf node acts as both an ingress and egress node. To build an MP2MP LSP, you can setup a downstream path and an upstream path so that:

- Downstream path is setup just like a normal P2MP LSP
- Upstream path is setup like a P2P LSP towards the upstream router, but inherits the downstream labels from the downstream P2MP LSP.

Packet Flow in mLDP-based Multicast VPN

For each packet coming in, MPLS creates multiple out-labels. Packets from the source network are replicated along the path to the receiver network. The CE1 router sends out the native IP multicast traffic. The PE1 router imposes a label on the incoming multicast packet and replicates the labeled packet towards the MPLS core network. When the packet reaches the core router (P), the packet is replicated with the appropriate labels for the MP2MP default MDT or the P2MP data MDT and transported to all the egress PEs. Once the packet reaches the egress PE, the label is removed and the IP multicast packet is replicated onto the VRF interface.

Realizing a mLDP-based Multicast VPN

There are different ways a Label Switched Path (LSP) built by mLDP can be used depending on the requirement and nature of application such as:

- P2MP LSPs for global table transit Multicast using in-band signaling.
- P2MP/MP2MP LSPs for MVPN based on MI-PMSI or Multidirectional Inclusive Provider Multicast Service Instance (Rosen Draft).
- P2MP/MP2MP LSPs for MVPN based on MS-PMSI or Multidirectional Selective Provider Multicast Service Instance (Partitioned E-LAN).

The Cisco ASR 9000 Series Router performs the following important functions for the implementation of MLDP:

1. Encapsulating VRF multicast IP packet with GRE/Label and replicating to core interfaces (imposition node).
2. Replicating multicast label packets to different interfaces with different labels (Mid node).
3. Decapsulate and replicate label packets into VRF interfaces (Disposition node).

Characteristics of mLDP Profiles

The characteristics of various mLDP profiles are listed in this section.

Profile 1:Rosen-mLDP (with no BGP-AD)

These are the characteristics of this profile:

- MP2MP mLDP trees are used in the core.
- VPN-ID is used as the VRF distinguisher.
- Configuration based on Default MDTs.
- Same Default-MDT core-tree used for IPv4 and IPv6 traffic.
- Data-MDT announcements sent by PIM (over Default-MDT).
- The multicast traffic can either be SM or SSM.
- Inter-AS Options A, B, and C are supported. Connector Attribute is announced in VPN-IP routes.

Profile 2:MS-PMSI-mLDP-MP2MP (No BGP-AD)

These are the characteristics of this profile:

- MP2MP mLDP trees are used in the core.
- Different MS-PMSI core-trees for IPv4 and IPv6 traffic.
- The multicast traffic can be SM or SSM.
- Extranet, Hub and Spoke are supported.
- Inter-AS Options A, B, and C are supported. Connector Attribute is announced in VPN-IP routes.

Profile 3:Rosen-GRE with BGP-AD

These are the characteristics of this profile:

- PIM-trees are used in the core. The data encapsulation method used is GRE.
- SM or SSM used in the core.
- Configuration is based on Default-MDTs.
- The multicast traffic can be SM or SSM.
- MoFRR in the core is supported.
- Extranet, Hub and Spoke, CsC, Customer-RP-discovery (Embedded-RP, AutoRP and BSR) are supported.
- Inter-AS Options A, B, and C are supported. VRF-Route-Import EC is announced in VPN-IP routes.

Profile 4: MS-PMSI-mLDP-MP2MP with BGP-AD

These are the characteristics of this profile:

- MP2MP mLDP trees are used in the core.
- The multicast traffic can be SM or SSM.
- Extranet, Hub and Spoke, CsC, Customer-RP-discovery (Embedded-RP, AutoRP, and BSR) are supported.
- Inter-AS Options A, B, and C are supported. VRF-Route-Import EC is announced in VPN-IP routes.

Profile 5: MS-PMSI-mLDP-P2MP with BGP-AD

These are the characteristics of this profile:

- P2MP mLDP trees are used in the core.
- The multicast traffic can be SM or SSM.
- Extranet, Hub and Spoke, CsC, Customer-RP-discovery (Embedded-RP, AutoRP and BSR) are supported.
- Inter-AS Options A, B, and C are supported. . VRF-Route-Import EC is announced in VPN-IP routes.

Profile 6: VRF In-band Signaling (No BGP-AD)

These are the characteristics of this profile:

- P2MP mLDP trees are used in the core.

- MoFRR in the core is supported.
- There is one core tree built per VRF-S,G route. There can be no (*,G) routes in VRF, with RPF reachability over the core.
- The multicast traffic can be SM S,G or SSM.

Profile 7: Global Inband Signalling

These are the characteristics of this profile:

- P2MP mLDP inband tree in the core; no C-multicast Routing.
- Customer traffic can be SM S,G or SSM.
- Support for global table S,Gs on PEs.

For more information on MLDP implementation and OAM concepts, see the Cisco IOS XR MPLS Configuration Guide for the Cisco ASR 9000 Series Router

Profile 8: Global P2MP-TE

These are the characteristics of this profile:

- P2MP-TE tree, with static Destination list, in the core; no C-multicast Routing.
- Static config of (S,G) required on Head-end PE.
- Only C-SSM support on PEs.
- Support for global table S,Gs on PEs.

Profile 9: Rosen-mLDP with BGP-AD

These are the characteristics of this profile:

- Single MP2MP mLDP core-tree as the Default-MDT, with PIM C-multicast Routing.
- All UMH options supported.
- Default and Data MDT supported.
- Customer traffic can be SM or SSM .
- RIB-Extranet, RPL-Extranet, Hub & Spoke supported.
- Customer-RP-discovery (Embedded-RP, AutoRP & BSR) is supported.

Profile 10 : VRF Static-P2MP-TE with BGP AD

These are the characteristics of this profile:

- P2MP-TE tree, with static Destination list, in the core; no C-multicast Routing.
- Static config of (S,G) required on Head-end PE.
- Only C-SSM support on PEs.
- Support for IPv4 MVPN S,Gs on PEs. No support for IPv6 MVPN routes.

Profile 11 : Rosen-PIM/GRE with BGP C-multicast Routing

These are the characteristics of this profile:

- PIM-trees in the core, data encapsulation is GRE, BGP C-multicast Routing.
- Static config of (S,G) required on Head-end PE.
- For PIM-SSM core-tree and PIM-SM core-tree with no spt-infinity, all UMH options are supported.
- For PIM-SM core-tree with spt-infinity case, only SFS (Highest PE or Hash-of-BGP-paths) is supported. Hash of installed-paths method is not supported.
- Default and Data MDTs supported.
- Customer traffic can be SM or SSM .
- Inter-AS Option A supported. Options B and C not supported.
- All PEs must have a unique BGP Route Distinguisher (RD) value. To configure BGP RD value, refer *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide* .

Profile 12 : Rosen-mLDP-P2MP with BGP C-multicast Routing

These are the characteristics of this profile:

- Full mesh of P2MP mLDP core-tree as the Default-MDT, with BGP C-multicast Routing.
- All UMH options supported.
- Default and Data MDT supported.
- Customer traffic can be SM or SSM .
- RPL-Tail-end-Extranet supported.
- Inter-AS Option A, B and C supported.
- All PEs must have a unique BGP Route Distinguisher (RD) value. To configure BGP RD value, refer *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide* .

Profile 13 : Rosen-mLDP-MP2MP with BGP C-multicast Routing

These are the characteristics of this profile:

- Single MP2MP mLDP core-tree as the Default-MDT, with BGP C-multicast Routing.
- Only SFS (Highest PE or Hash-of-BGP-paths) is supported. Hash of Installed-paths method is not supported.
- Default and Data MDT supported.
- Customer traffic can be SM or SSM .
- RIB-Tail-end-Extranet, RPL-Tail-end-Extranet supported.
- Customer-RP-discovery (Embedded-RP, AutoRP & BSR) is supported.
- Inter-AS Option A, B and C supported. For Options B and C, Root has to be on a PE or the roor-address reachability has to be leaked across all autonomous systems.

- All PEs must have a unique BGP Route Distinguisher (RD) value. To configure BGP RD value, refer *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide* .

Profile 14 : MP2MP-mLDP-P2MP with BGP C-multicast Routing

These are the characteristics of this profile:

- Full mesh of P2MP mLDP core-tree as the Default-MDT, with BGP C-multicast Routing.
- All UMH options supported.
- Default and Data MDT supported.
- Customer traffic can be SM or SSM .
- RPL-Tail-end-Extranet supported.
- Customer-RP-discovery (Embedded-RP, AutoRP & BSR) is supported.
- Inter-AS Option A, B and C supported.
- All PEs must have a unique BGP Route Distinguisher (RD) value. To configure BGP RD value, refer *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide* .

Profile 15 : MP2MP-mLDP-MP2MP with BGP C-multicast Routing

These are the characteristics of this profile:

- Full mesh of MP2MP mLDP core-tree as the Default-MDT, with BGP C-multicast Routing.
- All UMH options supported.
- Default and Data MDT supported.
- Customer traffic can be SM or SSM .
- RPL-Tail-end-Extranet supported.
- Customer-RP-discovery (Embedded-RP, AutoRP & BSR) is supported.
- Inter-AS Option A, B and C supported.
- All PEs must have a unique BGP Route Distinguisher (RD) value. To configure BGP RD value, refer *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide* .

Profile 16 : Rosen-Static-P2MP-TE with BGP C-multicast Routing

These are the characteristics of this profile:

- Full mesh of Static-P2MP-TE core-trees, as the Default-MDT, with BGP C-multicast Routing.
- All UMH options supported.
- Support for Data MDT, Default MDT.
- Customer traffic can be SM, SSM .
- RPL-Tail-end-Extranet supported.
- Customer-RP-discovery (Embedded-RP, AutoRP & BSR) is supported.

- Inter-AS Option A supported. Options B and C not supported.
- All PEs must have a unique BGP Route Distinguisher (RD) value. To configure BGP RD value, refer *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide* .

Note Whenever multicast stream crosses configured threshold on encap PE(Head PE), S-PMSI is announced. Core tunnel is static P2MP-TE tunnel configured under route-policy for the stream. Static P2MP-TE data mdt is implemented in such a way that it can work with dynamic data mdt, dynamic default mdtand default static P2MP.

Profile 17: Rosen-mLDP-P2MP with BGP AD/PIM C-multicast Routing

These are the characteristics of this profile:

- Full mesh of P2MP mLDP core-tree as the Default-MDT, with PIM C-multicast Routing.
- All UMH options supported.
- Default and Data MDT supported.
- Customer traffic can be SM or SSM .
- RPL-Extranet, Hub & Spoke supported.
- Customer-RP-discovery (Embedded-RP, AutoRP & BSR) is supported.
- Inter-AS Option A, B and C supported.

Profile 18 : Rosen-Static-P2MP-TE with BGP AD/PIM C-multicast Routing

These are the characteristics of this profile:

- Full mesh of Static-P2MP-TE core-trees, as the Default-MDT, with PIM C-multicast Routing.
- All UMH options supported.
- Default MDT supported; Data MDT is not supported.
- Customer traffic can be SM, SSM .
- RPL-Extranet, Hub & Spoke supported.
- Customer-RP-discovery (Embedded-RP, AutoRP & BSR) is supported.
- Inter-AS Option A supported. Options B and C not supported.

Profile 20 : Rosen-P2MP-TE with BGP AD/PIM C-multicast Routing

These are the characteristics of this profile:

- Dynamic P2MP-TE tunnels setup on demand, with PIM C-multicast Routing
- All UMH options supported.
- Default and Data MDT supported.
- Customer traffic can be SM, SSM .

- RPL-Extranet, Hub & Spoke supported.
- Customer-RP-discovery (Embedded-RP, AutoRP & BSR) is supported.
- Inter-AS Option A and C- supported.

Profile 22 : Rosen-P2MP-TE with BGP C-multicast Routing

These are the characteristics of this profile:

- Dynamic P2MP-TE tunnels with BGP C-multicast Routing
- All UMH options supported.
- Default and Data MDT supported.
- Customer traffic can be SM or SSM .
- RIB-Tail-end-Extranet, RPL-Tail-end-Extranet supported.
- Customer-RP-discovery (Embedded-RP, AutoRP & BSR) is supported.
- Inter-AS Option A and C- supported.
- All PEs must have a unique BGP Route Distinguisher (RD) value. To configure BGP RD value, refer *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide* .

Profile 24: Partitioned-P2MP-TE with BGP AD/PIM C-multicast Routing

These are the characteristics of this profile:

- Dynamic P2MP-TE tunnels setup on demand, with PIM C-multicast Routing
- All UMH options supported.
- Default and Data MDT supported.
- Customer traffic can be SM or SSM .
- RPL-Extranet, Hub & Spoke supported.
- Customer-RP-discovery (Embedded-RP, AutoRP & BSR) is supported.
- Inter-AS Option A and C- supported.

Profile 26 : Partitioned-P2MP-TE with BGP C-multicast Routing

These are the characteristics of this profile:

- Dynamic P2MP-TE tunnels with BGP C-multicast Routing
- All UMH options supported.
- Default and Data MDT supported.
- Customer traffic can be SM, SSM.
- RIB-Tail-end-Extranet, RPL-Tail-end-Extranet supported.
- Customer-RP-discovery (Embedded-RP, AutoRP & BSR) is supported.
- Inter-AS Option A and C- supported.

- All PEs must have a unique BGP Route Distinguisher (RD) value. To configure BGP RD value, refer *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide* .

Configuration rules for profiles

Rules for Rosen-mGRE profiles (profiles- 0, 3, 11)

- All profiles require VPNv4 or v6 unicast reachability.
- By default, encap 1400-byte size c-multicast IP packet is supported. To support decap or encap larger packet size, **mdt mtu** command.
- Loopback configuration is required. Use the **mdt source loopback0** command. Other loopbacks can be used for different VRFs, but this is not recommended.

Rules for Rosen-mLDP profiles (profiles- 1, 9, 12, 13, 17)

- mLDP must be globally enabled.
- VPN-id is mandatory for Rosen-mLDP MP2MP profiles.
- Root node must be specified manually. Multiple root nodes can be configured for Root Node Redundancy.
- If only profile 1 is configured, MVPN must be enabled under bgp.
- For BGP-AD profiles, the remote PE address is required.

Rules for mLDP profiles (profiles- 2, 4, 5, 14, 15)

- MVPN must be enabled under bgp, if only profile 2 is configured.
- Support only for static RP for customer RP.

Rules for inband mLDP profiles (profiles- 6, 7)

- MVPN must be enabled under bgp for vrf-inband profiles.
- Data MDT is not supported.
- Backbone facing interface (BFI) must be enabled on tail PE.
- Source route of SSM must be advertise to tail PE by iBGP.

MLDP inband signaling

MLDP Inband signaling allows the core to create (S,G) or (*,G) state without using out-of-band signaling such as BGP or PIM. It is supported in VRF (and in the global context). Both IPv4 and IPv6 multicast groups are supported.

In MLDP Inband signaling, one can configure an ACL range of multicast (S,G). This (S,G) can be transported in MLDP LSP. Each multicast channel (S,G), is 1 to 1 mapped to each tree in the inband tree. The (S,G) join, through IGMP/MLD/PIM, will be registered in MRIB, which is the client of MLDP.

MLDP In-band signalling supports transiting PIM (S,G) or (*,G) trees across a MPLS core without the need for an out-of-band protocol. In-band signaling is only supported for shared-tree-only forwarding (also known as sparse-mode threshold infinity). PIM Sparse-mode behavior is not supported (switching from (*,G) to (S,G).

The details of the MLDP profiles are discussed in the *Cisco ASR 9000 Series Aggregation Services Router Multicast Configuration Guide*

Summary of Supported MVPN Profiles

This tables summarizes the supported MVPN profiles:

Profile Number	Name	Opaque-value	BGP-AD	Data-MDT
0	Rosen GRE	N/A	N/A	PIM TLVs over default MDT
1	Rosen MLDP	Type 2 - Root Address:VPN-ID:0-n	N/A	PIM TLVs over default MDT
2	MS- PMSI (Partition) MLDP MP2MP	Cisco proprietary - Source- PE:RD:0	N/A	N/A
3	Rosen GRE with BGP -AD	N/A	• Intra-AS MI- PMSI • S- PMSI for Data-MDT	PIM or BGP -AD (knob controlled)
4	MS- PMSI (Partition) MLDP MP2MP with BGP -AD	Type 1 - Source-PE:Global -ID	• I- PMSI with empty PTA • MS- PMSI for partition mdt • S- PMSI for data-mdt • S- PMSI cust RP-discovery trees	BGP-AD
5	MS- PMSI (Partition) MLDP P2MP with BGP -AD	Type 1 - Source-PE:Global -ID	• I- PMSI with empty PTA • MS- PMSI for partition mdt • S- PMSI for data-mdt • S- PMSI cust RP-discovery trees	BGP-AD
6	VRF Inband MLDP	RD:S,G	N/A	N/A
7	Global Inband	S,G	N/A	N/A
8	Global P2MP TE	N/A	N/A	N/A

Profile Number	Name	Opaque-value	BGP-AD	Data-MDT
9	Rosen MLDP with BGP -AD	Type 2 - RootAddresss:VPN - ID:0 -n	• Intra-AS MI- PMSI • S- PMSI for Data-MDT	PIM or BGP-AD (knob controlled)

LSP-switch for P2MP-TE

Turnaround for P2MP-TE can be handled by LSP-switch with a partitioned profile. For partitioned profiles, there is no core tree (where all the PEs join). When the traffic arrives at the ingress PE, it is forwarded to the RP-PE on a LSP. The RP-PE must then switch the traffic to a different LSP for all the non-RP PE receivers.

Configuration Process for MLDP MVPN (Intranet)

These steps provide a broad outline of the different configuration process of MLDP MVPN for intranet:

Note For detailed summary of the various MVPN profiles, see the Summary of Supported MVPN Profiles, on page 106.

- Enabling MPLS MLDP
  - configure
  - mpls ldp mldp
- Configuring a VRF entry
  - configure
  - vrf vrf_name
  - address-family ipv4/ipv6 unicast
  - import route-target route-target-ext-community
  - export route-target route-target-ext-community
- Configuring VPN ID
  - configure
  - vrf *vrf_name*
  - vpn id *vpn_id*

  The configuring VPN ID procedure is needed for profiles 1 and 9 (Rosen MLDP).
- Configuring MVPN Routing and Forwarding instance

- configure
- multicast-routing vrf *vrf_name*
- address-family ipv4
- mdt default mldp ipv4 *root-node*

For profile 1 (MLDP Rosen), the **mdt default mldp ipv4** command and for profile 4/5 (MS- PMSI with BGP-AD), the **mdt partitioned mldp ipv4 mp2mp/p2mp** command are configured.

- Configuring the Route Distinguisher
  - configure
  - router bgp *AS Number*
  - vrf *vrf_name*
  - rd *rd_value*
- Configuring Data MDTs (optional)
  - configure
  - multicast-routing vrf *vrf_name*
  - address-family ipv4
  - mdt data <1-255>
- Configuring BGP MDT address family
  - configure
  - router bgp *AS Number*
  - address-family ipv4 mdt
- Configuring BGP vpnv4 address family
  - configure
  - router bgp *AS Number*
  - address-family vpnv4 unicast
- Configuring BGP IPv4 VRF address family
  - configure
  - router bgp *AS Number*
  - vrf *vrf_name*
  - address-family ipv4 unicast
- Configuring PIM SM/SSM Mode for the VRFs
  - configure

- router pim
- vrf *vrf_name*
- address-family ipv4
- rpf topology route-policy *rosen_mvpn_mldp*

For each profile, a different route-policy is configured.

- Configuring route-policy
  - route-policy *rosen_mvpn_mldp*
  - set core-tree *tree-type*
  - pass
  - end-policy

For profile 1 (MLDP Rosen), the mldp-rosen core tree type and for profile 4/5 (MS- PMSI with BGP-AD), the mldp-partitioned-mp2mp/p2mp core tree type are configured.

Note The configuration of the above procedures depends on the profile used for each configuration. For detailed examples of each profile, see Configuring LSM based MLDP: Examples, on page 225.

Next-Generation Multicast VPN

Next-Generation Multicast VPN (NG-MVPN) offers more scalability for Layer 3 VPN multicast traffic. It allows point-to-multipoint Label Switched Paths (LSP) to be used to transport the multicast traffic between PEs, thus allowing the multicast traffic and the unicast traffic to benefit from the advantages of MPLS transport, such as traffic engineering and fast re-route. This technology is ideal for video transport as well as offering multicast service to customers of the layer 3 VPN service.

NG-MVPN supports:

- VRF Route-Import and Source-AS Extended Communities
- Upstream Multicast Hop (UMH) and Duplicate Avoidance
- Leaf AD (Type-4) and Source-Active (Type-5) BGP AD messages
- Default-MDT with mLDP P2MP trees and with Static P2MP-TE tunnels
- BGP C-multicast Routing
- RIB-based Extranet with BGP AD
- Accepting (*,G) S-PMSI announcements
- Egress-PE functionality for Ingress Replication (IR) core-trees
- Enhancements for PIM C-multicast Routing
- Migration of C-multicast Routing protocol

- PE-PE ingress replication
- Dynamic P2MP-TE tunnels
- Flexible allocation of P2MP-TE attribute-sets
- Data and partitioned MDT knobs
- Multi-instance BGP support
- SAFI-129 and VRF SAFI-2 support
- Anycast-RP using MVPN SAFI

PE-PE Ingress Replication

The ingress PE replicates a C-multicast data packet belonging to a particular MVPN and sends a copy to all or a subset of the PEs that belong to the MVPN. A copy of the packet is tunneled to a remote PE over a Unicast Tunnel to the remote PE.

IR-MDT represents a tunnel that uses IR as the forwarding method. It is usually, one IR-MDT per VRF, with multiple labeled switch paths (LSP) under the tunnel.

When PIM learns of Joins over the MDT (using either PIM or BGP C-multicast Routing), it downloads IP S,G routes to the VRF table in MRIB, with IR-MDT forwarding interfaces. Each IR-MDT forwarding interface has a LSM-ID allocated by PIM. Currently, LSM-ID is managed by mLDP and can range from 0 to 0xFFFFF (20-bits). For IR, the LSM-ID space is partitioned between mLDP and IR. For IR tunnels, the top (20th) bit is always be set, leading to a range of 0x80000 to 0xFFFFF. mLDP's limit is 0 to 0x7FFFF.

MVPN over GRE

A unicast GRE tunnel could be the accepting or forwarding interface for either a mVPN-GRE VRF route or a core route. When multicast packets arrive on the VRF interface with the intent of crossing the core, they are first encapsulated with a multicast GRE header (S,G) which are applicable to the VRF's MDT. Then, before the packets are actually forwarded , they are encapsulated in a unicast GRE header. The (S,D) in this packet are the origination and termination addresses for the unicast GRE tunnel.

GRE tunnel stitching is when both the accepting and forwarding interfaces are unicast GRE tunnels. Here, the packet has two GRE encaps. The outer encap is the unicast header for the GRE tunnel. The inner encap is the multicast GRE header for the MDT. This is called as double encap. There is a loss in terms of both bandwidth and throughput efficiency. The bandwidth efficiency loss is because 48 bytes of encap headers are being added to the original (VRF) packet. The throughput efficiency loss is the result of the processing time required to apply two encaps.

For the mVPN-GRE, if the VRF interface is a GRE tunnel, the protocol packets received from LPTS will be accompanied with the receiving unicast GRE tunnel interface and the VRF id of the VRF in which the GRE tunnel is configured. Thus VRF specific processing can be done on the packet.

Restrictions

- MVPN over GRE is supported only on ASR 9000 Enhanced Ethernet LCs.

Native Multicast

GRE tunneling provides a method to transport native multicast traffic across a non-Multicast enabled IP network. Once the multicast traffic in encapsulated with GRE, it appears as an IP packet to the core transport network.

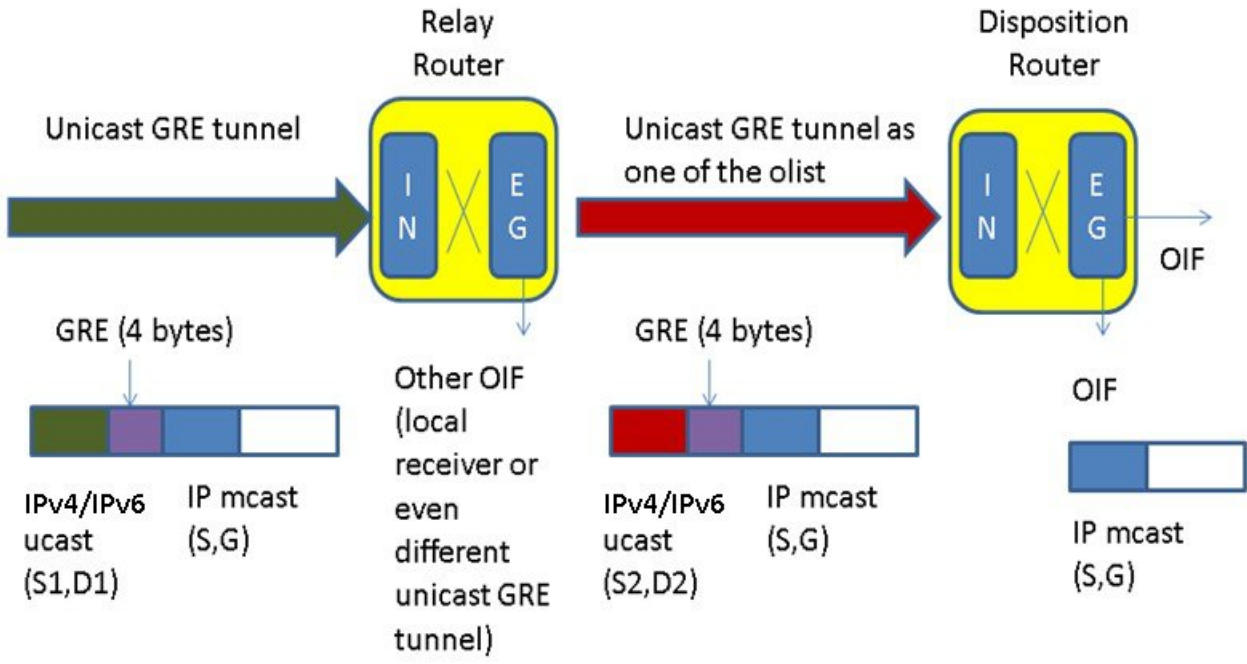
A GRE tunnel can be a forwarding interface when the router is the imposition (or encap) router for that GRE tunnel. The imposition router must prepend a unicast IPv4 header and GRE header to the multicast packet. The source and destination IPv4 addresses for the added header are determined by the user configuration of the tunnel. The newly encapsulated packet is then forwarded as a unicast packet.

When a GRE tunnel is an accepting interface for a multicast route, the router is the disposition (or decap) router for the tunnel. The outer IPv4 header and GRE header must be removed to expose the inner multicast packet. The multicast packet will then be forwarded just as any other multicast packet that arrives on a non-tunnel interface.

Forwarding behavior

Figure depicts a Unicast GRE tunnel between two routers. The imposition router has a multicast (S,G) route which has the GRE tunnel as a forwarding interface. At the disposition router, the GRE tunnel is an accepting interface for the multicast (S,G). As seen, the packet is unicast GRE encapsulated when it traverses the tunnel.

Figure 10: Unicast GRE tunnel between two routers



Note Starting with IOS XR 5.3.2 release, IPv6 traffic is supported.

GRE Limitations

Listed below are the limitations for unicast GRE tunnels:

- GRE unicast tunnel supports IPv4 encapsulation only.

Note Starting from the IOS XR 5.3.2 release, GRE unicast tunnels support IPv6 encapsulation.

- Native and mVPN traffic over underlying ECMP links are not supported.

Note Starting with IOS XR 5.3.2 release, native and mVPN traffic over underlying ECMP links, including bundles is supported.

- IPv6 multicast for GRE unicast tunnels is not supported, in releases prior to IOS XR 5.3.2.
- Transport header support is limited to IPv4.
- Path MTU discovery will not be supported over GRE tunnel interfaces. When size of the packet going over GRE tunnel interface exceeds the tunnel MTU, the microcode will punt the packet to the slow path for best effort fragmentation. Since punted packets are policed, this doesn't provide real fragmentation support. This combined with no support for path MTU discovery means that user is responsible for making sure the MTUs configured along the tunnel path are large enough to guarantee the GRE packet will not be fragmented between tunnel source and destination routers.
- No support for optional checksum, key, and sequence number fields.
- No support for nested and concatenated GRE tunnels. If packets with nested GRE header are received they will be dropped.
- No L3 features (like QoS, ACL and netflow) support for GRE tunnel interfaces. Features configured on the underlying physical interface will be applied.
- ASR9000 SIP-700 linecard unicast GRE is NOT supported on VRFs.
- Support for up to 500 GRE tunnels per system for multicast.

Signaling and RPF on GRE Tunnels

Signaling will use the same mechanism when a unicast GRE tunnel terminated at an ingress linecard regardless of whether the GRE tunnel interface belongs to a VRF or not. In the case of mVPN-GRE the Master Linecard / Master NP mechanism must still be used for egress punts of decapsulated VRF packets.

RPF selection can be static configured via a route policy configuration. Static RPF is more preferred and expected if the RPF should be the GRE tunnel. RPF may be selected dynamically via RIB updates for the upstream router’s unicast reach-ability, although this is not preferred.

PIM Registration

PIM registration packets can be forwarded on a unicast GRE tunnel as long as the IPv4 unicast GRE interface is selected by FIB for unicast forwarding of the encapsulated PIM registration packets toward the PIM RP. In this case, the packet is essentially double encapsulated with unicast, ie, the original multicast packet is encapsulated by PIM in a unicast PIM register packet. This is then encapsulated with the unicast GRE tunnel header.

At the PIM RP, outermost unicast header will be removed and the PIM registration packets will be delivered to PIM via LPTS as in the current PIM registration packet processing. It is advisable to avoid any MTU/TTL or ACL/QoS configuration issues that result in the registration packets getting dropped.

Auto-RP

Auto-RP lite on PEs, Auto-RP/BSR/static-RP/ Anycast-RP with MSDP peering etc can be supported over GRE tunnels with MFIB netio chain support. It is advisable to avoid any MTU/TTL or ACL/QoS configuration issues that result in the registration packets getting dropped. Auto-RP routes will flood autp-rp packets to every multicast egress interface including IPv4 unicast GRE tunnels.

Multicast IRB

Multicast IRB provides the ability to route multicast packets between a bridge group and a routed interface using a bridge-group virtual interface (BVI). It can be enabled with multicast-routing. THE BVI is a virtual interface within the router that acts like a normal routed interface. For details about BVI, refer *Cisco ASR 9000 Series Aggregation Services Router Interface and Hardware Component Configuration Guide*

BV interfaces are added to the existing VRF routes and integrated with the replication slot mask. After this integration, the traffic coming from a VRF BVI is forwarded to the VPN.

Supported bridge port types

- Bundles
- Satellites
- EFPs (physical, vlans, etc)
- Pseudowires

Restrictions

- Supported only on Ethernet line cards and enhanced ethernet line cards.
- Support only for IPv4
- Supports IGMP snooping

Multicast IRB

The CE-PE is collapsed into 1 router (IRB) and IGMP snooping is enabled on the BVIs.

BVI type is included in a multicast VRF. After the BVI slot mask is included in the VRF route slot mask, the traffic from the VRF BVI is forwarded to the VPN/ core.

Multicast support for PW-HE interfaces

Multicast support for Pseudowire Head-end (PW-HE) interfaces is available only on the enhanced ethernet cards.

Multicast support is available under these circumstances:

- IPv4 and IPv6 multicast traffic forwarding over the L3 PW-HE interface/sub-interface. PW-HE interface type can be PW-ether (VC4 or VC5) or PW-iw (VC11). IPv6 multicast is not available on VC11.
- L3 PW-HE interfaces/sub-interfaces in global , MVPNv4 and MVPNv6 VRFs.
- L3 PW-HE interface/sub-interfaces in MVPNv4 and MVPNv6 where the core can be GRE or MLDP.
- PIM-SM, PIM-SSM (PE-CE) , MSDP and PIM Auto-RP over the PW-HE interface.
- IGMP/ MLD snooping on L2 PW-HE VC5 sub-interface.
- VC label-based load balancing.

Multicast Source Discovery Protocol

Multicast Source Discovery Protocol (MSDP) is a mechanism to connect multiple PIM sparse-mode domains. MSDP allows multicast sources for a group to be known to all rendezvous points (RPs) in different domains. Each PIM-SM domain uses its own RPs and need not depend on RPs in other domains.

An RP in a PIM-SM domain has MSDP peering relationships with MSDP-enabled routers in other domains. Each peering relationship occurs over a TCP connection, which is maintained by the underlying routing system.

MSDP speakers exchange messages called Source Active (SA) messages. When an RP learns about a local active source, typically through a PIM register message, the MSDP process encapsulates the register in an SA message and forwards the information to its peers. The message contains the source and group information for the multicast flow, as well as any encapsulated data. If a neighboring RP has local joiners for the multicast group, the RP installs the S, G route, forwards the encapsulated data contained in the SA message, and sends PIM joins back towards the source. This process describes how a multicast path can be built between domains.

Note Although you should configure BGP or Multiprotocol BGP for optimal MSDP interdomain operation, this is not considered necessary in the Cisco IOS XR Software implementation. For information about how BGP or Multiprotocol BGP may be used with MSDP, see the MSDP RPF rules listed in the Multicast Source Discovery Protocol (MSDP), Internet Engineering Task Force (IETF) Internet draft.

VRF-aware MSDP

VRF (VPN Routing and Forwarding) -aware MSDP enables MSDP to function in the VRF context. This in turn, helps the user to locate the PIM (protocol Independent Multicast) RP on the Provider Edge and use MSDP for anycast-RP.

MSDP needs to be VRF-aware when:

- Anycast-RP is deployed in an MVPN (Multicast MVPN) in such a manner that one or more PIM RPs in the anycast-RP set are located on a PE. In such a deployment, MSDP needs to operate in the VRF context on the PE.
- The PIM RP is deployed in an MVPN in such a manner that it is not on a PE and when the customer multicast routing type for the MVPN is BGP and the PEs have suppress-shared-tree-join option configured. In this scenario, there is no PE-shared tree link, so traffic may stop at the RP and it does not flow to other MVPN sites. An MSDP peering between the PIM RP and one or more PEs resolves the issue.

Multicast Nonstop Forwarding

The Cisco IOS XR Software nonstop forwarding (NSF) feature for multicast enhances high availability (HA) of multicast packet forwarding. NSF prevents hardware or software failures on the control plane from disrupting the forwarding of existing packet flows through the router.

The contents of the Multicast Forwarding Information Base (MFIB) are frozen during a control plane failure. Subsequently, PIM attempts to recover normal protocol processing and state before the neighboring routers time out the PIM hello neighbor adjacency for the problematic router. This behavior prevents the NSF-capable router from being transferred to neighbors that will otherwise detect the failure through the timed-out adjacency. Routes in MFIB are marked as stale after entering NSF, and traffic continues to be forwarded (based on those routes) until NSF completion. On completion, MRIB notifies MFIB and MFIB performs a mark-and-sweep to synchronize MFIB with the current MRIB route information.

Multicast Configuration Submodes

Cisco IOS XR Software moves control plane CLI configurations to protocol-specific submodes to provide mechanisms for enabling, disabling, and configuring multicast features on a large number of interfaces.

Cisco IOS XR Software allows you to issue most commands available under submodes as one single command string from the global or XR config mode.

For example, the **ssm** command could be executed from the multicast-routing configuration submode like this:

```
RP/0/RSP0/CPU0:router(config)# multicast-routing
RP/0/RSP0/CPU0:router(config-mcast-ipv4)# ssm range
```

Alternatively, you could issue the same command from the global or XR config mode like this:

```
RP/0/RSP0/CPU0:router(config)# multicast-routing ssm range
```

The following multicast protocol-specific submodes are available through these configuration submodes:

Multicast-Routing Configuration Submode

In Cisco IOS XR software release 3.7.2 and later, basic multicast services start automatically when the multicast PIE (asr9k-mcast-p.pie) is installed, without any explicit configuration required. The following multicast services are started automatically:

- MFWD
- MRIB
- PIM
- IGMP

Other multicast services require explicit configuration before they start. For example, to start the MSDP process, you must enter the **router msdp** command and explicitly configure it.

When you issue the **multicast-routing ipv4 or multicast-routing ipv6** command, all default multicast components (PIM, IGMP, MLD, MFWD, and MRIB) are automatically started, and the CLI prompt changes to "config-mcast-ipv4" or "config-mcast-ipv6", indicating that you have entered multicast-routing configuration submode.

PIM Configuration Submode

When you issue the **router pim** command, the CLI prompt changes to "config-pim-ipv4," indicating that you have entered the default pim address-family configuration submode.

To enter pim address-family configuration submode for IPv6, type the **address-family ipv6** keyword together with the **router pim** command before pressing Enter.

IGMP Configuration Submode

When you issue the **router igmp** command, the CLI prompt changes to "config-igmp," indicating that you have entered IGMP configuration submode.

MLD Configuration Submode

When you issue the **router mld** command, the CLI prompt changes to "config-mld," indicating that you have entered MLD configuration submode.

MSDP Configuration Submode

When you issue the **router msdp** command, the CLI prompt changes to "config-msdp," indicating that you have entered router MSDP configuration submode.

Understanding Interface Configuration Inheritance

Cisco IOS XR Software allows you to configure commands for a large number of interfaces by applying command configuration within a multicast routing submode that could be inherited by all interfaces. To

override the inheritance mechanism, you can enter interface configuration submode and explicitly enter a different command parameter.

For example, in the following configuration you could quickly specify (under router PIM configuration mode) that all existing and new PIM interfaces on your router will use the hello interval parameter of 420 seconds. However, Packet-over-SONET/SDH (POS) interface 0/1/0/1 overrides the global interface configuration and uses the hello interval time of 210 seconds.

```
RP/0/RSP0/CPU0:router(config)# router pim
RP/0/RSP0/CPU0:router(config-pim-default-ipv4)# hello-interval 420
RP/0/RSP0/CPU0:router(config-pim-default-ipv4)# interface pos 0/1/0/1
RP/0/RSP0/CPU0:router(config-pim-ipv4-if)# hello-interval 210
```

The following is a listing of commands (specified under the appropriate router submode) that use the inheritance mechanism:

```
router pim
  dr-priority
  hello-interval
  join-prune-interval

multicast-routing
  version
  query-interval
  query-max-response-time
  explicit-tracking
router mld
  interface all disable
  version
  query-interval
  query-max-response-time
  explicit-tracking

router msdp
  connect-source
  sa-filter
  filter-sa-request list
  remote-as
  ttl-threshold
```

Understanding Interface Configuration Inheritance Disablement

As stated elsewhere, Cisco IOS XR Software allows you to configure multiple interfaces by applying configurations within a multicast routing submode that can be inherited by all interfaces.

To override the inheritance feature on specific interfaces or on all interfaces, you can enter the address-family IPv4 or IPv6 submode of multicast routing configuration mode, and enter the **interface-inheritance disable** command together with the **interface** *type interface-path-id* or **interface all** command. This causes PIM or IGMP protocols to disallow multicast routing and to allow only multicast forwarding on those interfaces specified. However, routing can still be explicitly enabled on specified individual interfaces.

The following configuration disables multicast routing interface inheritance under PIM and IGMP generally, although forwarding enablement continues. The example shows interface enablement under IGMP of GigabitEthernet 0/6/0/3:

```
RP/0/RSP0/CPU0:router# multicast-routing  address-family ipv4
RP/0/RSP0/CPU0:router(config-mcast-default-ipv4)# interface all enable
RP/0/RSP0/CPU0:router(config-mcast-default-ipv4)# interface-inheritance disable
```

```
!
!
RP/0/RSP0/CPU0:router(config)# router igmp
RP/0/RSP0/CPU0:router(config-igmp)# vrf default
RP/0/RSP0/CPU0:router(config-igmp)# interface GigabitEthernet0/6/0/0
RP/0/RSP0/CPU0:router(config-igmp-name-if)# router enable
```

For related information, see Understanding Enabling and Disabling Interfaces, on page 118

Understanding Enabling and Disabling Interfaces

When the Cisco IOS XR Software multicast routing feature is configured on your router, by default, no interfaces are enabled.

To enable multicast routing and protocols on a single interface or multiple interfaces, you must explicitly enable interfaces using the **interface** command in multicast routing configuration mode.

To set up multicast routing on all interfaces, enter the **interface all** command in multicast routing configuration mode. For any interface to be fully enabled for multicast routing, it must be enabled specifically (or be default) in multicast routing configuration mode, and it must not be disabled in the PIM and IGMP/MLD configuration modes.

For example, in the following configuration, all interfaces are explicitly configured from multicast routing configuration submode:

```
RP/0/RSP0/CPU0:router(config)# multicast-routing
RP/0/RSP0/CPU0:router(config-mcast)# interface all enable
```

To disable an interface that was globally configured from the multicast routing configuration submode, enter interface configuration submode, as illustrated in the following example:

```
RP/0/RSP0/CPU0:router(config-mcast)# interface GigabitEthernet0pos 0/1/0/0
RP/0/RSP0/CPU0:router(config-mcast-default-ipv4-if)# disable
```

Multicast Routing Information Base

The Multicast Routing Information Base (MRIB) is a protocol-independent multicast routing table that describes a logical network in which one or more multicast routing protocols are running. The tables contain generic multicast routes installed by individual multicast routing protocols. There is an MRIB for every logical network (VPN) in which the router is configured. MRIBs do not redistribute routes among multicast routing protocols; they select the preferred multicast route from comparable ones, and they notify their clients of changes in selected attributes of any multicast route.

Multicast Forwarding Information Base

Multicast Forwarding Information Base (MFIB) is a protocol-independent multicast forwarding system that contains unique multicast forwarding entries for each source or group pair known in a given network. There is a separate MFIB for every logical network (VPN) in which the router is configured. Each MFIB entry resolves a given source or group pair to an incoming interface (IIF) for reverse forwarding (RPF) checking and an outgoing interface list (olist) for multicast forwarding.

MSDP MD5 Password Authentication

MSDP MD5 password authentication is an enhancement to support Message Digest 5 (MD5) signature protection on a TCP connection between two Multicast Source Discovery Protocol (MSDP) peers. This feature provides added security by protecting MSDP against the threat of spoofed TCP segments being introduced into the TCP connection stream.

MSDP MD5 password authentication verifies each segment sent on the TCP connection between MSDP peers. The **password clear** command is used to enable MD5 authentication for TCP connections between two MSDP peers. When MD5 authentication is enabled between two MSDP peers, each segment sent on the TCP connection between the peers is verified.

Note MSDP MD5 authentication must be configured with the same password on both MSDP peers to enable the connection between them. The 'password encrypted' command is used only for applying the stored running configuration. Once you configure the MSDP MD5 authentication, you can restore the configuration using this command.

MSDP MD5 password authentication uses an industry-standard MD5 algorithm for improved reliability and security.

Overriding VRFs in IGMP Interfaces

All unicast traffic on the user-to-network interfaces of next-generation aggregation or core networks must be mapped to a specific VRF. They must then be mapped to an MPLS VPN on the network-to-network side. This requires the configuration of a physical interface in this specific VRF.

This feature allows mapping of IGMP packets entering through a user-to-user interface to the multicast routes in the global multicast routing table. This ensures that the interface in a specific VRF can be part of the outgoing list of interfaces in the table for a multicast route.

IGMP packets entering through a non-default VRF interface in the default (global) VRF are processed, with IGMP later distributing the interface-related multicast state (route/interface) to MRIB. This occurs through the default VRF rather than through the VRF to which the interface belongs. MRIB, PIM, MSDP, and MFIB then process the multicast state for this interface through the default VRF.

When an IGMP join for a specific (S, G) is received on the configured interface, IGMP stores this information in its VRF-specific databases. But, when sending an update to MRIB, IGMP sends this route through the default VRF. MRIB then programs this (S, G) along with this interface as an OLIST member in the default multicast routing table.

Similarly, when PIM requests information about IGMP routes from MRIB, MRIB sends this update to PIM in the context of the default VRF.

This feature specifically supports:

- Mapping of IGMP requests on an interface in a non-default VRF to the default VRF multicast routing table.
- Enabling and disabling of VRF override functionality at run time.
- Routing policy configuration at the global (default) VRF level, because routing policy configuration cannot be done at the granularity of an individual interface.

- Enablement and disablement of an IGMP VRF override on all Layer- 3 and Layer- 2 interface types, including physical Ethernet, VLAN sub-interface, bundles and VLANs over bundles.
- The same scale of multicast routes and OLIST interfaces currently supported by the platform even when VRF override functionality is operational.

VRF support for MLD

MLD receives MLD joins, membership queries and membership reports under VRF. The MLD process will have LPTS entries per VRF and traffic is redirected based on the matching VRF entry to the correct interface configured under the given VRF. Support for Source-Specific Multicast(SSM) is also provided under VRF .

How to Implement Multicast Routing

This section contains instructions for both building a basic multicast configuration, as well as optional tasks to help you to optimize, debug, and discover the routers in your multicast network.

Configuring PIM-SM and PIM-SSM

SUMMARY STEPS

1. **configure**
2. **multicast-routing** [**address-family** {**ipv4** | **ipv6**}]
3. **interface all enable**
4. exit
5. Use **router igmp** for IPv4 hosts or use **router mld** for IPv6
6. **version** {**1** | **2** | **3**} for IPv4 (IGMP) hosts or **version** {**1** | **2**} for IPv6 (MLD) hosts.
7. **commit**
8. **show pim** [**ipv4** | **ipv6**] **group-map** [*ip-address-name*] [**info-source**]
9. **show pim** [**vrf** *vrf-name*] [**ipv4** | **ipv6**] **topology** [*source-ip-address* [*group-ip-address*] | **entry-flag** *flag* | **interface-flag** | **summary**] [**route-count**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**multicast-routing** [**address-family** {**ipv4**	**ipv6**}] **Example:** `RP/0/RSP0/CPU0:router(config)# multicast-routing`	Enters multicast routing configuration mode. • The following multicast processes are started: MRIB, MFWD, PIM, and IGMP. • For IPv4, IGMP version 3 is enabled by default.

	Command or Action	Purpose
Step 3	**interface all enable** **Example:** `RP/0/RSP0/CPU0:router(config-mcast-ipv4)# interface all enable`	Enables multicast routing and forwarding on all new and existing interfaces.
Step 4	exit **Example:** `RP/0/RSP0/CPU0:router(config-mcast-ipv4)# exit`	Exits multicast routing configuration mode, and returns the router to the source configuration mode.

Step 5	Use **router igmp** for IPv4 hosts or use **router mld** for IPv6
	
	Example:
	
	`RP/0/RSP0/CPU0:router(config)# router igmp`
	`RP/0/RSP0/CPU0:router(config)# router mld`
		(Optional) Enters router IGMP configuration mode (for IPv4 hosts), or enters router MLD configuration mode (for IPv6 hosts).

Step 6	**version** {**1** | **2** | **3**} for IPv4 (IGMP) hosts or **version** {**1** | **2**} for IPv6 (MLD) hosts.
	
	Example:
	
	`RP/0/RSP0/CPU0:router(config-igmp)# version 3`
	`RP/0/RSP0/CPU0:router(config-mld)# version 2`
		(Optional) Selects the IGMP or MLD version that the router interface uses.
		• The version range for IGMP is 1-3; the range for MLD is 1-2.
		• The default for IGMP is version 3; the default for MLD is version 1.
		• Host receivers must support IGMPv3 for PIM-SSM operation.
		• If this command is configured in router IGMP or router MLD configuration mode, parameters are inherited by all new and existing interfaces. You can override these parameters on individual interfaces from interface configuration mode.

Step 7	**commit**	
Step 8	**show pim** [**ipv4** | **ipv6**] **group-map** [*ip-address-name*] [**info-source**] **Example:** `RP/0/RSP0/CPU0:router# show pim ipv4 group-map`	(Optional) Displays group-to-PIM mode mapping.
Step 9	**show pim** [**vrf** *vrf-name*] [**ipv4** | **ipv6**] **topology** [*source-ip-address* [*group-ip-address*] | **entry-flag** *flag* | **interface-flag** | **summary**] [**route-count**]	(Optional) Displays PIM topology table information for a specific group or all groups.

	Command or Action	Purpose
	Example: `RP/0/RSP0/CPU0:router# show pim topology`	

Configuring PIM-SSM for Use in a Legacy Multicast Deployment

Deploying PIM-SSM in legacy multicast-enabled networks can be problematic, because it requires changes to the multicast group management protocols used on the various devices attached to the network. Host, routers, and switches must all be upgraded in such cases.

To support legacy hosts and switches in a PIM-SSM deployment,Cisco ASR 9000 Series Routers offer a configurable mapping feature. Legacy group membership reports for groups in the SSM group range are mapped to a set of sources providing service for that set of (S,G) channels.

This configuration consists of two tasks:

Restrictions for PIM-SSM Mapping

PIM-SSM mapping does not modify the SSM group range. Instead, the legacy devices must report group membership for desired groups in the SSM group range .

Configuring a Set of Access Control Lists for Static SSM Mapping

This tasks configures a set of access control lists (ACLs) where each ACL describes a set of SSM groups to be mapped to one or more sources.

SUMMARY STEPS

1. **configure**
2. **ipv4 access-list** *acl-name*
3. [*sequence-number*] **permit** *source* [*source-wildcard*]
4. Repeat Step 3, on page 123 to add more entries to the ACL.
5. Repeat Step 2, on page 123 through Step 4, on page 123 until you have entered all the ACLs you want to be part of the set.
6. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

	Command or Action	Purpose
Step 2	**ipv4 access-list** *acl-name* **Example:** `RP/0/RSP0/CPU0:router(config)# ipv4 access-list mc3`	Enters IPv4 ACL configuration submode and creates a name for an IPv4 access list.
Step 3	[*sequence-number*] **permit** *source* [*source-wildcard*] **Example:** `RP/0/RSP0/CPU0:router(config-ipv4-acl)# permit 1 host 232.1.1.2 any`	Sets conditions for the access list to recognize the source as part of the specified access list set, in which each ACL describes a set of SSM groups to be mapped.
Step 4	Repeat Step 3, on page 123 to add more entries to the ACL.	—
Step 5	Repeat Step 2, on page 123 through Step 4, on page 123 until you have entered all the ACLs you want to be part of the set.	—
Step 6	**commit**	

Configuring a Set of Sources for SSM Mapping

This task consists of configuring a set of sources mapped by SSM groups, as described by access lists (ACLs).

SUMMARY STEPS

1. **configure**
2. **router igmp** [**vrf** *vrf-name*]
3. **ssm map static** *source-address access-list*
4. Repeat Step 3, on page 124 as many times as you have source addresses to include in the set for SSM mapping.
5. **commit**
6. **show igmp** [**vrf** *vrf-name*] **ssm map** [*group-address*][*detail*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router igmp** [**vrf** *vrf-name*] **Example:** `RP/0/RSP0/CPU0:router(config)# router igmp vrf vrf20`	Enters router IGMP configuration mode.

	Command or Action	Purpose
Step 3	**ssm map static** *source-address access-list* **Example:** `RP/0/RSP0/CPU0:router(config-igmp)# ssm map static 232.1.1.1 mc2`	Configures a source as part of a set of sources that map SSM groups described by the specified access list.
Step 4	Repeat Step 3, on page 124 as many times as you have source addresses to include in the set for SSM mapping.	—
Step 5	**commit**	

Step 6	**show igmp** [**vrf** *vrf-name*] **ssm map** [*group-address*][*detail*]
	
	Example:
	`RP/0/RSP0/CPU0:router# show igmp vrf vrf20 ssm map 232.1.1.1`
	
	`232.1.1.1 is static with 1 source`
	or
	
	`RP/0/RSP0/CPU0:router# show igmp vrf vrf20 ssm map 232.1.1.0 is static with 3 sources 232.1.1.1 is static with 1 source`
		(Optional) Queries the mapping state.
		• When you provide one address for mapping, you receive the state for that address alone.
		• When you provide no address for mapping, you receive the state for all sources.

Configuring a Static RP and Allowing Backward Compatibility

When PIM is configured in sparse mode, you must choose one or more routers to operate as a rendezvous point (RP) for a multicast group. An RP is a single common root placed at a chosen point of a shared distribution tree. An RP can either be configured statically in each router, or learned through Auto-RP or BSR.

This task configures a static RP. For more information about RPs, see the Rendezvous Points, on page 84. For configuration information for Auto-RP, see the Configuring Auto-RP to Automate Group-to-RP Mappings, on page 126.

SUMMARY STEPS

1. **configure**
2. **router pim** [**address-family** {**ipv4** | **ipv6**}]
3. **rp-address** *ip-address* [*group-access-list*] [**bidir**] [**override**]
4. **old-register-checksum**
5. **exit**
6. {**ipv4** | **ipv6**} **access-list** *name*
7. [*sequence-number*] **permit** *source* [*source-wildcard*]
8. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router pim** [**address-family** {**ipv4** | **ipv6**}] **Example:** `RP/0/RSP0/CPU0:router(config)# router pim`	Enters PIM configuration mode, or PIM address-family configuration submode.
Step 3	**rp-address** *ip-address* [*group-access-list*] [**bidir**] [**override**] **Example:** `RP/0/RSP0/CPU0:router(config-pim-default-ipv4)# rp-address 172.16.6.22 rp-access`	Assigns an RP to multicast groups. • If you specify a **group-access-list-number** value, you must configure that access list using the **ipv4 access-list** command.
Step 4	**old-register-checksum** **Example:** `RP/0/RSP0/CPU0:router(config-pim-ipv4)# old-register-checksum`	(Optional) Allows backward compatibility on the RP that uses old register checksum methodology.
Step 5	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-pim-ipv4)# exit`	Exits PIM configuration mode, and returns the router to the source configuration mode.
Step 6	{**ipv4** | **ipv6**} **access-list** *name* **Example:** `RP/0/RSP0/CPU0:router(config)# ipv4 access-list rp-access`	(Optional) Enters access list configuration mode and configures the RP access list. • The access list called “rp-access” permits multicast group 239.1.1.0 0.0.255.255.
Step 7	[*sequence-number*] **permit** *source* [*source-wildcard*] **Example:** `RP/0/RSP0/CPU0:router(config-ipv4-acl)# permit 239.1.1.0 0.0.255.255`	(Optional) Permits multicast group 239.1.1.0 0.0.255.255 for the “rp-access” list. **Tip** The commands in Step 6, on page 125 and Step 7, on page 125 can be combined in one command string like this: ipv4 access-list rp-access permit 239.1.1.0 0.0.255.255.
Step 8	**commit**	

Configuring Auto-RP to Automate Group-to-RP Mappings

This task configures the Auto-RP mechanism to automate the distribution of group-to-RP mappings in your network. In a network running Auto-RP, at least one router must operate as an RP candidate and another router must operate as an RP mapping agent. The VRF interface on Cisco ASR 9000 Series Routers cannot be an auto-rp candidate- rp.

SUMMARY STEPS

1. **configure**
2. **router pim** [**address-family ipv4**]
3. **auto-rp candidate-rp** *type instance* **scope** *ttl-value* [**group-list** *access-list-name*] [**interval** *seconds*] **bidir**
4. **auto-rp mapping-agent** *type number* **scope** *ttl-value* [**interval** *seconds*]
5. **exit**
6. **ipv4 access-list** *name*
7. [*sequence-number*] **permit** *source* [*source-wildcard*]
8. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router pim** [**address-family ipv4**] **Example:** `RP/0/RSP0/CPU0:router(config)# router pim`	Enters PIM configuration mode, or PIM address-family configuration submode.
Step 3	**auto-rp candidate-rp** *type instance* **scope** *ttl-value* [**group-list** *access-list-name*] [**interval** *seconds*] **bidir** **Example:** `RP/0/RSP0/CPU0:router(config-pim-ipv4)# auto-rp candidate-rp GigabitEthernet0/1/0/1 scope 31 group-list 2 bidir`	Configures an RP candidate that sends messages to the CISCO-RP-ANNOUNCE multicast group (224.0.1.39). • This example sends RP announcements out all PIM-enabled interfaces for a maximum of 31 hops. The IP address by which the router wants to be identified as an RP is the IP address associated with GigabitEthernet interface 0/1/0/1. • Access list 2 designates the groups this router serves as RP. • If you specify **group-list**, you must configure the optional **access-list** command.
Step 4	**auto-rp mapping-agent** *type number* **scope** *ttl-value* [**interval** *seconds*] **Example:** `RP/0/RSP0/CPU0:router(config-pim-ipv4)#`	Configures the router to be an RP mapping agent on a specified interface. • After the router is configured as an RP mapping agent and determines the RP-to-group mappings through the CISCO-RP-ANNOUNCE (224.0.1.39) group, the router sends

	Command or Action	Purpose
	`auto-rp mapping-agent GigabitEthernet0/1/0/1 scope 20`	the mappings in an Auto-RP discovery message to the well-known group CISCO-RP-DISCOVERY (224.0.1.40). • A PIM DR listens to this well-known group to determine which RP to use. • This example limits Auto-RP discovery messages to 20 hops.
Step 5	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-pim-ipv4)# exit`	Exits PIM configuration mode and returns the router to the source configuration mode.
Step 6	**ipv4 access-list** *name* **Example:** `RP/0/RSP0/CPU0:router(config)# ipv4 access-list 2`	(Optional) Defines the RP access list.
Step 7	[*sequence-number*] **permit** *source* [*source-wildcard*] **Example:** `RP/0/RSP0/CPU0:router(config-ipv4-acl)# permit 239.1.1.1 0.0.0.0`	(Optional) Permits multicast group 239.1.1.1 for the RP access list. **Tip** The commands in Step 6, on page 127 and Step 7, on page 127 can be combined in one command string and entered from the global or XR config mode like this: ipv4 access-list rp-access permit 239.1.1.1 0.0.0.0
Step 8	**commit**	

Configuring the Bootstrap Router

This task configures one or more candidate bootstrap routers (BSRs) and a BSR mapping agent. This task also connects and locates the candidate BSRs in the backbone portion of the network.

For more information about BSR, see the PIM Bootstrap Router, on page 86.

SUMMARY STEPS

1. **configure**
2. **router pim** [**address-family** {**ipv4** | **ipv6**}]
3. **bsr candidate-bsr** *ip-address* [**hash-mask-len** *length*] [**priority** *value*]
4. **bsr candidate-rp** *ip-address* [**group-list** *access-list* **interval** *seconds*] [**priority** *value*] **bidir**
5. **interface** *type interface-path-id*
6. **bsr-border**
7. **exit**
8. **exit**
9. {**ipv4** | **ipv6**} **access-list** *name*
10. Do one of the following:
    - [*sequence-number*] **permit** *source* [*source-wildcard*]
    - [*sequence-number*] **permit** *source-prefix dest-prefix*
11. **commit**
12. **clear pim** [**vrf** *vrf-name*] [**ipv4** | **ipv6**] **bsr**
13. **show pim** [**vrf** *vrf-name*] [**ipv4** | **ipv6**] **bsr candidate-rp**
14. **show pim** [**vrf** *vrf-name*] [**ipv4** | **ipv6**] **bsr election**
15. **show pim** [**vrf** *vrf-name*][**ipv4** | **ipv6**] **bsr rp-cache**
16. **show pim** [**vrf** *vrf-name*][**ipv4** | **ipv6**] **group-map** [*ip-address-name*] [**info-source**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router pim** [**address-family** {**ipv4**	**ipv6**}] **Example:** `RP/0/RSP0/CPU0:router(config)# router pim`	Enters PIM configuration mode, or address-family configuration submode.
Step 3	**bsr candidate-bsr** *ip-address* [**hash-mask-len** *length*] [**priority** *value*] **Example:** `RP/0/RSP0/CPU0:router(config-pim-default-ipv4)# bsr candidate-bsr 10.0.0.1 hash-mask-len 30`	Configures the router to announce its candidacy as a BSR.
Step 4	**bsr candidate-rp** *ip-address* [**group-list** *access-list* **interval** *seconds*] [**priority** *value*] **bidir**	Configures the router to advertise itself as a PIM Version 2 candidate RP to the BSR.

	Command or Action	Purpose
	Example: `RP/0/RSP0/CPU0:router(config-pim-default-ipv4)# bsr candidate-rp 172.16.0.0 group-list 4 bidir`	• See Step 9, on page 129 for group list 4 configuration.
Step 5	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config-pim-default-ipv4)# interface GigE 0/1/0/0`	(Optional) Enters interface configuration mode for the PIM protocol.
Step 6	**bsr-border** **Example:** `RP/0/RSP0/CPU0:router(config-pim-ipv4-if)# bsr-border`	(Optional) Stops the forwarding of bootstrap router (BSR) messages on a Protocol Independent Multicast (PIM) router interface.
Step 7	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-pim-ipv4-if)# exit`	(Optional) Exits PIM interface configuration mode, and returns the router to PIM configuration mode.
Step 8	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-pim-default-ipv4)# exit`	Exits PIM configuration mode.
Step 9	{**ipv4** | **ipv6**} **access-list** *name* **Example:** `RP/0/RSP0/CPU0:router(config)# ipv4 access-list 4`	(Optional) Defines the candidate group list to the BSR. • Access list number 4 specifies the group prefix associated with the candidate RP address 172.16.0.0. (See Step 4, on page 128). • This RP is responsible for the groups with the prefix 239.
Step 10	Do one of the following: • [*sequence-number*] **permit** *source* [*source-wildcard*] • [*sequence-number*] **permit** *source-prefix dest-prefix* **Example:** `RP/0/RSP0/CPU0:router(config-ipv4-acl)# permit`	(Optional) Permits multicast group 239.1.1.1 for the candidate group list. **Tip** The commands in Step 6, on page 129 and Step 7, on page 129 can be combined in one command string and entered from global configuration mode like this: ipv4 access-list rp-access permit 239.1.1.1 0.255.255.255

	Command or Action	Purpose
	`239.1.1.1 0.255.255.255`	
Step 11	**commit**	
Step 12	**clear pim** [**vrf** *vrf-name*] [**ipv4** | **ipv6**] **bsr** **Example:** `RP/0/RSP0/CPU0:router# clear pim bsr`	(Optional) Clears BSR entries from the PIM RP group mapping cache.
Step 13	**show pim** [**vrf** *vrf-name*] [**ipv4** | **ipv6**] **bsr candidate-rp** **Example:** `RP/0/RSP0/CPU0:router# show pim bsr candidate-rp`	(Optional) Displays PIM candidate RP information for the BSR.
Step 14	**show pim** [**vrf** *vrf-name*] [**ipv4** | **ipv6**] **bsr election** **Example:** `RP/0/RSP0/CPU0:router# show pim bsr election`	(Optional) Displays PIM candidate election information for the BSR.
Step 15	**show pim** [**vrf** *vrf-name*][**ipv4** | **ipv6**] **bsr rp-cache** **Example:** `RP/0/RSP0/CPU0:router# show pim bsr rp-cache`	(Optional) Displays PIM RP cache information for the BSR.
Step 16	**show pim** [**vrf** *vrf-name*][**ipv4** | **ipv6**] **group-map** [*ip-address-name*] [**info-source**] **Example:** `RP/0/RSP0/CPU0:router# show pim ipv4 group-map`	(Optional) Displays group-to-PIM mode mapping.

Calculating Rates per Route

This procedure enables multicast hardware forward-rate counters on a per-VRF-family basis.

SUMMARY STEPS

1. **configure**
2. **multicast-routing** [**vrf** *vrf-name*] [**address-family** {**ipv4** | **ipv6**}]
3. **rate-per-route**
4. **interface** {*type interface-path-id* | **all**} **enable**
5. Do one of the following:
   - **accounting per-prefix**
   - **accounting per-prefix forward-only**
6. **commit**
7. **show mfib** [**vrf** *vrf-name*] [**ipv4** | **ipv6**] **route** [**rate** | **statistics**] [***** | *source-address*] [*group-address* [/*prefix-length*] [**detail** | **old-output**] | **summary**] [**location** *node-id*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**multicast-routing** [**vrf** *vrf-name*] [**address-family** {**ipv4** | **ipv6**}] **Example:** `RP/0/RSP0/CPU0:router(config)# multicast-routing address-family ipv4`	Enters multicast routing configuration mode. • The following multicast processes are started: MRIB, MFWD, PIM, and IGMP. • For IPv4, IGMP version 3 is enabled by default.
Step 3	**rate-per-route** **Example:** `RP/0/RSP0/CPU0:router(config-mcast-default-ipv4)# rate-per-route`	Enables a per (S,G) rate calculation for a particular route.

Step 4	**interface** {*type interface-path-id* | **all**} **enable**
	
	Example:
	`RP/0/RSP0/CPU0:router(config-mcast-default-ipv4)# interface all enable`
	
	or
	
	`RP/0/RSP0/CPU0:router(config-mcast-default-ipv4)# interface FastEthernet0/3/3/1 enable`
		Enables multicast routing on all interfaces.

Step 5	Do one of the following: • **accounting per-prefix**	• Enables per-prefix counters present in hardware, assigning every existing and new (S, G) route forward, punt, and drop counters on the ingress route and forward

	Command or Action	Purpose
	• **accounting per-prefix forward-only** **Example:** `RP/0/RSP0/CPU0:router(config-mcast-default-ipv)# accounting per-prefix`	and punt counters on the egress route. The (*, G) routes are assigned a single counter. • Enables per-prefix counters present in hardware accounting per-prefix—Enables three counters on ingress (forward, punt and drop, and two on egress (forward and punt) on every existing and new (S, G) route. The (*, G) routes are assigned a single counter. • accounting per-prefix forward-only—Enables one counter on ingress and one on egress in hardware to conserve hardware statistics resources. (Recommended for configuration of multicast VPN routing or for any line card that has a route-intensive configuration.)
Step 6	**commit**	
Step 7	**show mfib** [**vrf** *vrf-name*] [**ipv4** | **ipv6**] **route** [**rate** | **statistics**] [***** | *source-address*] [*group-address* [*/prefix-length*] [**detail** | **old-output**] | **summary**] [**location** *node-id*] **Example:** `RP/0/RSP0/CPU0:router# show mfib vrf 12 route statistics location 0/1/cpU0`	Displays route entries in the Multicast Forwarding Information Base (MFIB) table. • When the **rate** keyword is used with the *source-* and *group-address*, the command displays the cumulative rates per route for all line cards in the Multicast Forwarding Information Base (MFIB) table. • When the **statistics** keyword is used, the command displays the rate per route for one line card in the Multicast Forwarding Information Base (MFIB) table.

Configuring Multicast Nonstop Forwarding

This task configures the nonstop forwarding (NSF) feature for multicast packet forwarding for the purpose of alleviating network failures, or software upgrades and downgrades.

Although we strongly recommend that you use the NSF lifetime default values, the optional Step 3, on page 133 through Step 6, on page 134 allow you to modify the NSF timeout values for Protocol Independent Multicast (PIM) and Internet Group Management Protocol (IGMP) or Multicast Listener Discovery (MLD). Use these commands when PIM and IGMP (or MLD) are configured with nondefault interval or query intervals for join and prune operations.

Generally, configure the IGMP NSF and PIM NSF lifetime values to equal or exceed the query or join query interval. For example, if you set the IGMP query interval to 120 seconds, set the IGMP NSF lifetime to 120 seconds (or greater).

If the Cisco IOS XR Software control plane does not converge and reconnect after NSF is enabled on your router, multicast packet forwarding continues for up to 15 minutes, then packet forwarding stops.

Before You Begin

For NSF to operate in your multicast network, you must also enable NSF for the unicast protocols (such as IS-IS, OSPF, and BGP) that PIM relies on for Reverse Path Forwarding (RPF) information. See the appropriate configuration modules to learn how to configure NSF for unicast protocols.

SUMMARY STEPS

1. **configure**
2. **router pim** [**address-family** {**ipv4** | **ipv6**}]
3. **nsf lifetime** *seconds*
4. **exit**
5. **router** {**igmp** | **mld**}
6. **nsf lifetime** *seconds*
7. **commit**
8. **show** {**igmp nsf**
9. **show mfib** [**ipv4** | **ipv6**] **nsf** [**location** *node-id*]
10. **show mrib** [**ipv4** | **ipv6**] **nsf**
11. **show pim** [**ipv4** | **ipv6**] **nsf**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router pim** [**address-family** {**ipv4** | **ipv6**}] **Example:** `RP/0/RSP0/CPU0:router(config)# router pim address-family ipv4`	(Optional) Enters PIM address-family configuration submode.
Step 3	**nsf lifetime** *seconds* **Example:** `RP/0/RSP0/CPU0:router(config-pim-default-ipv4)# nsf lifetime 30`	(Optional) Configures the NSF timeout value for multicast forwarding route entries under the PIM process. **Note** If you configure the PIM hello interval to a nondefault value, configure the PIM NSF lifetime to a value less than the hello hold time. Typically the value of the hold-time field is 3.5 times the interval time value, or 120 seconds if the PIM hello interval time is 30 seconds.
Step 4	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-pim-default-ipv4)# exit`	(Optional) Exits PIM configuration mode and returns the router to the source configuration mode.

	Command or Action	Purpose
Step 5	**router** {**igmp** | **mld**} **Example:** `RP/0/RSP0/CPU0:router(config)# router igmp`	(Optional) Enters router IGMP or MLD configuration mode.
Step 6	**nsf lifetime** *seconds* **Example:** `RP/0/RSP0/CPU0:router(config-igmp)# nsf lifetime 30`	(Optional) Configures the NSF timeout value for multicast forwarding route entries under the IGMP process.
Step 7	**commit**	
Step 8	**show** {**igmp nsf** **Example:** `RP/0/RSP0/CPU0:router# show igmp nsf`	(Optional) Displays the state of NSF operation in IGMP.
Step 9	**show mfib** [**ipv4** | **ipv6**] **nsf** [**location** *node-id*] **Example:** `RP/0/RSP0/CPU0:router# show mfib nsf`	(Optional) Displays the state of NSF operation for the MFIB line cards.
Step 10	**show mrib** [**ipv4** | **ipv6**] **nsf** **Example:** `RP/0/RSP0/CPU0:router# show mrib nsf`	(Optional) Displays the state of NSF operation in the MRIB.
Step 11	**show pim** [**ipv4** | **ipv6**] **nsf** **Example:** `RP/0/RSP0/CPU0:router# show pim nsf`	(Optional) Displays the state of NSF operation for PIM.

Configuring Multicast VPN

- Enabling a VPN for Multicast Routing, on page 136 (required)
- "Configuring BGP to Advertise VRF Routes for Multicast VPN from PE to PE" (required)

  See the module "Implementing BGP on Cisco IOS XR Software in *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide*.

- Configuring an MDT Address Family Session in BGP as a PE-to- PE Protocol (optional for PIM-SM MDT groups; required for PIM-SSM MDT groups)

  See the "Configuring an MDT Address Family Session in BGP" section in *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide*.

- Configuring a provider-edge-to-customer-edge protocol (optional)

  See the "Configuring BGP as a PE-CE Protocol," "Configuring OSPF as a PE-to-CE Protocol," and "Configuring EIGRP as a PE-to CE Protocol" sections in *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide*.

- Specifying the PIM VRF Instance, on page 138 (optional)

Prerequisites for Multicast VPN

- PIM and multicast forwarding must be configured on all interfaces used by multicast traffic. In an MVPN, you must enable PIM and multicast forwarding for the following interfaces:
  - Physical interface on a provider edge (PE) router that is connected to the backbone.
  - Interface used for BGP peering source address.
  - Any interfaces configured as PIM rendezvous points.

Note PIM and multicast forwarding are enabled in multicast routing configuration mode. No additional configuration is required in router pim mode to enable the PIM protocol.

- Interfaces in the VPN intended for use in forwarding multicast traffic must be enabled for PIM and multicast forwarding.
- BGP should already be configured and operational on all routers that are sending or receiving multicast traffic.
- To enable MVPN, you must include a VPN IPv4 address-family (AFI) in your BGP configuration. See Restrictions for Multicast VPN for Multicast Routing, on page 135. (See also the "Enabling BGP Routing" section in Cisco IOS XR Routing Configuration Guide.)
- All PE routers in the multicast domain must be running a Cisco IOS XR Software image that supports MVPN.
- Multicast forwarding must be configured for the global IPv4 address family.
- Each multicast SM VRF domain must have an associated PIM rendezvous point (RP) definition. Using Auto-RP and the bootstrap router (BSR), you may configure RP services in the MVPN on the customer-edge (CE) device because the MVPN learns about the RP dynamically. The VRF interface can be used as a listener on the PE device.

  To enable static RP services, you must configure every device in the domain for this purpose.

Restrictions for Multicast VPN for Multicast Routing

- Configuration of the MDT source on a per-VRF basis is only supported on IPv4.

- The MDT group address should be the same for both the address families in the same VRF.

Enabling a VPN for Multicast Routing

This task enables multicast VPN routing for IPv4.

The MDT group address is used by provider edge (PE) routers to form a virtual PIM “neighborship” for the MDT. This enables the PEs to communicate with other PEs in the VRF as if they shared a LAN.

When sending customer VRF traffic, PEs encapsulate the traffic in their own (S,G) state, where the G is the MDT group address, and the S is the MDT source for the PE. By joining the (S,G) MDT of its PE neighbors, a PE router is able to receive the encapsulated multicast traffic for that VRF.

Although the VRF itself may have many multicast sources sending to many groups, the provider network needs only to install state for one group per VRF, in other words, the MDT group.

SUMMARY STEPS

1. **configure**
2. **multicast-routing**
3. **address-family ipv4**
4. **nsf**
5. **mdt source** *type interface-path-id*
6. **interface all enable**
7. **vrf** *vrf-name*
8. **address-family** {**ipv4**}]
9. **mdt default** *mdt-group-address*
10. **mdt data** *mdt-group-address/prefix-length* **threshold** *threshold acl-name*
11. **mdt mtu** *size*
12. **interface all enable**
13. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**multicast-routing** **Example:** `RP/0/RSP0/CPU0:router(config)# multicast-routing`	Enters multicast routing configuration mode.
Step 3	**address-family ipv4** **Example:** `RP/0/RSP0/CPU0:router(config-mcast)#`	Enters ipv4 address-family submode.

	Command or Action	Purpose
	`address-family ipv4`	
Step 4	**nsf** **Example:** `RP/0/RSP0/CPU0:router(config-mcast-default-ipv4)# nsf`	Specifies that nonstop forwarding (NSF) maintains the forwarding state in case of a disruption to a multicast process.
Step 5	**mdt source** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config-mcast-default-ipv4)# mdt source GigE 0/1/0/0`	Specifies the MDT source address. **Note** The MDT source interface name should be the same as the one used for BGP.
Step 6	**interface all enable** **Example:** `RP/0/RSP0/CPU0:router(config-mcast-default-ipv4)# interface all enable`	Enables multicast routing and forwarding on all new and existing interfaces. You can also enable individual interfaces. **Caution** To avoid any possibility of a reverse-path forwarding (RPF) failure, you should proactively enable any interfaces that might possibly carry multicast traffic.
Step 7	**vrf** *vrf-name* **Example:** `RP/0/RSP0/CPU0:router(config-mcast-default-)# vrf vrf_A`	Configures a VPN routing and forwarding (VRF) instance and enters VRF configuration mode.
Step 8	**address-family** {**ipv4**}]	Specifies the virtual routing and forwarding instance for the ipv4 address family.
Step 9	**mdt default** *mdt-group-address* **Example:** `RP/0/RSP0/CPU0:router(config-mcast-vrf_A-ipv4)# mdt default 239.23.2.1`	Specifies the multicast distribution tree (MDT) default group address.
Step 10	**mdt data** *mdt-group-address/prefix-length* **threshold** *threshold acl-name* **Example:** `RP/0/RSP0/CPU0:router(config-mcast-vrf_A-ipv4)# mdt data 239.23.3.0/24 threshold 1200 acl-A`	(IPv4 MVPN configuration only) Specifies the multicast group address range to be used for data MDT traffic. **Note** This group range should not overlap the MDT default group. This is an optional command. The default threshold beyond which traffic is sent using a data MDT group is 1 kbps. However, you may configure a higher threshold, if desired. You may also, optionally, configure an access list to limit the number of groups to be tunneled through a data MDT

	Command or Action	Purpose
		group. Traffic from groups not on the access-list continues to be tunneled using the default MDT group.
Step 11	**mdt mtu** *size* **Example:** `RP/0/RSP0/CPU0:router(config-mcast-default-ipv4)# mdt mtu 1550`	This is an optional step. Specifies the MTU size. It is recommended to configure a high value, to accommodate the maximum multicast packet size. **Note** The default MTU for PIM/GRE MDT is 1376 and the default value for mLDP/P2MP-TE MDT is 9000 for Multicast VPN.
Step 12	**interface all enable** **Example:** `RP/0/RSP0/CPU0:router(config-mcast-default-ipv4)# interface all enable`	Enables multicast routing and forwarding on all new and existing interfaces.
Step 13	**commit**	

Specifying the PIM VRF Instance

If you are configuring Protocol Independent Multicast in sparse mode (PIM-SM) in the MVPN, you may also need to configure a rendezvous point (RP). This task specifies the optional PIM VPN instance.

SUMMARY STEPS

1. **configure**
2. **router pim vrf** *vrf-name* **address-family** {**ipv4** | **ipv6**}
3. **rp-address** *ip-address* [*group-access-list-name*] [**bidir**] [**override**]
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router pim vrf** *vrf-name* **address-family** {**ipv4**	**ipv6**} **Example:** `RP/0/RSP0/CPU0:router(config)# router pim vrf vrf_A address-family ipv4`	Enters PIM address-family configuration submode and configures the PIM VRF for either an IPv4 or IPv6 address family.

	Command or Action	Purpose
Step 3	**rp-address** *ip-address* [*group-access-list-name*] [**bidir**] [**override**] **Example:** `RP/0/RSP0/CPU0:router(config-pim-vrf_A-ipv4)# rp-address 10.0.0.0`	Configures the PIM rendezvous point (RP) address: • **group-access-list-name** = Specifies an access list of groups to be mapped to a given RP. • **bidir** = Specifies a bidirectional RP. • **override** = Specifies that a static RP configuration should override auto-RP and the bootstrap router (BSR).
Step 4	**commit**	

Specifying the IGMP VRF Instance

SUMMARY STEPS

1. **configure**
2. **router igmp**
3. **vrf** *vrf-name*
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router igmp** **Example:** `RP/0/RSP0/CPU0:router(config)# router igmp`	Enters IGMP configuration mode.
Step 3	**vrf** *vrf-name* **Example:** `RP/0/RSP0/CPU0:router(config-igmp)# vrf vrf_B`	Configures a VRF instance.
Step 4	**commit**	

Configuring the MDT Source per VRF

This optional feature lets you change the default routing mechanism in a multicast VPN network topology, which routes all unicast traffic through a BGP peering loopback configured on a default VRF. Instead, you may configure a loopback that allows you to specify the MDT source using a specific VRF, as opposed to the default VRF. This overrides the current behavior and updates BGP as part of a MDT group. BGP then modifies the source and connector attributes in the MDT SAFI and VPN IPv4 updates.

For VRFs on which the MDT source is not configured, the MDT source for the default VRF is applied. Also, when the MDT source on a VRF is unconfigured, the configuration of the MDT source default VRF takes effect.

Note In the configuration below, the default VRF does not require explicit reference in Step 5.

SUMMARY STEPS

1. **configure**
2. **multicast-routing**
3. **address-family** [**ipv4** | **ipv6**]
4. **mdt source loopback 0**
5. **exit**
6. **vrf 101**
7. **address-family ipv4**
8. **mdt source loopback 1**
9. Repeat the steps 6 to 8, as many times as needed to create other VRFs.
10. **commit**
11. **show pim vrf all mdt interface**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

Step 2	**multicast-routing**
	
	Example:
	`RP/0/RSP0/CPU0:router(config)# multicast-routing`
	`RP/0/RSP0/CPU0:router(config-mcast)#`
		Enables IP multicast routing and forwarding.

Step 3	**address-family** [**ipv4**	**ipv6**] **Example:** `RP/0/RSP0/CPU0:router(config-mcast)# address-family ipv4`	Enters ipv4 (or ipv6) address-family submode.

	Command or Action	Purpose
Step 4	**mdt source loopback 0** **Example:** `RP/0/RSP0/CPU0:router(config-mcast-default-ipv4)# mdt source loopback 0`	Configures the interface used to set the MDT source address for MVPN, using the default VRF.
Step 5	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-mcast-default-ipv4)# exit`	Exits from the current mode.
Step 6	**vrf 101** **Example:** `RP/0/RSP0/CPU0:router(config-mcast)# vrf 101`	
Step 7	**address-family ipv4** **Example:** `RP/0/RSP0/CPU0:router(config-mcast-101)# address-family ipv4`	Enters the IPv4 address-family mode.
Step 8	**mdt source loopback 1** **Example:** `RP/0/RSP0/CPU0:router(config-mcast-101-ipv4)# mdt source loopback 1`	Configures the interface used to set the MDT source address for MVPN.
Step 9	Repeat the steps 6 to 8, as many times as needed to create other VRFs. **Example:** `RP/0/RSP0/CPU0:router(config-mcast)# vrf 102 mdt source loopback 2`	—
Step 10	**commit**	

Step 11	**show pim vrf all mdt interface**
	
	Example:
	`RP/0/RSP0/CPU0:router# show pim vrf all mdt interface`
	`GroupAddress Interface Source Vrf`
	`239.0.0.239 mdtVRF_NAME Loopback1 VRF_NAME`
		To verify the MDT source per VRF configuration, use the **show pim vrf all mdt interface** command.

	Command or Action	Purpose

Configuring Label Switched Multicast

Deployment of an LSM MLDP-based MVPN involves configuring a default MDT and one or more data MDTs. A static default MDT is established for each multicast domain. The default MDT defines the path used by PE routers to send multicast data and control messages to other PE routers in the multicast domain. A default MDT is created in the core network using a single MP2MP LSP.

An LSP MLDP-based MVPN also supports dynamic creation of the data MDTs for high-bandwidth transmission. For high-rate data sources, a data MDT is created using the P2MP LSPs to off-load the traffic from the default MDT to avoid unnecessary waste of bandwidth to PEs that are not part of the stream. You can configure MLDP MVPN for both the intranet or extranet. This configuration section covers the rosen based MLDP profile. For configuration examples of other MLDP profiles, see Configuring LSM based MLDP: Examples, on page 225.

Note Before configuring MLDP based MVPN, ensure that the MPLS is enabled on the core facing interface. For information in MPLS configuration, see Cisco IOS XR MPLS Configuration Guide. Also, ensure that BGP and any interior gateway protocol (OSPF or ISIS) is enabled on the core router. For more information on BGP and route-policy configuration, see Cisco IOS XR Routing Configuration Guide.

Perform this task to configure label switched multicast:

SUMMARY STEPS

1. **configure**
2. **mpls ldp mldp**
3. **root**
4. **vrf** *vrf_name*
5. **vpn id** *vpn-id*
6. **address-family** [**ipv4** | **ipv6**] **unicast**
7. **import route-target** [*xx.yy.nn* | *as-number:nn* | *ip-address:nn*]
8. **export route-target** [*xx.yy.nn* | *as-number:nn* | *ip-address:nn*]
9. **root**
10. **multicast-routing vrf** *vrf_name*
11. **address-family** [**ipv4** | **ipv6**]
12. **mdt default mldp ipv4** *root-node*
13. **mdt data** *mdt-group-address* **threshold** *value*
14. **root**
15. **router bgp** *as-number* **vrf** *vrf-name*
16. **rd** *route-distinguisher*
17. **address-family ipv4 mdt**
18. **address-family vpnv4 unicast**
19. **root**
20. **router pim**
21. **vrf** *vrf_name*
22. **address-family** [**ipv4** | **ipv6**]
23. **rpf topology route-policy** *route_policy_name*
24. **root**
25. **route-policy** *route_policy_name*
26. **set core-tree** *tree_type*
27. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**mpls ldp mldp** **Example:** `RP/0/RSP0/CPU0:router(config)# mpls ldp mldp`	Enables MPLS MLDP support.

	Command or Action	Purpose
Step 3	**root** **Example:** `RP/0/RSP0/CPU0:router(config-ldp-mldp)# root`	Takes the user to the global configuration level.
Step 4	**vrf** *vrf_name* **Example:** `RP/0/RSP0/CPU0:router(config)# vrf vrf1`	Configures a VRF instance. The vrf-name argument is the name assigned to a VRF.
Step 5	**vpn id** *vpn-id* **Example:** `RP/0/RSP0/CPU0:router(config-vrf)# vpn id 1:1`	Sets or updates a VPN identifier on a VRF.
Step 6	**address-family** [**ipv4** | **ipv6**] **unicast** **Example:** `RP/0/RSP0/CPU0:router(config-vrf)# address-family ipv4 unicast`	Enters the address-family submode.
Step 7	**import route-target** [*xx.yy.nn* | *as-number:nn* | *ip-address:nn*] **Example:** `RP/0/RSP0/CPU0:router(config-vrf-af)# import route-target import 1:1`	Imports the selected route target, optionally expressed as one of the following: • 4-byte AS number of the route target in xx.yy:nn format. Range is 0-65535.0-65535:0-65535 • AS number of the route target in nn format. Range is 0-65535. • IP address of the route target in A.B.C.D. format.
Step 8	**export route-target** [*xx.yy.nn* | *as-number:nn* | *ip-address:nn*] **Example:** `RP/0/RSP0/CPU0:router(config-vrf-af)# export route-target export 1:1`	Exports the selected route target, optionally expressed as one of the following: • 4-byte AS number of the route target in xx.yy:nn format. Range is 0-65535.0-65535:0-65535 • AS number of the route target in nn format. Range is 0-65535. • IP address of the route target in A.B.C.D. format.

	Command or Action	Purpose
Step 9	**root** **Example:** `RP/0/RSP0/CPU0:router(config-vrf-af)# root`	Takes the user to the global configuration level.
Step 10	**multicast-routing vrf** *vrf_name* **Example:** `RP/0/RSP0/CPU0:router(config)# multicast-routing vrf vrf1`	Enables multicast routing for the specified VRF.
Step 11	**address-family** [**ipv4** | **ipv6**] **Example:** `RP/0/RSP0/CPU0:router(config-mcast-vrf1)# address-family ipv4`	Enters the address-family submode.
Step 12	**mdt default mldp ipv4** *root-node* **Example:** `RP/0/RSP0/CPU0:router(config-mcast-vrf1-ipv4)# mdt default mldp ipv4 2.2.2.2`	Configures MLDP MDT for a VRF. The root node can be IP address of a loopback or physical interface on any router (source PE, receiver PE or core router) in the provider network. The root node address should be reachable by all the routers in the network. The router from where the signalling occurs functions as the root node. The default MDT must be configured on each PE router to enable the PE routers to receive multicast traffic for this particular MVRF. **Note** By default MPLS MLDP is enabled. To disable, use the **no mpls ldp mldp** command. **Note** LSPVIF tunnel is created as a result of **mdt default mldp** *root-node* command.
Step 13	**mdt data** *mdt-group-address* **threshold** *value* **Example:** `RP/0/RSP0/CPU0:router(config-mcast-vrf1-ipv4)# mdt data 239.0.0.0/24 threshold 1000`	Configures the threshold value for data MDT.
Step 14	**root** **Example:** `RP/0/RSP0/CPU0:router(config-mcast-vrf1-ipv4)# root`	Takes the user to the global configuration mode.

	Command or Action	Purpose
Step 15	**router bgp** *as-number* **vrf** *vrf-name* **Example:** `RP/0/RSP0/CPU0:router(config)# router bgp 1 vrf vrf1`	Enters the BGP configuration mode.
Step 16	**rd** *route-distinguisher* **Example:** `RP/0/RSP0/CPU0:router(config-bgp-vrf)# rd 1.1.1.1:1`	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 value in either of these formats: • 16-bit autonomous system number. For example, 101:3. • 32-bit IP address: your 16-bit number. For example, 192.168.122.15:1.
Step 17	**address-family ipv4 mdt** **Example:** `RP/0/RSP0/CPU0:router(config-bgp-vrf)# address-family ipv4 mdt`	Configures the BGP MDT address family.
Step 18	**address-family vpnv4 unicast** **Example:** `RP/0/RSP0/CPU0:router(config-bgp-af)# address-family vpnv4 unicast`	Configures the BGP vpnv4 address family.
Step 19	**root** **Example:** `RP/0/RSP0/CPU0:router(config-bgp-af)# root`	Takes the user to the global configuration mode.
Step 20	**router pim** **Example:** `RP/0/RSP0/CPU0:router(config)# router pim`	Enters the PIM configuration mode.
Step 21	**vrf** *vrf_name* **Example:** `RP/0/RSP0/CPU0:router(config-pim)# vrf vrf1`	Specifies the VRF instance. .

	Command or Action	Purpose
Step 22	**address-family** [**ipv4** | **ipv6**] **Example:** `RP/0/RSP0/CPU0:router(config-pim-vrf1)# address-family ipv4`	Enters the address-family submode.
Step 23	**rpf topology route-policy** *route_policy_name* **Example:** `RP/0/RSP0/CPU0:router(config-pim-vrf1-af)# rpf topology route-policy FOO`	Assigns a given routing policy to an RPF topology table.
Step 24	**root** **Example:** `RP/0/RSP0/CPU0:router(config-pim-vrf1-af)# root`	Takes the user to the global configuration mode.
Step 25	**route-policy** *route_policy_name* **Example:** `RP/0/RSP0/CPU0:router(config)# route-policy FOO`	Configures the route policy for a profile. For more information about configuring route policy, see *Cisco IOS XR Routing Configuration Guide*.
Step 26	**set core-tree** *tree_type* **Example:** `RP/0/RSP0/CPU0:router(config-rpl)# set core-tree mldp-rosen`	Specifies the MDT type for the route policy.
Step 27	**commit**	

Verification of LSM mLDP based MVPN Configuration

Use these commands to verify the LSM mLDP based MVPN intranet configuration:

- To check the MLDP neighbors, use the **show mpls mldp neighbors** command:

```
Router# show mpls mldp neighbors
mLDP neighbor database
 MLDP peer ID      : 1.0.0.1:0, uptime 15:36:30 Up,
  Capabilities     : GR, Typed Wildcard FEC, P2MP, MP2MP, MBB
  Target Adj       : No
  Upstream count   : 0
  Branch count     : 0
  LDP GR           : Enabled
```

```
                  : Instance: 1
 Label map timer  : never
 Policy filter in : None
 Path count       : 1
 Path(s)          : 11.11.11.10       GigabitEthernet0/2/0/0 LDP
 Adj list         : 11.11.11.10       GigabitEthernet0/2/0/0
 Peer addr list   : 8.39.21.2
                  : 1.0.0.1
                  : 1.1.1.1
                  : 1.2.2.1
                  : 1.3.3.1
                  : 1.4.4.1
                  : 1.5.5.1
                  : 1.6.6.1
                  : 1.7.7.1
                  : 1.8.8.1
                  : 1.9.9.1
                  : 1.10.10.1
                  : 1.11.11.1
                  : 1.12.12.1
                  : 1.13.13.1
                  : 1.14.14.1
                  : 1.15.15.1
                  : 1.16.16.1
                  : 1.17.17.1
                  : 1.18.18.1
                  : 1.19.19.1
                  : 1.20.20.1
                  : 1.21.21.1
                  : 1.22.22.1
                  : 1.23.23.1
                  : 1.24.24.1
                  : 1.25.25.1
                  : 1.26.26.1
                  : 1.27.27.1
                  : 1.28.28.1
                  : 1.29.29.1
                  : 1.30.30.1
                  : 11.11.11.10
                  : 111.113.1.5
                  : 111.112.1.1
                  : 8.39.21.222

MLDP peer ID      : 3.0.0.1:0, uptime 15:36:31 Up,
 Capabilities     : GR, Typed Wildcard FEC, P2MP, MP2MP, MBB
 Target Adj       : No
 Upstream count   : 334
 Branch count     : 328
 LDP GR           : Enabled
                  : Instance: 1
 Label map timer  : never
 Policy filter in : None
 Path count       : 1
 Path(s)          : 11.113.1.2        GigabitEthernet0/2/0/3 LDP
 Adj list         : 11.113.1.2        GigabitEthernet0/2/0/3
 Peer addr list   : 8.39.15.2
                  : 3.0.0.1
                  : 189.189.189.189
                  : 13.13.13.18
                  : 11.113.1.2
                  : 22.113.1.2
                  : 111.113.1.6
                  : 112.113.1.6
```

- To check the PIM neighbors, use the **show pim vrf vrf-name neighbor** command:

```
Router# show pim vrf A1_MIPMSI neighbor
PIM neighbors in VRF A1_MIPMSI

Neighbor Address             Interface              Uptime    Expires  DR pri  s

101.2.2.101*                 Loopback2              15:54:43  00:00:02 1 (DR) BP
```

```
101.0.0.101*                LmdtA1/MIPMSI          15:54:43  00:00:02 1      B
102.0.0.102                 LmdtA1/MIPMSI          03:52:08  00:00:02 1      B
103.0.0.103                 LmdtA1/MIPMSI          15:28:13  00:00:02 1 (DR) B
60.3.0.1                    Multilink0/2/1/0/3     15:54:39  00:01:21 1      B
60.3.0.2*                   Multilink0/2/1/0/3     15:54:43  00:00:02 1 (DR) BP
60.1.0.5                    Serial0/2/2/0/1:1.16   15:54:42  00:01:42 1      B
60.1.0.6*                   Serial0/2/2/0/1:1.16   15:54:43  00:00:02 1 (DR) BP
60.2.0.1                    Serial0/5/0/0/1        15:54:42  00:01:17 1      B
60.2.0.2*                   Serial0/5/0/0/1        15:54:43  00:00:02 1 (DR) BP
```

- To check the multicast routes for a given VRF, use **show mrib vrf vrf_name route** command:

```
Router# show mrib vrf A1_MIPMSI route
IP Multicast Routing Information Base
Entry flags: L - Domain-Local Source, E - External Source to the Domain,
    C - Directly-Connected Check, S - Signal, IA - Inherit Accept,
    IF - Inherit From, D - Drop, MA - MDT Address, ME - MDT Encap,
    MD - MDT Decap, MT - MDT Threshold Crossed, MH - MDT interface handle
    CD - Conditional Decap, MPLS - MPLS Decap, MF - MPLS Encap, EX - Extranet
    MoFE - MoFRR Enabled, MoFS - MoFRR State
Interface flags: F - Forward, A - Accept, IC - Internal Copy,
    NS - Negate Signal, DP - Don't Preserve, SP - Signal Present,
    II - Internal Interest, ID - Internal Disinterest, LI - Local Interest,
    LD - Local Disinterest, DI - Decapsulation Interface
    EI - Encapsulation Interface, MI - MDT Interface, LVIF - MPLS Encap,
    EX - Extranet, A2 - Secondary Accept

(*,224.0.0.0/24) Flags: D
  Up: 15:57:19


(*,224.0.1.39) Flags: S
  Up: 15:57:19


(*,224.0.1.40) Flags: S
  Up: 15:57:19

  Outgoing Interface List
    Serial0/5/0/0/1 Flags: II LI, Up: 15:57:12

(*,225.0.0.0/19) RPF nbr: 101.2.2.101 Flags: L C
  Up: 15:57:19

  Outgoing Interface List
    Decapstunnel98 Flags: NS DI, Up: 15:57:10

(*,225.0.32.0/19) RPF nbr: 102.0.0.102 Flags: C
  Up: 15:57:19


(*,225.0.32.1) RPF nbr: 102.0.0.102 Flags: C
  Up: 04:08:30

  Incoming Interface List
    LmdtA1/MIPMSI Flags: A LMI, Up: 04:08:30
  Outgoing Interface List
    Serial0/2/2/0/1:1.16 Flags: F NS, Up: 04:08:30

(*,225.0.32.2) RPF nbr: 102.0.0.102 Flags: C
  Up: 04:08:30

  Incoming Interface List
    LmdtA1/MIPMSI Flags: A LMI, Up: 04:08:30
  Outgoing Interface List
    Serial0/2/2/0/1:1.16 Flags: F NS, Up: 04:08:30

(*,225.0.32.3) RPF nbr: 102.0.0.102 Flags: C
  Up: 04:08:30

  Incoming Interface List
    LmdtA1/MIPMSI Flags: A LMI, Up: 04:08:30
```

```
  Outgoing Interface List
    Serial0/2/2/0/1:1.16 Flags: F NS, Up: 04:08:30

(*,225.0.32.4) RPF nbr: 102.0.0.102 Flags: C
  Up: 04:08:30

  Incoming Interface List
    LmdtA1/MIPMSI Flags: A LMI, Up: 04:08:30
  Outgoing Interface List
    Serial0/2/2/0/1:1.16 Flags: F NS, Up: 04:08:30
```

- To check the MPLS forwarding status, use **show mpls forwarding** command:

```
Router# show mpls forwarding
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
16000  16255       MLDP LSM ID: 0x1   Gi0/2/0/3    11.113.1.2      348727240
16001  16254       MLDP LSM ID: 0x3   Gi0/2/0/3    11.113.1.2      348727234
16002  16253       MLDP LSM ID: 0x5   Gi0/2/0/3    11.113.1.2      348727234
16003  16252       MLDP LSM ID: 0x7   Gi0/2/0/3    11.113.1.2      348727234
16004  16251       MLDP LSM ID: 0x9   Gi0/2/0/3    11.113.1.2      421876882
16005  16250       MLDP LSM ID: 0xb   Gi0/2/0/3    11.113.1.2      348726916
```

Configuring Multitopology Routing

This set of procedures configures multitopology routing, which is used by PIM for reverse-path forwarding (RPF) path selection.

- “Configuring a Global Topology and Associating It with an Interface” (required)

  For information, see *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide*.

- “Enabling an IS-IS Topology” (required)

  For information, see *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide*.

- “Placing an Interface in a Topology in IS-IS” (required)

  For information, see *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide*.

- “Configuring a Routing Policy” (required)

  For information, see *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide*.

Restrictions for Configuring Multitopology Routing

- Only the default VRF is currently supported in a multitopology solution.
- Only protocol-independent multicast (PIM) and intermediate system-intermediate system (IS-IS) routing protocols are currently supported.
- Topology selection is restricted solely to (S, G) route sources for both SM and SSM. Static and IS-IS are the only interior gateway protocols (IGPs) that support multitopology deployment.

  For non-(S, G) route sources like a rendezvous point or bootstrap router (BSR), or when a route policy is not configured, the current policy default remains in effect. In other words, either a unicast-default or multicast-default table is selected for all sources based on any of the following configurations:

  - Open Shortest Path First (OSPF)

- Intermediate System-to-Intermediate System (IS-IS)
- Multiprotocol Border Gateway Protocol (MBGP)

Note Although both **multicast** and **unicast** keywords are available when using the **address-family** {**ipv4** | **ipv6**} command in routing policy language (RPL), only topologies under multicast SAFI can be configured globally.

Information About Multitopology Routing

Configuring multitopology networks requires the following tasks:

- "Configuring a Global Topology and Associating It with an Interface" (required)

  For information, see *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide*.
- "Enabling an IS-IS Topology" (required)

  For information, see *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide*.
- "Placing an Interface in a Topology in IS-IS" (required)

  For information, see *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide*.
- "Configuring a Routing Policy" (required)

  For information, see *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide*.

Configuring an RPF Topology in PIM

SUMMARY STEPS

1. **configure**
2. **router pim address-family** {**ipv4** | **ipv6**}
3. **rpf topology route-policy** *policy-name*
4. **exit**
5. **multicast-routing address-family** {**ipv4** | **ipv6**}
6. **interface all enable**
7. **commit**
8. **show pim [vrf** *vrf-name*] [**ipv4** | **ipv6**] [{**unicast** | **multicast** | **safi-all**} **topology** {*table-name* | **all**}] **rpf** [*ip-address* | **hash** | **summary** | **route-policy**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

	Command or Action	Purpose

Step 2	**router pim address-family** {**ipv4** | **ipv6**}
	
	Example:
	
	`RP/0/RSP0/CPU0:router(config)#`
	`RP/0/RSP0/CPU0:router(config-pim-default-ipv4)#`
		Enters PIM address-family configuration submode for the IP prefix you select.

Step 3	**rpf topology route-policy** *policy-name* **Example:** `RP/0/RSP0/CPU0:router(config-pim-default-ipv)# rpf topology route-policy mtpolicy`	Assigns a given routing policy to an RPF topology table.

Step 4	**exit**
	
	Example:
	
	`RP/0/RSP0/CPU0:router(config-pim-default-ipv6)# exit`
	`RP/0/RSP0/CPU0:router(config)#`
		Exits pim address-family configuration submode.

Step 5	**multicast-routing address-family** {**ipv4** | **ipv6**} **Example:** `RP/0/RSP0/CPU0:router(config)# multicast-routing address-family ipv4`	Enters multicast address-family configuration submode.
Step 6	**interface all enable** **Example:** `RP/0/RSP0/CPU0:router(config-mcast-default- ipv4))# interface all enable`	Enables multicast routing and forwarding on all new and existing interfaces.
Step 7	**commit**	
Step 8	**show pim [vrf** *vrf-name*] [**ipv4** | **ipv6**] [{**unicast** | **multicast** | **safi-all**} **topology** {*table-name* | **all**}] **rpf** [*ip-address* | **hash** | **summary** | **route-policy**] **Example:** `RP/0/RSP0/CPU0:router# show pim vrf mtt rpf ipv4 multicast topology all rpf`	Shows PIM RPF entries for one or more tables.

Configuring MVPN Extranet Routing

To be able to import unicast routes from source VRFs to receiver VRFs, the import route targets of receiver VRFs must match the export route targets of a source VRF. Also, all VRFs on the PEs where the extranet source-receiver switchover takes place should be added to the BGP router configuration on those PEs.

Configuring MVPN extranet routing consists of these mandatory and optional tasks, which should be performed in the sequence shown:

- "Configuring a Routing Policy" (required only if performing the following task)

  For information, see *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide*.

For examples of an end-to-end configuration of each of the two available MVPN extranet topology solutions, see Configuring MVPN Extranet Routing: Example, on page 204.

Prerequisites for MVPN Extranet Routing

- PIM-SM and PIM-SSM are supported. You must configure the multicast group range in the source and receiver VRFs with a matching PIM mode.
- Because only static RP configuration is currently supported for a given multicast group range, both source and receiver MVRFs must be configured with the same RP.
- In the IPv6 Connectivity over MVPN topology model, the data MDT encapsulation range should be large enough to accommodate extranet streams without any aggregation. This prevents extranet traffic, flowing to multiple VRFs, from being carried into only one data MDT.
- Data MDT configuration is required on only the Source VRF and Source PE Router.

Restrictions for MVPN Extranet Routing

- PIM-DM and PIM-BIDIR are not supported.
- Cisco IOS XR Software software supports only IPv4 extranet multicast routing over IPv4 core multicast routing.
- Any PE can be configured as an RP except a PE in the "Receiver VRF on the Source PE Router" model where the extranet switchover occurs, and where the source VRF has no interfaces. This is because the source VRF must have some physical interface to signal the data packets being received from the first hop.
- Cisco IOS XR Software currently supports only one encapsulation of VRF traffic on an extranet. This means that only one encapsulation interface (or MDT) is allowed in the outgoing forwarding interface list of the multicast route. If, for a given stream, there are multiple receiver VRFs joining the same source VRF, only the first receiver VRF receives traffic; other receiver VRF joins are discarded.

Note This limitation applies only to IPv6 Connectivity over MVPN topology model.

Configuring VPN Route Targets

This procedure demonstrates how to configure a VPN route target for each topology.

Note Route targets should be configured so that the receiver VRF has unicast reachability to prefixes in the source VRF. These configuration steps can be skipped if prefixes in the source VRF are already imported to the receiver VRF.

SUMMARY STEPS

1. **configure**
2. **vrf** *source-vrf*
3. **address-family** [**ipv4** | **ipv6**} **unicast**
4. **import route-target** [*xx.yy:nn* | *as-number:nn* | *ip-address:nn*]
5. **export route-target** [*xx.yy:nn* | *as-number:nn* | *ip-address:nn*]
6. **commit**
7. **configure**
8. **vrf** *receiver-vrf*
9. Repeat Step 3 through Step 6.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

Step 2	**vrf** *source-vrf*
	
	Example:
	`RP/0/RSP0/CPU0:router(config)# vrf green`
	`RP/0/RSP0/CPU0:router(config-vrf)#`
		Configures a VRF instance for the source PE router.

Step 3	**address-family** [**ipv4** | **ipv6**} **unicast**
	
	Example:
	`RP/0/RSP0/CPU0:router(config-vrf)#`
	`address-family ipv4 unicast`
		Specifies a unicast IPv4 or IPv6 address family and enters address family configuration submode.
		Note Only IPv4 addressing is supported for extranet.

Step 4	**import route-target** [*xx.yy:nn* | *as-number:nn* | *ip-address:nn*]
	
	Example:
	`RP/0/RSP0/CPU0:router(config-vrf-af)# import`
	`route-target 234:222`
	`RP/0/RSP0/CPU0:router(config-vrf-af)# import`
		Imports the selected route target, optionally expressed as one of the following :
		• 4-byte AS number of the route target in *xx.yy:nn* format. Range is 0-65535.0-65535:0-65535
		• AS number of the route target in *nn* format. Range is 0-65535.

	Command or Action	Purpose
	`route-target 100:100`	• IP address of the route target in *A.B.C.D.* format.
Step 5	**export route-target** [*xx.yy:nn* | *as-number:nn* | *ip-address:nn*] **Example:** `RP/0/RSP0/CPU0:router(config-vrf-af)# export route-target 100:100`	Exports the selected route target, optionally expressed as one of the following: • 4-byte AS number of the route target in *xx.yy:nn* format. Range is 0-65535.0-65535:0-65535 • AS number of the route target in *nn* format. Range is 0-65535. • IP address of the route target in *A.B.C.D.* format.
Step 6	**commit**	
Step 7	**configure**	

Step 8	**vrf** *receiver-vrf*
	
	Example:
	
	`RP/0/RSP0/CPU0:router(config)# vrf red`
	`RP/0/RSP0/CPU0:router(config-vrf)#`
		Configures a VRF instance for the receiver PE router.

Step 9	Repeat Step 3 through Step 6.	—

Interconnecting PIM-SM Domains with MSDP

To set up an MSDP peering relationship with MSDP-enabled routers in another domain, you configure an MSDP peer to the local router.

If you do not want to have or cannot have a BGP peer in your domain, you could define a default MSDP peer from which to accept all Source-Active (SA) messages.

Finally, you can change the Originator ID when you configure a logical RP on multiple routers in an MSDP mesh group.

Before You Begin

You must configure MSDP default peering, if the addresses of all MSDP peers are not known in BGP or multiprotocol BGP.

SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. **ipv4 address** *address mask*
4. **exit**
5. **router msdp**
6. **default-peer** *ip-address* [**prefix-list** *list*]
7. **originator-id** *type interface-path-id*
8. **peer** *peer-address*
9. **connect-source** *type interface-path-id*
10. **mesh-group** *name*
11. **remote-as** *as-number*
12. **commit**
13. **show msdp** [**ipv4**] **globals**
14. **show msdp** [**ipv4**] **peer** [*peer-address*]
15. **show msdp** [**ipv4**] **rpf** *rpf-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config)# interface loopback 0`	(Optional) Enters interface configuration mode to define the IPv4 address for the interface. **Note** This step is required if you specify an interface type and number whose primary address becomes the source IP address for the TCP connection.
Step 3	**ipv4 address** *address mask* **Example:** `RP/0/RSP0/CPU0:router(config-if)# ipv4 address 10.0.1.3 255.255.255.0`	(Optional) Defines the IPv4 address for the interface. **Note** This step is required only if you specify an interface type and number whose primary address becomes the source IP address for the TCP connection. See optional for information about configuring the **connect-source** command.
Step 4	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-if)# end`	Exits interface configuration mode.

	Command or Action	Purpose
Step 5	**router msdp** **Example:** `RP/0/RSP0/CPU0:router(config)# router msdp`	Enters MSDP protocol configuration mode.
Step 6	**default-peer** *ip-address* [**prefix-list** *list*] **Example:** `RP/0/RSP0/CPU0:router(config-msdp)# default-peer 172.23.16.0`	(Optional) Defines a default peer from which to accept all MSDP SA messages.
Step 7	**originator-id** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config-msdp)# originator-id GigabitEthernet0/1/1/0`	(Optional) Allows an MSDP speaker that originates a (Source-Active) SA message to use the IP address of the interface as the RP address in the SA message.
Step 8	**peer** *peer-address* **Example:** `RP/0/RSP0/CPU0:router(config-msdp)# peer 172.31.1.2`	Enters MSDP peer configuration mode and configures an MSDP peer. • Configure the router as a BGP neighbor. • If you are also BGP peering with this MSDP peer, use the same IP address for MSDP and BGP. You are not required to run BGP or multiprotocol BGP with the MSDP peer, as long as there is a BGP or multiprotocol BGP path between the MSDP peers.
Step 9	**connect-source** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config-msdp-peer)# connect-source loopback 0`	(Optional) Configures a source address used for an MSDP connection.
Step 10	**mesh-group** *name* **Example:** `RP/0/RSP0/CPU0:router(config-msdp-peer)# mesh-group internal`	(Optional) Configures an MSDP peer to be a member of a mesh group.
Step 11	**remote-as** *as-number* **Example:** `RP/0/RSP0/CPU0:router(config-msdp-peer)# remote-as 250`	(Optional) Configures the remote autonomous system number of this peer.

	Command or Action	Purpose
Step 12	**commit**	
Step 13	**show msdp** [**ipv4**] **globals** **Example:** `RP/0/RSP0/CPU0:router# show msdp globals`	Displays the MSDP global variables.
Step 14	**show msdp** [**ipv4**] **peer** [*peer-address*] **Example:** `RP/0/RSP0/CPU0:router# show msdp peer 172.31.1.2`	Displays information about the MSDP peer.
Step 15	**show msdp** [**ipv4**] **rpf** *rpf-address* **Example:** `RP/0/RSP0/CPU0:router# show msdp rpf 172.16.10.13`	Displays the RPF lookup.

Controlling Source Information on MSDP Peer Routers

Your MSDP peer router can be customized to control source information that is originated, forwarded, received, cached, and encapsulated.

When originating Source-Active (SA) messages, you can control to whom you will originate source information, based on the source that is requesting information.

When forwarding SA messages you can do the following:

- Filter all source/group pairs
- Specify an extended access list to pass only certain source/group pairs
- Filter based on match criteria in a route map

When receiving SA messages you can do the following:

- Filter all incoming SA messages from an MSDP peer
- Specify an extended access list to pass certain source/group pairs
- Filter based on match criteria in a route map

In addition, you can use time to live (TTL) to control what data is encapsulated in the first SA message for every source. For example, you could limit internal traffic to a TTL of eight hops. If you want other groups to go to external locations, you send those packets with a TTL greater than eight hops.

By default, MSDP automatically sends SA messages to peers when a new member joins a group and wants to receive multicast traffic. You are no longer required to configure an SA request to a specified MSDP peer.

SUMMARY STEPS

1. **configure**
2. **router msdp**
3. **sa-filter** {**in** | **out**} {*ip-address* | *peer-name*} [**list** *access-list-name*] [**rp-list** *access-list-name*]
4. **cache-sa-state** [**list** *access-list-name*] [**rp-list** *access-list-name*]
5. **ttl-threshold** *ttl-value*
6. **exit**
7. **ipv4 access-list** *name* [*sequence-number*] **permit** *source* [*source-wildcard*]
8. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router msdp** **Example:** `RP/0/RSP0/CPU0:router(config)# router msdp`	Enters MSDP protocol configuration mode.
Step 3	**sa-filter** {**in** | **out**} {*ip-address* | *peer-name*} [**list** *access-list-name*] [**rp-list** *access-list-name*] **Example:** `RP/0/RSP0/CPU0:router(config-msdp)# sa-filter out router.cisco.com list 100`	Configures an incoming or outgoing filter list for messages received from the specified MSDP peer. • If you specify both the **list** and **rp-list** keywords, all conditions must be true to pass any source, group (S, G) pairs in outgoing Source-Active (SA) messages. • You must configure the **ipv4 access-list** command in Step 7, on page 160. • If all match criteria are true, a **permit** from the route map passes routes through the filter. A **deny** filters routes. • This example allows only (S, G) pairs that pass access list 100 to be forwarded in an SA message to the peer named router.cisco.com.
Step 4	**cache-sa-state** [**list** *access-list-name*] [**rp-list** *access-list-name*] **Example:** `RP/0/RSP0/CPU0:router(config-msdp)# cache-sa-state 100`	Creates and caches source/group pairs from received Source-Active (SA) messages and controls pairs through access lists.

	Command or Action	Purpose
Step 5	**ttl-threshold** *ttl-value* **Example:** `RP/0/RSP0/CPU0:router(config-msdp)# ttl-threshold 8`	(Optional) Limits which multicast data is sent in SA messages to an MSDP peer. • Only multicast packets with an IP header TTL greater than or equal to the *ttl-value* argument are sent to the MSDP peer specified by the IP address or name. • Use this command if you want to use TTL to examine your multicast data traffic. For example, you could limit internal traffic to a TTL of 8. If you want other groups to go to external locations, send those packets with a TTL greater than 8. • This example configures a TTL threshold of eight hops.
Step 6	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-msdp)# exit`	Exits the current configuration mode.
Step 7	**ipv4 access-list** *name* [*sequence-number*] **permit** *source* [*source-wildcard*] **Example:** `RP/0/RSP0/CPU0:router(config)# ipv4 access-list 100 20 permit 239.1.1.1 0.0.0.0`	Defines an IPv4 access list to be used by SA filtering. • In this example, the access list 100 permits multicast group 239.1.1.1. • The **ipv4 access-list** command is required if the keyword **list** is configured for SA filtering in Step 3, on page 159.
Step 8	**commit**	

Configuring MSDP MD5 Password Authentication

SUMMARY STEPS

1. **configure**
2. **router msdp**
3. **peer** *peer-address*
4. **password** {**clear** | **encrypted**} *password*
5. **commit**
6. **show mfib** [**vrf** *vrf-name*] [**ipv4** | **ipv6**] **hardware route** {* | *source-address* | *group-address*[/*prefix-length*]} **location** *node-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router msdp** **Example:** `RP/0/RSP0/CPU0:router(config)# router msdp`	Enters MSDP configuration mode.
Step 3	**peer** *peer-address* **Example:** `RP/0/RSP0/CPU0:router(config-msdp)# peer 10.0.5.4`	Configures the MSDP peer.
Step 4	**password** {**clear** | **encrypted**} *password* **Example:** `RP/0/RSP0/CPU0:router(config-msdp-peer)# password encrypted a34bi5m`	Configures the password.
Step 5	**commit**	
Step 6	**show mfib** [**vrf** *vrf-name*] [**ipv4** | **ipv6**] **hardware route** {***** | *source-address* | *group-address*[/*prefix-length*]} **location** *node-id* **Example:** `RP/0/RSP0/CPU0:router# show mfib hardware route * location 0/1/cpu0`	Displays multicast routes configured with multicast QoS and the associated parameters.

Configuring VRF for MSDP

Use the **vrf** keyword in the MSDP configuration mode to enable VRF for MSDP.

SUMMARY STEPS

1. **configure**
2. **router msdp**
3. **vrf** *vrf-name*
4. **peer** *peer-address*
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router msdp** **Example:** `RP/0/RSP0/CPU0:router(config)# router msdp`	Enters MSDP configuration mode.
Step 3	**vrf** *vrf-name* **Example:** `RP/0/RSP0/CPU0:router(config-msdp) # vrf vrf1`	Enables VRF configuration for MSDP.
Step 4	**peer** *peer-address* **Example:** `RP/0/RSP0/CPU0:router(config-msdp) # peer 1.1.1.1`	Configures the VRF MSDP peer .
Step 5	**commit**	

Multicast only fast reroute (MoFRR)

MoFRR allows fast reroute for multicast traffic on a multicast router. MoFRR minimizes packet loss in a network when node or link failures occur(at the topology merge point). It works by making simple enhancements to multicast routing protocols.

MoFRR involves transmitting a multicast join message from a receiver towards a source on a primary path and transmitting a secondary multicast join message from the receiver towards the source on a backup path. Data packets are received from the primary and secondary paths. The redundant packets are discarded at topology merge points with the help of Reverse Path Forwarding (RPF) checks. When a failure is detected on the primary path, the repair occurs locally by changing the interface on which packets are accepted to the secondary interface, thus improving the convergence times in the event of a node or link failure on the primary path.

MoFRR supports ECMP (Equal Cost Multipath) and non-ECMP topologies as well.

TI (Topology Independent) MoFRR is a multicast feature that performs fast convergence (Fast ReRoute) for specified routes/flows when failure is detected on one of the paths between the router and the source.

Operating Modes of MoFRR

- Flow-based MoFRR— exposes the primary and secondary RPF interfaces to the forwarding plane, with switchover occurring entirely at the hardware level.

Faster convergence is obtainable in Flow-based MoFRR by monitoring the packet counts of the primary stream. If no activity is detected for 30 ms, the switch over is triggered to the backup stream and the traffic loss is within 50 ms.

Restrictions

These limitations apply to MoFRR deployments when the Cisco ASR 9000 Series SPA Interface Processor-700 linecard is used in the Cisco ASR 9000 Series Router chassis.

1. Cisco ASR 9000 Series SPA Interface Processor-700 cannot be used on ingress interface as either the primary or backup (ECMP paths) path back to the multicast source.
2. The egress interfaces on Cisco ASR 9000 Series SPA Interface Processor-700 may lead to duplicate multicast streams for short periods of time (the time between the switch from Trident primary to Trident backup paths on ingress).

Non-ECMP MoFRR

TI (Topology-Independent) MoFRR is a multicast feature that performs fast convergence (Fast ReRoute) for specified routes/flows when failure is detected on one of the paths between the router and the source.

Flow based non-ECMP approach uses a mechanism where two copies of the same multicast stream flow through disjoint paths in the network. At the point in the network (usually the tail PE that is closer to the receivers) where the two streams merge, one of the streams is accepted and forwarded on the downstream links, while the other stream is discarded. When a failure is detected in the primary stream due to a link or node failure in the network, MoFRR instructs the forwarding plane to start accepting packets from the backup stream (which now becomes the primary stream).

For more information about topology independent MoFRR, refer the *Cisco ASR 9000 Series Aggregation Services Router Multicast Configuration Guide*.

Implementing Non-ECMP MoFRR

The config handler in PIM creates a mapping between (S1, G) and (S2, G) in an internal mapping database. No explicit route is created till a downstream join / data signal is received for (S1, G).

Downstream (S, G) join

The tail PE on receipt of (S, G) JOIN looks up the mapping database and,

- Creates the (S1, G) route entry with proxy info and marks it as primary mofrr route.
- Creates the (S2, G) route entry with the proxy info and marks it as backup mofrr route.
- Creates reference to (S2, G) from (S1, G) route and vice versa.
- Redistributes route with MoFRR primary & backup flags to PD.

Downstream (S,G) prune

The tail PE on receipt of (S, G) PRUNE looks up the mapping database and,

- ◦ Deletes the (S1, G) route entry with proxy info and redistributes the route delete.
  ◦ Deletes the (S2, G) route entry with proxy info and redistributes the route delete.

Data Signaling

Head PE on receipt of (S, G) traffic will clone the traffic as (S1, G) and (S2, G) and send it out on the interfaces on which (S1, G) / (S2, G) join has been received. This is done because the (S, G) entry is created with an encap-id with 2 encap-oles corresponding to (S1, G) and (S2, G).

On Tail PE on receipt of (S1, G) traffic the header is replaced as (S, G) and sent out on the interfaces on which (S, G) join has been received. If traffic is not received on (S1, G) on tail node for 50 ms then ucode initiates a switchover event and starts accepting traffic on (S2, G) and sends switchover notifications to control plane.

Configuring MoFRR

RIB-based MoFRR

SUMMARY STEPS

1. **configure**
2. **router pim**
3. **mofrr rib** *acl-name*
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router pim** **Example:** `RP/0/RSP0/CPU0:router(config)# router pim`	Enters the PIM configuration mode.
Step 3	**mofrr rib** *acl-name* **Example:** `RP/0/RSP0/CPU0:router(pim)# mofrr rib acl1`	Enter the ACL name.
Step 4	**commit**	

Flow-based MoFRR

SUMMARY STEPS

1. **configure**
2. **ipv4 access-list** *acl-name*
3. *sequence number* [**permit**|**deny**] **ipv4 host** *address* [**host** *address* | **any**]
4. **exit**
5. **router pim**
6. **mofrr** *acl-name*
7. **commit**
8. **show mfib hardware route summary location**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**ipv4 access-list** *acl-name* **Example:** `RP/0/RSP0/CPU0:router (config)# ipv4 access-list flow_mofrr`	Enters IPv4 access list configuration mode and configures the named access list.
Step 3	*sequence number* [**permit**|**deny**] **ipv4 host** *address* [**host** *address*	**any**] **Example:** `RP/0/RSP0/CPU0:router(config-ipv4-acl) #10 permit ipv4 host 20.0.0.2 any`	Specifies one or more conditions allowed or denied in the created IPv4 access list.
Step 4	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-ipv4-acl)# exit`	Saves the MoFRR acl configuration and exists the IPv4 acl configuration mode. You need to **exit** twice here.
Step 5	**router pim** **Example:** `RP/0/RSP0/CPU0:router(config)# router pim`	Enters the PIM configuration mode.

	Command or Action	Purpose
Step 6	**mofrr** *acl-name* **Example:** `RP/0/RSP0/CPU0:router(pim)# mofrr flow_mofrr`	Enables MoFRR for the specified access list source group with hardware switchover triggers. This is supported on IPv4 only.
Step 7	**commit**	
Step 8	**show mfib hardware route summary location** **Example:** `RP/0/RSP0/CPU0:router#` **show mfib hardware route 4**	Displays the number of enabled MoFRR routes.

Configuring Head PE Router (for MoFRR)

Pre-requisites

- ACL configurations. (for detailed information on how to configure ACLs, refer the Configuring ACLs chapter of the IP Addresses and Services configuration guide.)

The head PE router can be configured as follows:

SUMMARY STEPS

1. **configure**
2. **router pim** [**address-family ipv4**]
3. **mofrr**
4. **mofrr** *acl-name*
5. **clone source S to S1 S2masklen***n*
6. **commit**
7. **show pim topology** *route*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router pim** [**address-family ipv4**] **Example:** `RP/0/RSP0/CPU0:router(config)# router pim`	Enters PIM configuration mode, or PIM address-family configuration submode.

	Command or Action	Purpose
Step 3	**mofrr** **Example:** `RP/0/RSP0/CPU0:router(config-pim)# mofrr`	Enters PIM Multicast only FRR configuration mode.
Step 4	**mofrr** *acl-name* **Example:** `RP/0/RSP0/CPU0:router(config-pim-ipv4-mofrr)# flow acl1`	Enables MoFRR with hardware switchover triggers for the specified access-list.
Step 5	**clone source S to S1 S2masklen***n* **Example:** `RP/0/RSP0/CPU0:router(config-pim-ipv4-mofrr)# clone source 10.1.1.1 to 20.2.2.2 50.5.5.5 masklen 32`	Duplicates source (S) to S1 and S2 with the specified mask length. A mapping is created between (S,G) , (S1,G) and (S2,G). S1 is the primary path and S2 is the secondary path.
Step 6	**commit**	
Step 7	**show pim topology** *route* **Example:** `RP/0/RSP0/CPU0:router# show pim topology 232.0.0.1`	This command verifies the mapping between the source S and S1 and S2. S, S1, S2 entries are updated in the displayed MoFRR details.

Configuring Tail PE Router (for MoFRR)

•

SUMMARY STEPS

1. **configure**
2. **router pim** [**address-family ipv4**]
3. **mofrr**
4. **mofrr** *acl-name*
5. **clone join S to S1 S2masklen***length*
6. **rpf-vector** *source***masklen***length*
7. **commit**
8. **show mfib hardware router mofrr** *route***location***interface-path-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router pim** [**address-family ipv4**] **Example:** `RP/0/RSP0/CPU0:router(config)# router pim`	Enters PIM configuration mode, or PIM address-family configuration submode.
Step 3	**mofrr** **Example:** `RP/0/RSP0/CPU0:router(config-pim)# mofrr`	Enters PIM Multicast only FRR configuration mode.
Step 4	**mofrr** *acl-name* **Example:** `RP/0/RSP0/CPU0:router(config-pim-ipv4-mofrr)# flow acl1`	Enables MoFRR with hardware switchover triggers for the specified access-list.
Step 5	**clone join S to S1 S2masklen***length* **Example:** `RP/0/RSP0/CPU0:router(config-pim-ipv4-mofrr)# clone join 10.1.1.1 to 20.2.2.2 50.5.5.5 masklen 32`	Duplicates source to S1 and S2 with the specified mask length. A mapping is created between (S,G) ,(S1,G) and (S2,G).
Step 6	**rpf-vector** *source***masklen***length* **Example:** `RP/0/RSP0/CPU0:router(config-pim-ipv4-mofrr)# rpf-vector 10.1.1.1 masklen 10 R1, R2.....`	Configures the reachability of the tail-node.
Step 7	**commit**	
Step 8	**show mfib hardware router mofrr** *route***location***interface-path-id* **Example:** `RP/0/RSP0/CPU0:router# show mfib hardware router mofrr 232.0.0.1 location 0/1/1/1`	This command verifies the mapping between the source S and S1 and S2. S, S1, S2 entries are updated in the displayed MoFRR details.

Enabling multicast on PW-HE interfaces

This task enables multicast on PW-HE interfaces.

SUMMARY STEPS

1. **configure**
2. **multicast-routing** [**address-family** {**ipv4** | **ipv6**}]
3. **interface pw-ether1**
4. **enable**
5. exit
6. **vrf** *vrf-name*
7. **address-family ipv4**
8. **interface** *type interface path-id*
9. **enable**
10. exit
11. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**multicast-routing** [**address-family** {**ipv4** | **ipv6**}] **Example:** `RP/0/RSP0/CPU0:router(config)# multicast-routing address-family ipv4`	Enters multicast routing configuration mode.
Step 3	**interface pw-ether1** **Example:** `RP/0/RSP0/CPU0:router(config-mcast-ipv4)# interface pw-ether1`	Enters the pseudowire interface configuration mode.
Step 4	**enable** **Example:** `RP/0/RSP0/CPU0:router(config-mcast-ipv4)# enable`	Enables multicast routing on pseudowire interfaces.
Step 5	exit **Example:** `RP/0/RSP0/CPU0:router(config-mcast-ipv4)# exit`	Exits the current configuration mode.

	Command or Action	Purpose
Step 6	**vrf** *vrf-name* **Example:** `RP/0/RSP0/CPU0:router (config-mcast) # vrf v1`	Enters the vrf configuration mode.
Step 7	**address-family ipv4** **Example:** `RP/0/RSP0/CPU0:router(config)# address-family ipv4`	Enters the IPv4 address-family configuration mode.
Step 8	**interface** *type interface path-id* **Example:** `RP/0/RSP0/CPU0:router (config-mcast-vrf-v4) # interface pw-ether2`	Enters the vrf mode for the specified pw interface.
Step 9	**enable** **Example:** `RP/0/RSP0/CPU0:router(config-mcast-ipv4)# enable`	Enables multicast routing on the pw interface in the vrf.
Step 10	exit **Example:** `RP/0/RSP0/CPU0:router(config-mcast-ipv4)# exit`	Exits the current configuration mode. **Note** This step can be used, more than once.
Step 11	**commit**	

Static join

The static join can be achieved with IGMP or MLD. The **router mld** or **router igmp** commands can be used to enter the MLD or IGMP modes respectively. The examples section (later in this chapter) includes the examples for both the cases.

SUMMARY STEPS

1. **configure**
2. **router mld**
3. **interface** *type interface-path-id*
4. **static-group** *ip-group-address source-address*
5. exit
6. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router mld** **Example:** `RP/0/RSP0/CPU0:router(config)# router mld`	Enters the MLD multicast routing configuration mode.
Step 3	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config-mld)# interface pw-ether1`	Enters the pseudowire interface configuration mode.
Step 4	**static-group** *ip-group-address source-address* **Example:** `RP/0/RSP0/CPU0:router(config-mld-default-if)# static-group ff35::e100 2000:10::1`	Enables pw-ether1 interface to statistically join a multicast group.
Step 5	exit **Example:** `RP/0/RSP0/CPU0:router(config-mcast-ipv4)# exit`	Exits the current configuration mode. **Note** This step can be used, more than once.
Step 6	**commit**	

Configuring Route Policy for Static RPF

SUMMARY STEPS

1. **configure**
2. **router static**
3. **address-family**[**ipv4** | **ipv6**][**multicast** |**unicast**]*destination prefix interface-typeinterface-path-id*
4. **exit**
5. **route-policy***policy-name*
6. **set rpf-topology** *policy-name***address-family**[**ipv4** |**ipv6**]**multicast** | **unicasttopology***name*
7. **end route-policy**
8. **router pim address-family**[**ipv4** |**ipv6**]
9. **rpf topology route-policy***policy-name***pim policy**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router static** **Example:** `RP/0/RSP0/CPU0:router(config) # router static`	Enables a static routing process.
Step 3	**address-family**[**ipv4**	**ipv6**][**multicast** |**unicast**]*destination prefix interface-typeinterface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config-static) # address-family ipv4 multicast 202.93.100.4/ 32 202.95.1.1`	Configures the ipv4 multicast address-family topology with a destination prefix.
Step 4	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-ipv4-afi) # exit`	Exits from the address family configuration mode.
Step 5	**route-policy***policy-name* **Example:** `RP/0/RSP0/CPU0:router(config) # route-policy r1`	Configures the route policy to select the RPF topology.
Step 6	**set rpf-topology** *policy-name***address-family**[**ipv4** |**ipv6**]**multicast**	**unicasttopology***name* **Example:** `RP/0/RSP0/CPU0:router(config-rpl) # set rpf-topology p1 ipv4 multicast topology t1`	Configures the PIM rpf-topology attributes for the selected multicast address-family.

	Command or Action	Purpose
Step 7	**end route-policy** **Example:** `RP/0/RSP0/CPU0:router(config-rpl) #  end route-policy r1`	Ends the route policy.
Step 8	**router pim address-family**[**ipv4** |**ipv6**] **Example:** `RP/0/RSP0/CPU0:router(config) # router pim address-family ipv4`	Enters the PIM address-family configuration sub-mode.
Step 9	**rpf topology route-policy***policy-name***pim policy** **Example:** `RP/0/RSP0/CPU0:router(config) # rpf topology route-policy r1 pim policy`	Selects the RPF topology for the configured route-policy.

Point-to-Multipoint Traffic Engineering Label-Switched Multicast

IP multicast was traditionally used for IPTV broadcasting and content delivery services. MPLS-TE (traffic engineering) is fast replacing the IP multicast technique because of the various advantages of MPLS-TE, such as:

- Fast re-routing and restoration in case of link/ node failure
- Bandwidth guarantee
- Explicit path setting along with off-line computation

MPLS supports point-to-point path. However, in order to use MPLS for multicast service, MPLS has to be extended to handle point-to-multipoint paths. A reliable solution to signal Point-to-Multipoint (P2MP) label switched paths(LSP) is the Point-to-Multipoint TE LSP. This solution uses the Resource Reservation Protocol-Traffic Engineering (RSVP-TE) extension as the signaling protocol for establishing P2MP TE LSPs.

Point to Multipoint LSP(P2MP)

P2MP LSP is unidirectional. In case of native IP multicast, the multicast forwarding always has to perform an acceptance check. This check ensures all multicast packets undergo a RPF check to ensure that the packets have arrived on the correct interface in the direction of the source. However, the acceptance check with MPLS forwarding may be different in case of an unicast or upstream label.

Depending on the multicast signaling protocol, the labeled packet may require an additional L3 lookup at the P and PE routers in order to forward the multicast packet to the physical interfaces according to multicast routing. In this case, the incoming P2MP LSP as the incoming interface for the received multicast packet must

also be available to the multicast forwarding plane during the L3 lookup. For more details on RSVP-TE and P2MP LSP, refer the *Cisco ASR 9000 Series Aggregation Services Router MPLS Configuration Guide*

Multicast Routing Protocol support for P2MP

All multicast routing protocols support P2MP TE LSP. At ingress node, a multicast protocol must make a mapping between the multicast traffic and the P2MP TE LSP with the configuration of static-join. At egress node, the multicast protocol must conduct a special RPF check for the multicast packet which is received from MPLS core and forward it to the customer facing interface. The RPF check is based on the configuration of static-rpf. These multicast groups which are forwarded over the P2MP TE LSPs can be specified with the static-rpf configuration in case of PIM-SSM.

Enabling Multicast Forwarding Over Tunnel Interface (at Ingress Node)

This configuration is used for allowing the forwarding of the multicast packet over the specified interface.

SUMMARY STEPS

1. **configure**
2. **multicast-routing**
3. **address-family** {**ipv4**|**ipv6**}
4. **interface tunnel-mte** *range*
5. **enable** | **disable**
6. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**multicast-routing** **Example:** `RP/0/RSP0/CPU0:router(config)# multicast-routing`	Enters multicast routing configuration mode.
Step 3	**address-family** {**ipv4**|**ipv6**} **Example:** `RP/0/RSP0/CPU0:router(config-mcast)# address-family ipv4`	Enters ipv4 or ipv6 address-family submode.
Step 4	**interface tunnel-mte** *range* **Example:** `RP/0/RSP0/CPU0:router(config-mcast-default-ipv4)#`	Specify the range. The range is 0 to 65535.

	Command or Action	Purpose
	`interface tunnel-mte 100`	
Step 5	**enable** | **disable** **Example:** `RP/0/RSP0/CPU0:router(config-mcast-default-ipv4)# enable`	If enable is set, MFIB forwards multicast packets over the interface. If disable is set, MFIB stops forwarding multicast packets over the interface.
Step 6	**commit**	

P2MP configurations at egress node and bud node

Configuring Static Reverse Path Forwarding (RPF)

SUMMARY STEPS

1. **configure**
2. **multicast-routing**
3. **address-family** {**ipv4** | **ipv6**}
4. **static-rpf** *address range prefix*
5. **mpls address**
6. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**multicast-routing** **Example:** `RP/0/RSP0/CPU0:router(config)# multicast-routing`	Enters multicast routing configuration mode.
Step 3	**address-family** {**ipv4** | **ipv6**} **Example:** `RP/0/RSP0/CPU0:router(config-mcast)# address-family ipv4`	Enters ipv4 (or ipv6) address-family submode.

	Command or Action	Purpose
Step 4	**static-rpf** *address range prefix* **Example:** `RP/0/RSP0/CPU0:router(config-mcast-default-ipv4)# static-rpf 10.1.1.1 32`	Enter the source and prefix length.
Step 5	**mpls address** **Example:** `RP/0/RSP0/CPU0:router(config-mcast-default-ipv4)# mpls 10.2.2.2`	Enter the source PE address of the MPLS P2MP tunnel.
Step 6	**commit**	

Configuring Core Tree Protocol

SUMMARY STEPS

1. **configure**
2. **multicast-routing**
3. **address-family** {**ipv4** | **ipv6**}
4. **core-tree-protocol rsvp-te group-list** *name*
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**multicast-routing** **Example:** `RP/0/RSP0/CPU0:router(config)# multicast-routing`	Enters multicast routing configuration mode.
Step 3	**address-family** {**ipv4**	**ipv6**} **Example:** `RP/0/RSP0/CPU0:router(config-mcast)# address-family ipv4`	Enters ipv4 (or ipv6)address-family submode.

	Command or Action	Purpose
Step 4	**core-tree-protocol rsvp-te group-list** *name* **Example:** `RP/0/RSP0/CPU0:router(config-mcast-default-ipv4)# core-tree-protocol rsvp-te group-list acl1`	Enters the core-tree-protocol configuration mode.
Step 5	**commit**	

Configuring IGMP VRF Override

This process consists of the following tasks:

Specifying VRF definition

SUMMARY STEPS

1. **configure**
2. **vrf** *vrf-name*
3. **address-family ipv4 unicast**
4. **import route-target 1:1**
5. **export route-target 1:1**
6. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**vrf** *vrf-name* **Example:** `RP/0/RSP0/CPU0:router(config)# vrf name1`	Enters the VRF configuration sub mode.
Step 3	**address-family ipv4 unicast** **Example:** `RP/0/RSP0/CPU0:router(config-vrf)# address-family ipv4 unicast`	AFI configuration for IPv4. This is supported on unicast topologies only.

	Command or Action	Purpose
Step 4	**import route-target 1:1** **Example:** `RP/0/RSP0/CPU0:router(config-vrf-af)# import route-target 1:1`	Enables VRF import.
Step 5	**export route-target 1:1** **Example:** `RP/0/RSP0/CPU0:router(config-vrf-af)# export route-target 1:1`	Enables VRF export.
Step 6	**commit**	

Enabling Multicast Routing on default and non-default VRFs

This task enables multicast routing and forwarding on all new and existing interfaces. For the VRF override feature, multicast routing needs to be enabled on both, the default and the non-default VRFs.

SUMMARY STEPS

1. **configure**
2. **multicast-routing vrf** [*vrf-name* | *default]*
3. **interface** {*type interface-path-id* | **all**} **enable**
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**multicast-routing vrf** [*vrf-name*	*default]* **Example:** `RP/0/RSP0/CPU0:router(config)# multicast-routing vrf green`	Enters multicast configuration mode for the specified VRF. Note that the default configuration mode for multicast routing is default vrf (if the non-default VRF name is not specified).
Step 3	**interface** {*type interface-path-id*	**all**} **enable** **Example:** `RP/0/RSP0/CPU0:router(config-mcast-green)#`	Enables multicast routing and forwarding on one or on all new and existing interfaces.

	Command or Action	Purpose
	`interface all enable`	
Step 4	**commit**	

Configuring an Interface for a Non-default VRF Instance

SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. **vrf** *vrf-name*
4. **ipv4 address** *address mask*
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config)# interface tengige 0/1/0/0`	Enters PIM address-family IPv4 submode.
Step 3	**vrf** *vrf-name* **Example:** `RP/0/RSP0/CPU0:router(config-if)# vrf name1`	Sets the VRF for the interface.
Step 4	**ipv4 address** *address mask* **Example:** `RP/0/RSP0/CPU0:router(config-if)# ipv4 address 10.1.1.1 255.0.0.0`	Sets the IPv4 address for the interface.
Step 5	**commit**	

Configuring route-policy

SUMMARY STEPS

1. **configure**
2. **route-policy** *policy-name*
3. **set rpf-topology vrf default**
4. **end-policy**
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**route-policy** *policy-name* **Example:** `RP/0/RSP0/CPU0:router(config)# route-policy policy1`	Defines a route policy.
Step 3	**set rpf-topology vrf default** **Example:** `RP/0/RSP0/CPU0:router(config-rpl)# set rpf-topology vrf default`	Sets the PIM RPF topology attributes for the default VRF.
Step 4	**end-policy** **Example:** `RP/0/RSP0/CPU0:router(config-rpl)# end-policy`	Ends the route-policy definition configuration.
Step 5	**commit**	

Associating a route policy to PIM configuration for the VRF receiving IGMP reports

SUMMARY STEPS

1. **configure**
2. **router pim vrf** *vrf-name* **address-family ipv4**
3. **rpf-topology route-policy** *policy-name*
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router pim vrf** *vrf-name* **address-family ipv4**	Enters PIM address-family IPv4 submode.
Step 3	**rpf-topology route-policy** *policy-name* **Example:** `RP/0/RSP0/CPU0:router(config-rpl)# rpf-topology extranet-igmp-reports`	Associates a previously defined route-policy with the non-default VRF that receives the IGMP reports.
Step 4	**commit**	

Configuration Examples for Implementing Multicast Routing on Software

This section provides the following configuration examples:

Calculating Rates per Route: Example

The following example illustrates output from hardware counters based on rate per route for a specific source and group address location:

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# multicast-routing vrf vpn12 address-family ipv4
RP/0/RSP0/CPU0:router(config-mcast-default-ipv4)# rate-per-route
RP/0/RSP0/CPU0:router(config-mcast-default-ipv4)# interface all enable
RP/0/RSP0/CPU0:router(config-mcast-default-ipv4)# accounting per-prefix
RP/0/RSP0/CPU0:router(config-mcast-default-ipv4)# commit
RP/0/RSP0/CPU0:router(config-mcast-default-ipv4)# exit
```

```
RP/0/RSP0/CPU0:router(config-mcast)# exit
RP/0/RSP0/CPU0:router(config)# exit
RP/0/RSP0/CPU0:router# show mfib route rate

IP Multicast Forwarding Rates Source Address, Group Address HW Forwarding Rates: bps In/pps
 In/bps Out/pps Out

(*,224.0.0.0/24)
bps_in /pps_in /bps_out /pps_out
N/A / N/A / N/A / N/A

(*,224.0.1.39)
bps_in /pps_in /bps_out /pps_out
N/A / N/A / N/A / N/A

(*,224.0.1.40)
bps_in /pps_in /bps_out /pps_out
N/A / N/A / N/A / N/A

(*,232.0.0.0/8)
bps_in /pps_in /bps_out /pps_out
N/A / N/A / N/A / N/A
(10.0.70.2,225.0.0.0)
bps_in /pps_in /bps_out /pps_out
22649 / 50 / 22951 / 50

(10.0.70.2,225.0.0.1)
bps_in /pps_in /bps_out /pps_out
22649 / 50 / 22951 / 50

(10.0.70.2,225.0.0.2)
bps_in /pps_in /bps_out /pps_out
22649 / 50 / 22951 / 50

(10.0.70.2,225.0.0.3)
bps_in /pps_in /bps_out /pps_out
22649 / 50 / 22951 / 50

(10.0.70.2,225.0.0.4)
bps_in /pps_in /bps_out /pps_out
22649 / 50 / 22951 / 50

(10.0.70.2,225.0.0.5)
bps_in /pps_in /bps_out /pps_out
22649 / 50 / 22951 / 50

(10.0.70.2,225.0.0.6)
bps_in /pps_in /bps_out /pps_out
```

Preventing Auto-RP Messages from Being Forwarded on Software: Example

This example shows that Auto-RP messages are prevented from being sent out of the GigabitEthernet interface 0/3/0/0. It also shows that access list 111 is used by the Auto-RP candidate and access list 222 is used by the **boundary** command to contain traffic on GigabitEthernet interface 0/3/0/0.

```
ipv4 access-list 111
 10 permit 224.1.0.0 0.0.255.255 any
 20 permit 224.2.0.0 0.0.255.255 any
!
!Access list 111 is used by the Auto-RP candidate.
!
ipv4 access-list 222
 10 deny any host 224.0.1.39
 20 deny any host 224.0.1.40
!
!Access list 222 is used by the boundary command to contain traffic (on
```

```
GigabitEthernet0/3/0/0) that is sent to groups 224.0.1.39 and 224.0.1.40.
!
router pim
 auto-rp mapping-agent loopback 2 scope 32 interval 30
 auto-rp candidate-rp loopback 2 scope 15 group-list 111 interval 30
multicast-routing
 interface GigabitEthernet0/3/0/0
 boundary 222
!
```

Inheritance in MSDP on Software: Example

The following MSDP commands can be inherited by all MSDP peers when configured under router MSDP configuration mode. In addition, commands can be configured under the peer configuration mode for specific peers to override the inheritance feature.

- **connect-source**
- **sa-filter**
- **ttl-threshold**

If a command is configured in both the router msdp and peer configuration modes, the peer configuration takes precedence.

In the following example, MSDP on Router A filters Source-Active (SA) announcements on all peer groups in the address range 226/8 (except IP address 172.16.0.2); and filters SAs sourced by the originator RP 172.16.0.3 to 172.16.0.2.

MSDP peers (172.16.0.1, 172.16.0.2, and 172.17.0.1) use the loopback 0 address of Router A to set up peering. However, peer 192.168.12.2 uses the IPv4 address configured on the GigabitEthernet interface to peer with Router A.

Router A

```
!
ipv4 access-list 111
 10 deny ip host 172.16.0.3 any
 20 permit any any
!

ipv4 access-list 112
 10 deny any 226.0.0.0 0.255.255.255
 30 permit any any
!
router msdp
 connect-source loopback 0
 sa-filter in rp-list 111
 sa-filter out rp-list 111
 peer 172.16.0.1
!
peer 172.16.0.2
 sa-filter out list 112
!
peer 172.17.0.1
!
peer 192.168.12.2
 connect-source GigabitEthernet0/2/0/0
!
```

MSDP-VRF: Example

This is an example where, peer 1.1.1.1 is configured in the VRF context for vrf1.

```
config
 router msdp
  vrf vrf1
    peer 1.1.1.1
  exit
end
!
```

-

MoFRR Provider Edge Configuration: Example

The following example shows Tail PE configuration details. Here, joins for (1.1.1.1, 232.1.1.1) will be sent as joins for (1.1.1.1, 232.1.1.1.1) and joins for (3.3.1.1, 232.1.1.1).

```
config
router pim
mofrr
flow mofrr_acl
join source 1.1.1.1 to 3.3.0.0 masklen 16
rpf-vector 1.1.1.1 masklen 16 10.1.1.1 20.1.1.1
rpf-vector 3.3.1.1 masklen 16 30.1.1.1 40.1.1.1
ipv4 access-list extended mofrr_acl
! 10 permit ipv4 any 232.1.1.1
```

Configuring Route Policy for Static RPF: Example

```
router static
 address-family ipv4 multicast
  202.93.192.74 /32 202.40.148.11

!
route-policy pim-policy
 set rpf-topology ipv4 multicast topology default

end-policy
!
router pim
 address-family ipv4
  rpf topology route-policy pim-policy
```

Configuring IPv4 Multicast VPN: Example

Cisco ASR 9000 Series Routers support only IPv4 addressing.

This end-to-end configuration example shows how to establish a multicast VPN topology (Figure 11: Topology in CE4PE1PE2 CE3MVPN Configuration, on page 185), using two different routing protocols (OSPF or BGP) to broadcasting traffic between customer-edge(CE) routers and provider-edge (PE) routers:

Figure 11: Topology in CE4PE1PE2 CE3MVPN Configuration

CE4---------- PE1 ---------- PE2 ---------- CE3

For more configuration information, see the Configuring Multicast VPN, on page 134 of this module and also related configuration information in *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide* .

Configuring MVPN to Advertise Routes Between the CE and the PE Using OSPF: Example

PE1:

```
!
vrf vpn1
 address-family ipv4 unicast
  import route-target
   1:1
  !
  export route-target
   1:1
  !
 !
!
interface Loopback0
 ipv4 address 1.1.1.1 255.255.255.255
!
interface Loopback1
 vrf vpn1
 ipv4 address 2.2.2.2 255.255.255.255
!
interface GigabitEthernet0/5/0/0
 vrf vpn1
 ipv4 address 101.1.1.1 255.255.255.0
!
interface TenGigE0/6/0/0
 ipv4 address 12.1.1.1 255.255.255.0
!
mpls ldp
 router-id 1.1.1.1
 interface TenGigE0/6/0/0
 !
!
multicast-routing
 vrf vpn1 address-family ipv4
  mdt data 233.1.0.0/16 threshold 3
  mdt default ipv4 232.1.1.1
  rate-per-route
  interface all enable
  accounting per-prefix
 !
 address-family ipv4
  nsf
  mdt source Loopback0
  interface all enable
  accounting per-prefix
 !
!
router bgp 100
 bgp router-id 1.1.1.1
 address-family ipv4 unicast
 !
```

```
 address-family vpnv4 unicast
 !
 address-family ipv4 mdt
 !
 neighbor 9.9.9.9
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
  !
  address-family vpnv4 unicast
  !
  address-family ipv4 mdt
  !
 !
 vrf vpn1
  rd 1:1
  address-family ipv4 unicast
   redistribute ospf 1
  !
 !
!
router ospf 1
 vrf vpn1
  router-id 2.2.2.2
  redistribute bgp 100
  area 0
   interface Loopback1
   !
   interface GigabitEthernet0/5/0/0
   !
  !
 !
!
router ospf 100
 router-id 1.1.1.1
 area 0
  interface Loopback0
  !
  interface TenGigE0/6/0/0
  !
 !
!
router pim vrf vpn1 address-family ipv4
 rp-address 2.2.2.2
 log neighbor changes
!
router pim vrf default address-family ipv4
 rp-address 1.1.1.1
!
end
```

PE2:

```
!
vrf vpn1
 address-family ipv4 unicast
  import route-target
   1:1
  !
  export route-target
   1:1
  !
 !
!
interface Loopback0
 ipv4 address 9.9.9.9 255.255.255.255
!
interface Loopback1
 vrf vpn1
 ipv4 address 10.10.10.10 255.255.255.255
```

```
!
interface GigabitEthernet0/2/2/7
 vrf vpn1
 ipv4 address 122.1.1.1 255.255.255.0
 negotiation auto
!
interface TenGigE0/3/0/0
 ipv4 address 12.1.1.2 255.255.255.0
!
mpls ldp
 router-id 9.9.9.9
 interface TenGigE0/3/0/0
 !
!
multicast-routing
 vrf vpn1 address-family ipv4
  mdt data 233.1.0.0/16 threshold 3
  mdt default ipv4 232.1.1.1
  rate-per-route
  interface all enable
  accounting per-prefix
 !
 address-family ipv4
  nsf
  mdt source Loopback0
  interface all enable
  accounting per-prefix
 !
!
router bgp 100
 bgp router-id 9.9.9.9
 address-family ipv4 unicast
 !
 address-family vpnv4 unicast
 !
 address-family ipv4 mdt
 !
 neighbor 1.1.1.1
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
  !
  address-family vpnv4 unicast
  !
  address-family ipv4 mdt
  !
 !
 vrf vpn1
  rd 1:1
  address-family ipv4 unicast
   redistribute ospf 1
  !
 !
!
router ospf 1
 vrf vpn1
  router-id 10.10.10.10
  redistribute bgp 100
  area 0
   interface Loopback1
   !
   interface GigabitEthernet0/2/2/7
   !
  !
 !
!
router ospf 100
 router-id 9.9.9.9
 area 0
  interface Loopback0
  !
  interface TenGigE0/3/0/0
  !
```

```
 !
!
router pim vrf vpn1 address-family ipv4
 rp-address 2.2.2.2
!
router pim vrf default address-family ipv4
 rp-address 1.1.1.1
!
end
```

CE4:

For information about configuring the CE router, using Cisco IOS software, see the appropriate Cisco IOS software configuration documentation.

```
!
interface Loopback0
 ipv4 address 101.101.101.101 255.255.255.255
!
interface GigabitEthernet0/0/0/0
 ipv4 address 101.1.1.2 255.255.255.0
!
interface GigabitEthernet0/0/0/3
 ipv4 address 11.1.1.1 255.255.255.0
!
multicast-routing
 address-family ipv4
  interface all enable
 !
!
router ospf 1
 router-id 101.101.101.101
 area 0
  interface Loopback0
  !
  interface GigabitEthernet0/0/0/0
  !
  interface GigabitEthernet0/0/0/3
  !
 !
!
router pim vrf default address-family ipv4
 rp-address 2.2.2.2
 interface Loopback0
 !
 interface GigabitEthernet0/0/0/0
 !
 interface GigabitEthernet0/0/0/3
 !
!
end
```

CE3:

For information about configuring the CE router, using Cisco IOS software, see the appropriate Cisco IOS software configuration documentation.

```
interface Loopback0
 ipv4 address 122.122.122.122 255.255.255.255
!

interface GigabitEthernet0/1/3/0
 ipv4 address 22.1.1.1 255.255.255.0
!

interface GigabitEthernet0/2/3/0
 ipv4 address 122.1.1.2 255.255.255.0
```

```
multicast-routing
 address-family ipv4
  interface all enable
!
router ospf 1
 router-id 122.122.122.122
 area 0
  interface Loopback0
  !
  interface GigabitEthernet0/1/3/0
  !
  interface GigabitEthernet0/2/3/0
  !
 !
!
router pim vrf default address-family ipv4
 rp-address 2.2.2.2
 interface Loopback0
 !
 interface GigabitEthernet0/1/3/0
 !
 interface GigabitEthernet0/2/3/0
 !
!
end
```

Configuring MVPN to Advertise Routes Between the CE and the PE Using BGP: Example

PE1:

```
vrf vpn1
 address-family ipv4 unicast
  import route-target
   1:1
  !
  export route-target
   1:1
  !
 !
!
interface Loopback0
 ipv4 address 1.1.1.1 255.255.255.255
!
interface Loopback1
 vrf vpn1
 ipv4 address 2.2.2.2 255.255.255.255
!
interface GigabitEthernet0/5/0/0
 vrf vpn1
 ipv4 address 101.1.1.1 255.255.255.0
!
interface TenGigE0/6/0/0
 ipv4 address 12.1.1.1 255.255.255.0
!
mpls ldp
 router-id 1.1.1.1
 interface TenGigE0/6/0/0
 !
!
multicast-routing
 vrf vpn1 address-family ipv4
  mdt data 233.1.0.0/16 threshold 3
  mdt default ipv4 232.1.1.1
  rate-per-route
  interface all enable
  accounting per-prefix
 !
```

```
 address-family ipv4
  nsf
  mdt source Loopback0
  interface all enable
  accounting per-prefix
 !
!
!
route-policy pass-all
  pass
end-policy
!
router bgp 100
 bgp router-id 1.1.1.1
 address-family ipv4 unicast
 !
 address-family vpnv4 unicast
 !
 address-family ipv4 mdt
 !
 neighbor 9.9.9.9
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
  !
  address-family vpnv4 unicast
  !
  address-family ipv4 mdt
  !
 !
 vrf vpn1
  rd 1:1
  address-family ipv4 unicast
   redistribute connected
  !
  neighbor 101.1.1.2
   remote-as 400
   address-family ipv4 unicast
    route-policy pass-all in
    route-policy pass-all out
   !
  !
 !
!
router ospf 100
 router-id 1.1.1.1
 area 0
  interface Loopback0
  !
  interface TenGigE0/6/0/0
  !
 !
!
router pim vrf vpn1 address-family ipv4
 rp-address 2.2.2.2
 log neighbor changes
!
router pim vrf default address-family ipv4
 rp-address 1.1.1.1
!
end
```

PE2:

```
!
vrf vpn1
 address-family ipv4 unicast
  import route-target
   1:1
  !
```

```
  export route-target
   1:1
  !
 !
!
interface Loopback0
 ipv4 address 9.9.9.9 255.255.255.255
!
interface Loopback1
 vrf vpn1
 ipv4 address 10.10.10.10 255.255.255.255
!
interface GigabitEthernet0/2/2/7
 vrf vpn1
 ipv4 address 122.1.1.1 255.255.255.0
 negotiation auto
!
interface TenGigE0/3/0/0
 ipv4 address 12.1.1.2 255.255.255.0
!
mpls ldp
 router-id 9.9.9.9
 interface TenGigE0/3/0/0
 !
!
multicast-routing
 vrf vpn1 address-family ipv4
  mdt data 233.1.0.0/16 threshold 3
  mdt default ipv4 232.1.1.1
  rate-per-route
  interface all enable
  accounting per-prefix
 !
 address-family ipv4
  nsf
  mdt source Loopback0
  interface all enable
  accounting per-prefix
 !
!
!
route-policy pass-all
  pass
end-policy
!
router bgp 100
 bgp router-id 9.9.9.9
 address-family ipv4 unicast
 !
 address-family vpnv4 unicast
 !
 address-family ipv4 mdt
 !
 neighbor 1.1.1.1
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
  !
  address-family vpnv4 unicast
  !
  address-family ipv4 mdt
  !
 !
 vrf vpn1
  rd 1:1
  address-family ipv4 unicast
   redistribute connected
  !
  neighbor 122.1.1.2
   remote-as 500
   address-family ipv4 unicast
    route-policy pass-all in
    route-policy pass-all out
```

```
   !
  !
 !
!
router ospf 100
 router-id 9.9.9.9
 area 0
  interface Loopback0
  !
  interface TenGigE0/3/0/0
  !
 !
!
router pim vrf vpn1 address-family ipv4
 rp-address 2.2.2.2
!
router pim vrf default address-family ipv4
 rp-address 1.1.1.1
!
end
```

CE4:

For information about configuring the CE router, using Cisco IOS software, see the appropriate Cisco IOS software configuration documentation.

```
interface Loopback0
 ipv4 address 101.101.101.101 255.255.255.255
!
interface GigabitEthernet0/0/0/0
 ipv4 address 101.1.1.2 255.255.255.0
!
interface GigabitEthernet0/0/0/3
 ipv4 address 11.1.1.1 255.255.255.0
!
multicast-routing
 address-family ipv4
  interface all enable
 !
!
!
route-policy pass-all
  pass
end-policy
!
router bgp 400
 bgp router-id 101.101.101.101
 address-family ipv4 unicast
  redistribute connected
 !
 neighbor 101.1.1.1
  remote-as 100
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
!
router pim vrf default address-family ipv4
 rp-address 2.2.2.2
 interface Loopback0
 !
 interface GigabitEthernet0/0/0/0
 !
 interface GigabitEthernet0/0/0/3
 !
!
end
```

CE3:

For information about configuring the CE router, using Cisco IOS software, see the appropriate Cisco IOS software configuration documentation.

```
interface Loopback0
 ipv4 address 122.122.122.122 255.255.255.255
!

interface GigabitEthernet0/1/3/0
 ipv4 address 22.1.1.1 255.255.255.0
!

interface GigabitEthernet0/2/3/0
 ipv4 address 122.1.1.2 255.255.255.0

multicast-routing
 address-family ipv4
  interface all enable
!
!
!
route-policy pass-all
  pass
end-policy
!
router bgp 500
 bgp router-id 122.122.122.122
 address-family ipv4 unicast
  redistribute connected
 !
 neighbor 122.1.1.1
  remote-as 100
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
!
!
router pim vrf default address-family ipv4
 rp-address 2.2.2.2
 interface Loopback0
 !
 interface GigabitEthernet0/1/3/0
 !
 interface GigabitEthernet0/2/3/0
 !
!
end
```

Configuration Examples for MVPN Profiles

The profile-wise configuration examples for the various MVPN profiles.

Configuration Examples for Inband mLDP profiles

Profile-6: VRF Inband mLDP

```
router bgp 100
 mvpn
!
multicast-routing
 mdt source Loopback0
 vrf v61
```

```
  address-family ipv4
    mdt mtu 1600
    mdt mldp in-band-signaling ipv4
    interface all enable
   !
  address-family ipv6
    mdt mtu 1600
    mdt mldp in-band-signaling ipv4
   interface all enable
 !
!
router pim
 vrf v61
  address-family ipv4
   rpf topology route-policy mldp-inband
  !
  address-family ipv6
   rpf topology route-policy mldp-inband
  !
!
route-policy mldp-inband
  set core-tree mldp-inband
end-policy
!
```

Profile-7: Global Inband mLDP

```
multicast-routing
 address-family ipv4
   mdt source Loopback0
   mdt mldp in-band-signaling ipv4
   ssm range Global-SSM-Group
  interface all enable
!
 address-family ipv6
  mdt source Loopback0
  mdt mldp in-band-signaling ipv4
  ssm range Global-SSM-Group-V6
  interface all enable
!
 router pim
  address-family ipv4
   rpf topology route-policy mldp-inband
  !
  address-family ipv6
   rpf topology route-policy mldp-inband
 !
!
route-policy mldp-inband
  set core-tree mldp-inband
end-policy
!
```

Configuration Examples for P2MP-TE profiles

Profile-8: Global Static P2MP-TE

```
interface Loopback0
 ipv4 address 200.200.1.1 255.255.255.255
!
multicast-routing
 address-family ipv4
  mdt source Loopback0
   ssm range Global-SSM-Group
  interface all enable
!
 address-family ipv6
  mdt source Loopback0
  ssm range Global-SSM-Group-V6
  interface all enable
!
router igmp
```

```
 interface tunnel-mte1
   static-group 228.1.1.1 2.2.2.1
 !
router mld
 interface tunnel-mte1
  static-group ff3e:0:228::1 2001:2:2:2::1
```

Profile-10: VRF static P2MP-TE with BGP-AD

```
!
multicast-routing
 mdt source Loopback0
 vrf v101
  address-family ipv4
   mdt static p2mp-te tunnel-mte10
    interface all enable
   bgp auto-discovery pim
  !
!
router igmp
 vrf v101
  interface tunnel-mte10
   static-group 227.101.1.1 101.7.1.1
 !
!
```

Profile-18: Rosen Static P2MP-TE with BGP-AD and PIM signaling

```
multicast-routing
 mdt source Loopback0
 vrf v181
  address-family ipv4
    mdt default p2mp-te static tunnel-mte18
   interface all enable
   bgp auto-discovery pim
   !
  address-family ipv6
   mdt default p2mp-te static tunnel-mte181
    interface all enable
   bgp auto-discovery pim
 !
!
router pim
 vrf v181
  address-family ipv4
   rpf topology route-policy p2mp-te-default
    !
  address-family ipv6
   rpf topology route-policy p2mp-te-default
  !
!
route-policy p2mp-te-default
  set core-tree p2mp-te-default
end-policy
!
```

Profile-16: Rosen Static P2MP-TE with BGP-Ad and BGP signaling

```
!
multicast-routing
 mdt source Loopback0
 vrf v161
  address-family ipv4
   mdt default p2mp-te static tunnel-mte16
    interface all enable
   bgp auto-discovery pim
   !
  address-family ipv6
   mdt default p2mp-te static tunnel-mte161
    interface all enable
   bgp auto-discovery pim
 !
!
```

```
router pim
 vrf v161
  address-family ipv4
   rpf topology route-policy p2mp-te-default
   mdt c-multicast-routing bgp
   !
  address-family ipv6
   rpf topology route-policy p2mp-te-default
   mdt c-multicast-routing bgp
 !
!
route-policy p2mp-te-default
  set core-tree p2mp-te-default
end-policy
!
```

Profile-20: Default MDT - P2MP-TE - BGP-AD - PIM

```
multicast-routing
 mdt source Loopback0
vrf p20_vrf1
 address-family ipv4 unicast
  import route-target
   20:1
  !
  export route-target
   20:1
  !
route-policy rpf-for-p20_vrf1
  set core-tree p2mp-te-default
end-policy
!
vrf p20_vrf1
  rd 20:1
  address-family ipv4 unicast
   redistribute connected
  !
  address-family ipv4 mvpn
  !
 !
rsvp
 interface Bundle-Ether1
  bandwidth
 !
 !
 vrf p20_vrf1
  address-family ipv4
   mdt source Loopback0
   rate-per-route
   interface all enable
   accounting per-prefix
   bgp auto-discovery p2mp-te
   !
   mdt default p2mp-te
  !
 !
vrf p20_vrf1
  address-family ipv4
   rpf topology route-policy rpf-for-p20_vrf1
   rp-address 100.1.20.1 static_p20vrf1_v4
   log neighbor changes
   interface GigabitEthernet0/0/0/0.202
    enable
   !
   interface GigabitEthernet0/0/0/0.203
    enable
   !
   interface GigabitEthernet0/0/0/1.202
    enable
   !
   interface GigabitEthernet0/0/0/1.203
    enable
   !
```

Configuration examples for Partitioned mLDP profiles

Profile-2: Partitioned mLDP MP2MP without BGP-AD

```
router bgp 100
  mvpn
!
multicast-routing
 vrf v21
  address-family ipv4
    mdt mtu 1600
    mdt partitioned mldp ipv4 mp2mp
    interface all enable
   !
  address-family ipv6
    mdt mtu 1600
    mdt partitioned mldp ipv4 mp2mp
    interface all enable
 !
!
router pim
 vrf v21
  address-family ipv4
   rpf topology route-policy mldp-partitioned-mp2mp
  !
  address-family ipv6
   rpf topology route-policy mldp-partitioned-mp2mp
 !
!
route-policy mldp-partitioned-mp2mp
  set core-tree mldp-partitioned-mp2mp
end-policy
!
```

Profile-4: Partitioned mLDP MP2MP with BGP-AD and PIM signaling

```
!
multicast-routing
 mdt source Loopback0
 vrf v41
  address-family ipv4
    mdt mtu 1600
mdt partitioned mldp ipv4 mp2mp
mdt data 255
   interface all enable
   bgp auto-discovery mldp
   !
  address-family ipv6
    mdt mtu 1600
mdt partitioned mldp ipv4 mp2mp
mdt data 255
    interface all enable
   bgp auto-discovery mldp
   !
!
router pim
 vrf v41
  address-family ipv4
   rpf topology route-policy mldp-partitioned-mp2mp
  !
  address-family ipv6
   rpf topology route-policy mldp-partitioned-mp2mp
 !
!
route-policy mldp-partitioned-mp2mp
  set core-tree mldp-partitioned-mp2mp
end-policy
!
```

Profile-15: Partitioned mLDP MP2MP with BGP-AD and BGP signaling

```
!
multicast-routing
 mdt source Loopback0
 vrf v151
  address-family ipv4
    mdt mtu 1600
mdt partitioned mldp ipv4 mp2mp
mdt data 255
    interface all enable
   bgp auto-discovery mldp
   !
  address-family ipv6
    mdt mtu 1600
mdt partitioned mldp ipv4 mp2mp
mdt data 255
    interface all enable
   bgp auto-discovery mldp
 !
!
router pim
 vrf v151
  address-family ipv4
   rpf topology route-policy mldp-partitioned-mp2mp
   mdt c-multicast-routing bgp
  !
  address-family ipv6
   rpf topology route-policy mldp-partitioned-mp2mp
   mdt c-multicast-routing bgp
  !
!
route-policy mldp-partitioned-mp2mp
  set core-tree mldp-partitioned-mp2mp
end-policy
!
```

Profile-5: Partitioned mLDP P2MP with BGP-AD and PIM signaling

```
!
multicast-routing
 mdt source Loopback0
vrf v51
  address-family ipv4
    mdt mtu 1600
mdt partitioned mldp ipv4 p2mp
mdt data 255
   interface all enable
   bgp auto-discovery mldp
  !
  address-family ipv6
    mdt mtu 1600
mdt partitioned mldp ipv4 p2mp
mdt data 255
    interface all enable
   bgp auto-discovery mldp
  !
!
router pim
 vrf v51
  address-family ipv4
   rpf topology route-policy mldp-partitioned-p2mp
  !
  address-family ipv6
   rpf topology route-policy mldp-partitioned-p2mp
 !
!
route-policy mldp-partitioned-p2mp
  set core-tree mldp-partitioned-p2mp
end-policy
```

Profile-14: Partitioned mLDP P2MP with BGP-AD and BGP siganling

```
!
multicast-routing
```

```
 mdt source Loopback0
 vrf v141
  address-family ipv4
    mdt mtu 1600
    mdt partitioned mldp ipv4 p2mp
    mdt data 255
   interface all enable
   bgp auto-discovery mldp
  !
  address-family ipv6
    mdt mtu 1600
mdt partitioned mldp ipv4 p2mp
mdt data 255
    interface all enable
   bgp auto-discovery mldp
 !
!
router pim
 vrf v141
  address-family ipv4
   rpf topology route-policy mldp-partitioned-p2mp
   mdt c-multicast-routing bgp
   !
  address-family ipv6
   rpf topology route-policy mldp-partitioned-p2mp
   mdt c-multicast-routing bgp
 !
!
route-policy mldp-partitioned-p2mp
  set core-tree mldp-partitioned-p2mp
end-policy
!
```

Configuration Examples for Rosen-mGRE profiles

Profile-0: Rosen mGRE with MDT SAFI

```
router bgp 100
 address-family ipv4 mdt
 !
 neighbor X.X.X.X   < -----RR or Remote PE ip address
 address-family ipv4 mdt
 !
!

multicast-routing
 address-family ipv4
  mdt source Loopback0
  interface all enable
!
 address-family ipv6
   mdt source Loopback0
  interface all enable
!
 vrf v1
  address-family ipv4
   mdt mtu 1600
   mdt data 231.1.1.2/32
   mdt default ipv4 231.1.1.1
   interface all enable
  !
  address-family ipv6
   mdt mtu 1600
   mdt data 231.1.1.2/32
   mdt default ipv4 231.1.1.1
   interface all enable
 !
!
```

Profile-3: Rosen mGRE with BGP-AD and PIM signaling

```
router bgp 100
 !
 address-family ipv4 mvpn
 !
 address-family ipv6 mvpn
 !
 neighbor X.X.X.X   < -----RR or Remote PE ip address
  address-family ipv4 mvpn
  !
  address-family ipv6 mvpn
  !
 !
 vrf v31
  rd 100:31
  address-family ipv4 mvpn
  !
  address-family ipv6 mvpn
  !
 !
!
multicast-routing
 mdt source Loopback0
 vrf v31
  address-family ipv4
   mdt mtu 1600
   mdt data 232.31.1.2/32
   mdt default ipv4 232.31.1.1
   interface all enable
   bgp auto-discovery pim
  !
  address-family ipv6
   mdt mtu 1600
   mdt data 232.31.1.2/32
   mdt default ipv4 232.31.1.1
   interface all enable
   bgp auto-discovery pim
 !
!
```

Profile-11: Rosen mGRE with BGP-AD and BGP signaling

```
router bgp 100
 !
 address-family ipv4 mvpn
 !
 address-family ipv6 mvpn
 !
 neighbor X.X.X.X   < -----RR or Remote PE ip address
  address-family ipv4 mvpn
  !
  address-family ipv6 mvpn
  !
 !
 vrf v111
  rd 100:111
  address-family ipv4 mvpn
  !
  address-family ipv6 mvpn
  !
 !
!
multicast-routing
 mdt source Loopback0
 vrf v111
  address-family ipv4
   mdt mtu 1600
   mdt data 232.111.1.2/32
   mdt default ipv4 232.111.1.1
   interface all enable
   bgp auto-discovery pim
  !
  address-family ipv6
```

```
   mdt mtu 1600
   mdt data 232.111.1.2/32
   mdt default ipv4 232.111.1.1
   interface all enable
   bgp auto-discovery pim
 !
!
router pim
 vrf v111
  address-family ipv4
   mdt c-multicast-routing bgp
  !
  address-family ipv6
   mdt c-multicast-routing bgp
 !
```

Configuration Examples for Rosen mLDP profiles

Profile-1: Rosen mLDP with M2MP without BGP-AD

```
vrf v11
 vpn id 100:11
!
router bgp 100
  mvpn
!
multicast-routing
 mdt source Loopback0
 vrf v11
  address-family ipv4
    mdt mtu 1600
    mdt default mldp ipv4 100.100.1.1
   mdt default mldp ipv4 100.100.1.2
   mdt data 255
   interface all enable
  !
  address-family ipv6
    mdt mtu 1600
    mdt default mldp ipv4 100.100.1.1
   mdt default mldp ipv4 100.100.1.2
   mdt data 255
   interface all enable
 !
!
router pim
 vrf v11
  address-family ipv4
   rpf topology route-policy rosen-mldp
  !
  address-family ipv6
   rpf topology route-policy rosen-mldp
 !
!
route-policy rosen-mldp
  set core-tree mldp-default
end-policy!
```

Profile-9: Rosen mLDP MP2MP with BGP-AD and PIM signaling

```
vrf v91
 vpn id 100:91

!
!
multicast-routing
 mdt source Loopback0
 vrf v91
  address-family ipv4
    mdt mtu 1600
    mdt default mldp ipv4 100.100.1.1
   mdt default mldp ipv4 100.100.1.2
```

```
   mdt data 255
   interface all enable
   bgp auto-discovery mldp
   !
  address-family ipv6
    mdt mtu 1600
   mdt default mldp ipv4 100.100.1.1
   mdt default mldp ipv4 100.100.1.2
   mdt data 255
   interface all enable
   bgp auto-discovery mldp
  !
!
router pim
 vrf v91
  address-family ipv4
   rpf topology route-policy rosen-mldp
  !
  address-family ipv6
   rpf topology route-policy rosen-mldp
  !
 !
route-policy rosen-mldp
  set core-tree mldp-default
end-policy
```

Profile-13: Rosen mLDP MP2MP with BGP-AD and BGP signaling

```
vrf v131
 vpn id 100:131
 !

!
multicast-routing
 mdt source Loopback0
 vrf v131
  address-family ipv4
    mdt mtu 1600
    mdt default mldp ipv4 100.100.1.1
   mdt default mldp ipv4 100.100.1.2
   mdt data 255
   interface all enable
   bgp auto-discovery mldp
   !
  address-family ipv6
    mdt mtu 1600
    mdt default mldp ipv4 100.100.1.1
   mdt default mldp ipv4 100.100.1.2
   mdt data 255
   interface all enable
   bgp auto-discovery mldp
  !
!
router pim
 vrf v131
  address-family ipv4
   rpf topology route-policy rosen-mldp
   mdt c-multicast-routing bgp
  !
  address-family ipv6
   rpf topology route-policy rosen-mldp
   mdt c-multicast-routing bgp
  !
!
route-policy rosen-mldp
  set core-tree mldp-default
end-policy
```

Profile-17: Rosen mLDP P2MP with BGP-AD and PIM signaling

```
!
multicast-routing
 mdt source Loopback0
```

```
 vrf v171
  address-family ipv4
    mdt mtu 1600
mdt default mldp p2mp
mdt data 255
    interface all enable
   bgp auto-discovery mldp
   !
  address-family ipv6
    mdt mtu 1600
mdt default mldp p2mp
mdt data 255
    interface all enable
   bgp auto-discovery mldp
 !
!
router pim
 vrf v171
  address-family ipv4
   rpf topology route-policy rosen-mldp
   !
  address-family ipv6
   rpf topology route-policy rosen-mldp
 !
!
route-policy rosen-mldp
  set core-tree mldp-default
end-policy
!
```

Profile-12: Rosen mLDP P2MP with BGP-AD and BGP signaling

```
!
multicast-routing
 mdt source Loopback0
 vrf v121
  address-family ipv4
    mdt mtu 1600
mdt default mldp p2mp
mdt data 255
    interface all enable
   bgp auto-discovery mldp
   !
  address-family ipv6
    mdt mtu 1600
mdt default mldp p2mp
mdt data 255
    interface all enable
   bgp auto-discovery mldp
 !
!
router pim
 vrf v121
  address-family ipv4
   rpf topology route-policy rosen-mldp
  mdt c-multicast-routing bgp
  !
  address-family ipv6
   rpf topology route-policy rosen-mldp
  mdt c-multicast-routing bgp
 !
!
route-policy rosen-mldp
  set core-tree mldp-default
end-policy
!
```

Configuration Examples for multicast support on PW-HE

Enabling multicast

```
configure
multicast-routing
address-family ipv4
interface pw-ether1
enable
vrf v1
address-family ipv4
interface pw-ether2
enable
 !
!
```

Configuring Static Join (with IGMP)

```
configure
router igmp
interface pw-ether1
static-group 225.0.0.1
static-group 225.0.0.2 10.1.1.1
 !
!
```

Configuring Static Join (with MLD)

```
configure
router mld
interface pw-ether2
static-group ff15::e100
static-group ff15::e100 2000:10::1
  !
!
```

Configuring MVPN Extranet Routing: Example

These examples describe two ways to configure MVPN extranet routing:

For the full set of configuration tasks, see Configuring MVPN Extranet Routing, on page 153.

Configuring the Source MVRF on the Receiver PE Router: Example

The following examples show how to configure MVPN extranet routing by specifying the source MVRF on the receiver PE router.

You must configure both the source PE router and the receiver PE router.

Configure the Source PE Router Using Route Targets

```
interface Loopback5
 ipv4 address 201.5.5.201 255.255.255.255
!
interface Loopback22
 vrf provider-vrf
 ipv4 address 201.22.22.201 255.255.255.255
!
interface GigabitEthernet0/6/0/0
 vrf provider-vrf
 ipv4 address 10.10.10.1 255.255.0.0
```

```
!
vrf provider-vrf
 address-family ipv4 unicast
 import route-target
 1100:1
 !
export route-target
 1100:1
 !
!
router bgp 1
 regular BGP MVPN config
vrf provider-vrf
 rd 1100:1
 address-family ipv4 unicast
 redistribute connected

 !
!
multicast-routing
 vrf provider-vrf address-family ipv4
 mdt data 226.1.4.0/24 threshold 3
 log-traps
 mdt default ipv4 226.0.0.4
 rate-per-route
 interface all enable
 accounting per-prefix
 !
!
address-family ipv4
 nsf
 mdt source Loopback5
 interface all enable
 !
!
router pim vrf provider-vrf address-family ipv4
 rp-address 201.22.22.201
!
```

Configure the Receiver PE Router Using Route Targets

```
interface Loopback5
 ipv4 address 202.5.5.202 255.255.255.255
!
interface GigabitEthernet0/3/0/2
 vrf receiver-vrf
 ipv4 address 20.20.20.1 255.255.0.0
!
vrf provider-vrf
 address-family ipv4 unicast
 import route-target
 1100:1
 !
 export route-target
 1100:1
 !
!
vrf receiver-vrf
 address-family ipv4 unicast
 import route-target
 1100:1
 1101:1
 !
 export route-target
 1101:1
 !
!
multicast-routing
 vrf provider-vrf address-family ipv4
 log-traps
```

```
 mdt default ipv4 226.0.0.4
 rate-per-route
 interface all enable
 accounting per-prefix
!

 vrf receiver_vrf address-family ipv4
 log-traps
 mdt default ipv4 226.0.0.5
 rate-per-route
 interface all enable
 accounting per-prefix
!
address-family ipv4
 nsf
 mdt source Loopback5
 interface all enable
!
router pim vrf provider-vrf address-family ipv4
 rp-address 201.22.22.201
!

router pim vrf receiver_vrf address-family ipv4
 rp-address 201.22.22.201
!
router bgp 1
 regular BGP MVPN config
vrf provider-vrf
 rd 1100:1
 address-family ipv4 unicast
 redistribute connected
 !

 vrf receiver_vrf
 rd 1101:1
 address-family ipv4 unicast
 redistribute connected
!
```

Configuring RPL Policies in Receiver VRFs to Propagate Joins to a Source VRF: Example

In addition to configuring route targets, Routing Policy Language (RPL) policies can be configured in receiver VRFs on receiver PE routers to propagate joins to a specified source VRF. However, this configuration is optional.

The following configuration example shows a policy where the receiver VRF can pick either “provider_vrf_1” or “provider_vrf_2” to propogate PIM joins.

In this example, provider_vrf_1 is used for multicast streams in the range of from 227.0.0.0 to 227.255.255.255, while provider_vrf_2 is being used for streams in the range of from 228.0.0.0 to 228.255.255.255.

```
route-policy extranet_streams_from_provider_vrf
 if destination in (227.0.0.0/32 ge 8 le 32) then
  set rpf-topology vrf provider_vrf_1
 elseif destination in (228.0.0.0/32 ge 8 le 32) then
  set rpf-topology vrf provider_vrf_2
 else
  pass
 endif
end-policy
!
router pim vrf receiver_vrf address-family ipv4
 rpf topology route-policy extranet_streams_from_provider_vrf
!
```

Configuring the Receiver MVRF on the Source PE Router: Example

The following examples show how to configure MVPN extranet routing by specifying the receiver MVRF on the source PE router.

Note You must configure both the source PE router and the receiver PE router.

Configure the Source PE Router Using Route Targets

```
interface Loopback5
 ipv4 address 202.5.5.202 255.255.255.255
!
interface GigabitEthernet0/3/0/2
 vrf provider-vrf
 ipv4 address 20.20.20.1 255.255.0.0
!
vrf provider-vrf
 address-family ipv4 unicast
 import route-target
 1100:1
 !
 export route-target
 1100:1
 !
!

vrf receiver-vrf
 address-family ipv4 unicast
 import route-target
 1100:1
 1101:1
 !
 export route-target
 1101:1
 !
!

router bgp 1
 regular BGP MVPN config
vrf provider-vrf
 rd 1100:1
 address-family ipv4 unicast
 redistribute connected
 !

 vrf receiver-vrf
 rd 1101:1
 address-family ipv4 unicast
 redistribute connected
 !
!

multicast-routing
 vrf provider-vrf address-family ipv4
 log-traps
 mdt default ipv4 226.0.0.4
 rate-per-route
 interface all enable
 accounting per-prefix
!

 vrf receiver_vrf address-family ipv4
 log-traps
 mdt default ipv4 226.0.0.5
```

```
 rate-per-route
 interface all enable
 accounting per-prefix
!
address-family ipv4
 nsf
 mdt source Loopback5
 interface all enable
!
router pim vrf provider-vrf address-family ipv4
 rp-address 201.22.22.201
!
router pim vrf receiver_vrf address-family ipv4
 rp-address 201.22.22.201
!
```

Configure the Receiver PE Router Using Route Targets

```
interface Loopback5
 ipv4 address 201.5.5.201 255.255.255.255
!
interface Loopback22
 vrf receiver_vrf
 ipv4 address 201.22.22.201 255.255.255.255
!
interface GigabitEthernet0/6/0/0
 vrf receiver_vrf
 ipv4 address 10.10.10.1 255.255.0.0
!

vrf receiver_vrf
 address-family ipv4 unicast
 import route-target
 1100:1
 1101:1
 !
 export route-target
 1101:1
 !
!

router bgp 1
 regular BGP MVPN config
vrf receiver_vrf
 rd 1101:1
 address-family ipv4 unicast
 redistribute connected
 !

multicast-routing
 vrf receiver_vrf address-family ipv4
 log-traps
 mdt default ipv4 226.0.0.5
 rate-per-route
 interface all enable
 accounting per-prefix
 !
address-family ipv4
 nsf
 mdt source Loopback5
 interface all enable
!

router pim vrf receiver_vrf address-family ipv4
 rp-address 201.22.22.201
!
```

Configuring RPL Policies in Receiver VRFs on Source PE Routers to Propagate Joins to a Source VRF: Example

In addition to configuring route targets , RPL policies can be configured in receiver VRFs on a ource PE router to propagate joins to a specified source VRF. However, this configuration is optional.

The configuration below shows a policy in which the receiver VRF can select either “provider_vrf_1” or “provider_vrf_2” to propagate PIM joins. Provider_vrf_1 will be selected if the rendezvous point (RP) for a multicast stream is 201.22.22.201, while provider_vrf_2 will be selected if the RP for a multicast stream is 202.22.22.201.

As an alternative, you can configure a multicast group-based policy as shown in the Configuring RPL Policies in Receiver VRFs to Propagate Joins to a Source VRF: Example, on page 206.

```
route-policy extranet_streams_from_provider_rp
 if source in (201.22.22.201) then
  set rpf-topology vrf provider_vrf_1
 else if source in (202.22.22.201) then
  set rpf-topology vrf provider_vrf_2
 else
  pass
 endif
end-policy
!
router pim vrf receiver_vrf address-family ipv4
 rpf topology route-policy extranet_streams_from_provider_rp
 rp-address 201.22.22.201 grange_227
 rp-address 202.22.22.201 grange_228
!
```

Configuring Multicast Hub and Spoke Topology: Example

These examples describe two ways to configure Multicast Hub and Spoke:

Figure 12: Example for CE1 PE1PE3 CE3Multicast Hub and Spoke Topology

CE1---------- PE1 ---------- PE3 ---------- CE3

CE1, PE1, and PE3 are all on Cisco IOS XR Software, CE3 has Cisco IOS Software in order to configure autorp on VRF interface. For information about configuring the CE router, using Cisco IOS software, see the appropriate Cisco IOS software documentation.

Hub and Spoke Non-Turnaround Configuration: Example

A1-Hub-1 (bsr RP) A1-Hub-4 (auto-rp RP)

A1-Spoke-3

No turnaround case with bsr and autorp relay

PE1:

```
vrf A1-Hub-1
address-family ipv4 unicast
import route-target

   1000:10
```

```
   1001:10
  !
  export route-target
   1000:10
  !
 !
vrf A1-Hub-Tunnel
address-family ipv4 unicast
  import route-target
   1000:10
  !
 !
!
vrf A1-Spoke-Tunnel
address-family ipv4 unicast
  import route-target
   1001:10
  !
 !
!
router pim
 vrf A1-Hub-1
  address-family ipv4
   rpf topology route-policy A1-Hub-Policy
   bsr relay vrf A1-Hub-Tunnel
   bsr candidate-bsr 201.10.10.201 hash-mask-len 30 priority 4
   bsr candidate-rp 201.10.10.201 group-list A1_PE1_RP_grange priority 4 interval 60
   auto-rp relay vrf A1-Hub-Tunnel
  !
 !
!
router pim
 vrf A1-Hub-Tunnel
  address-family ipv4
  !
 !
!
multicast-routing
```

```
 vrf A1-Hub-1
  address-family ipv4
   log-traps
   multipath
   rate-per-route
   interface all enable
   accounting per-prefix
  !
 !
!
multicast-routing
 vrf A1-Hub-Tunnel
  address-family ipv4
   mdt data 226.202.1.0/24 threshold 10
   log-traps
   mdt default ipv4 226.202.0.0
   rate-per-route
   accounting per-prefix
  !
 !
!
multicast-routing
 vrf A1-Spoke-Tunnel
  address-family ipv4
   mdt mtu 2000
   mdt data 226.202.2.0/24 threshold 5
   log-traps
   mdt default ipv4 226.202.0.1
   rate-per-route
   accounting per-prefix
  !
 !
!
router bgp 1
 vrf A1-Hub-1
  rd 1000:1
  address-family ipv4 unicast
```

```
   route-target download
   redistribute connected
   redistribute eigrp 20 match internal external metric 1000
  !
 !
!
router bgp 1
 vrf A1-Hub-Tunnel
  rd 1002:1
  address-family ipv4 unicast
   redistribute connected
  !
 !
!
router bgp 1
 vrf A1-Spoke-Tunnel
  rd 1002:2
  address-family ipv4 unicast
   redistribute connected
  !
 !
!
route-policy A1-Hub-Policy
  if extcommunity rt matches-any (1000:10) then
    set rpf-topology vrf A1-Hub-Tunnel
  elseif extcommunity rt matches-any (1001:10) then
    set rpf-topology vrf A1-Spoke-Tunnel
  else
    pass
  endif
end-policy
!
route-policy A1-Spoke-Policy
  if extcommunity rt matches-any (1000:10) then
    set rpf-topology vrf A1-Hub-Tunnel
  else
    pass
```

```
  endif
end-policy

!
```

PE3:

```
vrf A1-Hub-4
address-family ipv4 unicast
import route-target

   1000:10

   1001:10

  !

  export route-target

   1000:10

  !

 !

!

vrf A1-Spoke-2
address-family ipv4 unicast
import route-target

   1000:10
  !

  export route-target

   1001:10

  !

 !

!

vrf A1-Hub-Tunnel
address-family ipv4 unicast
import route-target

   1000:10

  !

 !

!

vrf A1-Spoke-Tunnel
address-family ipv4 unicast
import route-target

   1001:10

  !

 !

!

router pim
```

```
 vrf A1-Hub-4
  address-family ipv4
   rpf topology route-policy A1-Hub-Policy
   bsr relay vrf A1-Hub-Tunnel listen
   auto-rp relay vrf A1-Hub-Tunnel
  !
 !
!
router pim
 vrf A1-Spoke-2
  address-family ipv4
   rpf topology route-policy A1-Spoke-Policy
   bsr relay vrf A1-Hub-Tunnel listen
   auto-rp relay vrf A1-Hub-4
  !
 !
!
multicast-routing
 vrf A1-Hub-4
  address-family ipv4
   log-traps
   rate-per-route
   interface all enable
   accounting per-prefix
  !
 !
!
multicast-routing
 vrf A1-Spoke-2
  address-family ipv4
   log-traps
   rate-per-route
   interface all enable
   accounting per-prefix
  !
 !
!
```

```
multicast-routing
 vrf A1-Hub-Tunnel
  address-family ipv4
   mdt data 226.202.1.0/24 threshold 10
   log-traps
   mdt default ipv4 226.202.0.0
   rate-per-route
   accounting per-prefix
  !
 !
!
multicast-routing
 vrf A1-Spoke-Tunnel
  address-family ipv4
   mdt data 226.202.2.0/24 threshold 5
   log-traps
   mdt default ipv4 226.202.0.1
   rate-per-route
   accounting per-prefix
  !
 !
!
router bgp 1
 vrf A1-Hub-4
  rd 1000:4
  address-family ipv4 unicast
   route-target download
   redistribute connected
   redistribute eigrp 4 match internal external metric 1000
  !
 !
!
router bgp 1
 vrf A1-Spoke-2
  rd 1001:2
  address-family ipv4 unicast
   route-target download
   redistribute connected
```

```
   redistribute eigrp 6 match internal external metric 1000
  !
 !
router bgp 1
 vrf A1-Hub-Tunnel
  rd 1002:1
  address-family ipv4 unicast
   redistribute connected
  !
 !
!
router bgp 1
 vrf A1-Spoke-Tunnel
  rd 1002:2
  address-family ipv4 unicast
   redistribute connected
  !
 !
!
route-policy A1-Hub-Policy
  if extcommunity rt matches-any (1000:10) then
    set rpf-topology vrf A1-Hub-Tunnel
  elseif extcommunity rt matches-any (1001:10) then
    set rpf-topology vrf A1-Spoke-Tunnel
  else
    pass
  endif
end-policy
!
route-policy A1-Spoke-Policy
  if extcommunity rt matches-any (1000:10) then
    set rpf-topology vrf A1-Hub-Tunnel
  else
    pass
  endif
end-policy
!
```

CE1:

```
vrf A1-Hub-1
 address-family ipv4 unicast
  import route-target
   1000:10
   1001:10
  !
  export route-target
   1000:10
  !
 !
!
multicast-routing
 vrf A1-Hub-1
  address-family ipv4
   log-traps
   rate-per-route
   interface all enable
   accounting per-prefix
  !
 !
!
No router pim configuration required
```

CE3: Where autorp is configured (this is an Cisco IOS Software example, because auto-rp on vrf interface is not supported in Cisco IOS XR Software)

```
ip vrf A1-Hub-4
 rd 1000:4
 route-target export 1000:10
 route-target import 1000:10
 route-target import 1001:10
!
ip vrf A1-Spoke-2
 rd 1001:2
 route-target export 1001:10
 route-target import 1000:10
!
```

```
ip multicast-routing vrf A1-Hub-4
ip multicast-routing vrf A1-Spoke-2

interface Loopback10
 ip vrf forwarding A1-Hub-4
 ip address 103.10.10.103 255.255.255.255
 ip pim sparse-mode
!
ip pim vrf A1-Hub-4 autorp listener
ip pim vrf A1-Hub-4 send-rp-announce Loopback10 scope 32
ip pim vrf A1-Hub-4 send-rp-discovery Loopback10 scope 32
```

Hub and Spoke with Turnaround: Example

Multicast turnaround mandates a 2-interface connection to the hub site

To configure a CE as a turnaround router, it is connected to its respective PE through two interfaces and each interface is placed in a separate hub site vrf called **hub-x-in vrf** and **hub-x-out vrf**. Hub-x-in vrf carries joins that come from the receiver spoke site through the Hub Tunnel and hub-x-out vrf will carry the same joins towards the source spoke site through the Spoke Tunnel without violating the four basic rules below. The source spoke sends traffic to the spoke tunnel to hub-x-out which is turned around to the hub-tunnel on the hub-x-in interface.

1. Hub sites sends traffic only to MDTHub.
2. Spoke sites sends traffic only to MDTspoke.
3. Hub sites receives traffic from both tunnels.
4. Spoke sites receives traffic only from MDTHub.

A2-Spoke-1 A2-Hub-2

A2-Spoke-2 A2-Hub-3in

A2-Hub-2out

A2-Spoke-3 (spoke has auto-rp)

Figure 13: Example for CE1PE1PE2 CE2Multicast Hub and Spoke Topology with Turnaround

CE1---------- PE1 ---------- PE2 ---------- CE2

Routes exported by hub sites are imported by hub sites and spoke sites. Routes exported by spoke sites are imported by both **hub-x-out** and **hub-x-in** and hub site exports spoke routes back into the core by hub VRF route targets. This causes routes originated from one spoke site to be learned by all other spoke sites but with the nexthop of **hub-x-out**. For example, Spoke2 will see the RPF for Spoke1 reachable with nexthop of **A2-Hub-3in**. This is the fundamental difference in leaking of routes which helps in achieving turnaround of multicast traffic.

PE1:

```
vrf A2-Spoke-1
 address-family ipv4 unicast
  import route-target
   4000:1
   4000:2
   4000:3
   4000:4
  !
  export route-target
   4001:1
  !
 !
!
```

vrf A2-Spoke-2

```
address-family ipv4 unicast
  import route-target
   4000:1
   4000:2
   4000:3
   4000:4
  !
  export route-target
   4001:2
  !
 !
!
```

PE2:

```
vrf A2-Hub-2
 address-family ipv4 unicast
  import route-target
   4000:1
   4000:2
   4000:3
```

```
   4000:4
   4001:1
   4001:2
   4001:3
   4001:4
  !
  export route-target
   4000:2
  !
 !
!


vrf A2-Hub-3out
 address-family ipv4 unicast
  import route-target
   4000:1
   4000:2
   4000:3
   4000:4
   4001:1   --------à exports the spoke routes into CE2 into vrf default
   4001:2   --------à exports the spoke routes into CE2 into vrf default
   4001:3   --------à exports the spoke routes into CE2 into vrf default
   4001:4   --------à exports the spoke routes into CE2 into vrf default
  !
  export route-target
   4000:4
  !
 !
!
vrf A2-Hub-3in
 address-family ipv4 unicast
  import route-target
   4000:1
   4000:2
   4000:3
   4000:4
  !
```

```
  export route-target
   4000:3--------à selected spoke routes (in the prefix-set below) can be re-exported with
 hub route target so other spokes can reach them via A2-Hub-3in
  !
 !
!
prefix-set A2-Spoke-family
  112.31.1.0/24,
  112.32.1.0/24,
  152.31.1.0/24,
  132.30.1.0/24,
  102.9.9.102/32,
  103.31.31.103/32,
  183.31.1.0/24,
  183.32.1.0/24
end-set
!
route-policy A2-Spoke-family
  if destination in A2-Spoke-family then
    pass
  else
    drop
  endif
end-policy
!

router bgp 1
 vrf A2-Hub-3in
  rd 4000:3
  address-family ipv4 unicast
   route-target download
   redistribute connected
  !
  neighbor 113.113.114.9
   remote-as 12
   address-family ipv4 unicast
```

route-policy A2-Spoke-family in ------à leaking the selected spoke routes with hub route targets so they can be imported by the spoke sites with RPF A2-Hub-3in.

```
    route-policy pass-all out
   !
  !
 !
!
router bgp 1
 vrf A2-Hub-3out
  rd 4000:4
  address-family ipv4 unicast
   route-target download
   redistribute connected
  !
 !
!
router bgp 1
 vrf A2-Hub-2
  rd 4000:2
  address-family ipv4 unicast
   route-target download
   redistribute connected
   redistribute eigrp 20 match internal external metric 1000
  !
 !
!
multicast-routing
 vrf A2-Hub-2
  address-family ipv4
   log-traps
   rate-per-route
   interface all enable
   accounting per-prefix
  !
 !
!
multicast-routing
```

```
 vrf A2-Hub-3in
  address-family ipv4
   log-traps
   rate-per-route
   interface all enable
   accounting per-prefix
  !
 !
!

multicast-routing
 vrf A2-Hub-3out
  address-family ipv4
   log-traps
   rate-per-route
   interface all enable
   accounting per-prefix
  !


 !
!
router pim
 vrf A2-Hub-2
  address-family ipv4
   rpf topology route-policy A2-Hub-Policy
   bsr relay vrf A2-Spoke-3 listen
   auto-rp relay vrf A2-Hub-Tunnel
  !
 !
!
router pim
vrf A2-Hub-3in
address-family ipv4
rpf topology route-policy A2-Hub-Policy
!
!
!
router pim
vrf A2-Hub-3out
address-family ipv4
rpf topology route-policy A2-Hub-Policy
!
!
!
```

```
route-policy A2-Hub-Policy
if extcommunity rt matches-any (4000:1, 4000:2, 4000:3, 4000:4) then
set rpf-topology vrf A2-Hub-Tunnel
elseif extcommunity rt matches-any (4001:1, 4001:2, 4001:3, 4001:4) then
set rpf-topology vrf A2-Spoke-Tunnel
else
pass
endif
end-policy

!
```

Any CE-PE protocol can be used. In this example, A2-Hub-3out exports all the hub and spoke routes to CE2 through EIGRP.

A2-Hub-3in uses route policy A2-Spoke-family to re-import selected spoke routes into PE2 through BGP.

```
router eigrp 20
vrf A2-Hub-3out
address-family ipv4
default-metric 1000 1 255 1 1500
autonomous-system 20
redistribute bgp 1
interface GigabitEthernet0/1/0/1.13
hold-time 60

   !

  !

 !

!
```

CE2:

Here A2-Hub-3in and A2-Hub-3out interfaces are in vrf default and not in a hub site vrf.

```
interface GigabitEthernet0/12/1/0.12
description To PE2 or vrf A2-Hub-3in
ipv4 address 113.113.114.9 255.255.255.252
encapsulation dot1q 3001

!
interface GigabitEthernet0/12/1/0.13
description To PE2 or vrf A2-Hub-3out
ipv4 address 113.113.114.13 255.255.255.252
encapsulation dot1q 3002
!
router bgp 12
nsr
bgp graceful-restart

 address-family ipv4 unicast
redistribute connected
redistribute eigrp 20
!
neighbor 113.113.114.10   --à this is the A2-Hub-3in neighbor on PE2.
remote-as 1
address-family ipv4 unicast
route-policy pass-all in
route-policy pass-all out
 !
!
!
```

Configuring LSM based MLDP: Examples

These examples describe multiple profiles to configure MLDP based MVPN:

Rosen MLDP without BGP-Advertisement

```
vrf 1
 vpn id 1:1
 address-family ipv4 unicast
  import route-target
   1:1
  !
  export route-target
   1:1
  !
 !
!
interface Loopback0
 ipv4 address 1.1.1.1 255.255.255.255
!
route-policy mldp-1
  set core-tree mldp-default
end-policy
!
router ospf 1
 address-family ipv4 unicast
 area 0
  mpls traffic-eng
 !
!
router bgp 100 mvpn
 address-family ipv4 unicast
  redistribute connected
 !
 address-family vpnv4 unicast
 !
 address-family vpnv6 unicast
 !
 address-family ipv4 mdt
 !
 neighbor 5.5.5.5
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
  !
  address-family vpnv4 unicast
  !
  address-family vpnv6 unicast
  !
  address-family ipv4 mdt
  !
 !
 vrf 1
  rd 1:1
  address-family ipv4 unicast
   redistribute connected
  !
 !
mpls traffic-eng
 interface GigabitEthernet0/0/2/0
 !
!
mpls ldp
 router-id 1.1.1.1
 graceful-restart
 mldp
  logging internal
 !
```

```
 <all core-facing interfaces>
!
multicast-routing
 address-family ipv4
  nsf
  mdt source Loopback0
  interface all enable
  accounting per-prefix
 !
 vrf 1
  address-family ipv4
   interface all enable
   mdt default mldp ipv4 1.1.1.1
   accounting per-prefix
  !
 !
router pim
 vrf 1
  address-family ipv4
   rpf topology route-policy mldp-1
   rp-address 10.1.1.1
  !
 !
```

Rosen MLDP with BGP Advertisement

```
vrf 101
  vpn id 101:101
 address-family ipv4 unicast
  import route-target
   101:101
  !
  export route-target
   101:101
  !
 !
!
interface Loopback0
 ipv4 address 1.1.1.1 255.255.255.255
!
interface Loopback101
 vrf 101
 ipv4 address 10.1.101.1 255.255.255.255
!
route-policy mldp-101
  set core-tree mldp-default
end-policy
!
router ospf 1
 address-family ipv4 unicast
 area 0
  mpls traffic-eng
  interface Loopback0
  !
  interface Loopback1
  !
  interface GigabitEthernet0/0/2/0
  !
  interface GigabitEthernet0/3/2/1
  !
  interface GigabitEthernet0/3/2/2
  !
 !
 mpls traffic-eng router-id Loopback0
!
router bgp 100 mvpn
 address-family ipv4 unicast
  redistribute connected
 !
 address-family vpnv4 unicast
 !
```

```
 address-family vpnv6 unicast
 !
 address-family ipv4 mvpn
 !
 neighbor 5.5.5.5
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
  !
  address-family vpnv4 unicast
  !
  address-family vpnv6 unicast
  !
  address-family ipv4 mvpn
  !
 !
 vrf 101
  rd 101:101
  address-family ipv4 unicast
   redistribute connected
  !
  address-family ipv4 mvpn
  !
 !
mpls traffic-eng
 interface GigabitEthernet0/0/2/0
 !
!
mpls ldp
 router-id 1.1.1.1
 graceful-restart
 mldp
  logging internal
 !
 <all core-facing interfaces>
 !
!
multicast-routing
 address-family ipv4
  nsf
  mdt source Loopback0
  interface all enable
  accounting per-prefix
 !
!
router pim
 vrf 101
  address-family ipv4
   rpf topology route-policy mldp-101
   vpn-id 101
   rp-address 10.1.101.1
  !
 !
```

VRF In-band Profile

```
vrf 250
 address-family ipv4 unicast
  import route-target
   250:250
  !
  export route-target
   250:250
  !
 !
!
interface Loopback0
 ipv4 address 1.1.1.1 255.255.255.255
!
interface Loopback250
 vrf 250
```

```
 ipv4 address 10.1.250.1 255.255.255.255
!
route-policy mldp-250
  set core-tree mldp-inband
end-policy
!
router ospf 1
 address-family ipv4 unicast
 area 0
  mpls traffic-eng
  interface Loopback0
  !
  interface Loopback1
  !
  interface GigabitEthernet0/0/2/0
  !
  interface GigabitEthernet0/3/2/1
  !
  interface GigabitEthernet0/3/2/2
  !
 !
 mpls traffic-eng router-id Loopback0
!
router bgp 100
 address-family ipv4 unicast
  redistribute connected
 !
 address-family vpnv4 unicast
 !
 address-family vpnv6 unicast
 !
 neighbor 5.5.5.5
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
  !
  address-family vpnv4 unicast
  !
  address-family vpnv6 unicast
  !
 !
 vrf 250
  rd 250:250
  address-family ipv4 unicast
   redistribute connected
  !
  address-family ipv4 mvpn
  !
 !
mpls traffic-eng
 interface GigabitEthernet0/0/2/0
 !
!
mpls ldp
 router-id 1.1.1.1
 graceful-restart
 mldp
  logging internal
 !
 <all core-facing interfaces>
 !
!
multicast-routing
 address-family ipv4
  nsf
  mdt source Loopback0
  interface all enable
  accounting per-prefix
 !
 vrf 250
  address-family ipv4
   mdt mldp in-band-signaling
   interface all enable
```

```
  !
 !
router pim
 vrf 250
  address-family ipv4
   rpf topology route-policy mldp-250
   rp-address 10.1.250.1
  !
 !
```

Partitioned-MDT MP2MP without BGP-AD

```
vrf 251
 address-family ipv4 unicast
  import route-target
   251:251
  !
  export route-target
   251:251
  !
 !
!
interface Loopback0
 ipv4 address 1.1.1.1 255.255.255.255
!
interface Loopback251
 vrf 251
 ipv4 address 10.11.1.1 255.255.255.255
!
route-policy mldp-251
  set core-tree mldp-partitioned-mp2mp
end-policy
!
router ospf 1
 address-family ipv4 unicast
 area 0
  mpls traffic-eng
  interface Loopback0
  !
  interface Loopback1
  !
  interface GigabitEthernet0/0/2/0
  !
  interface GigabitEthernet0/3/2/1
  !
  interface GigabitEthernet0/3/2/2
  !
 !
 mpls traffic-eng router-id Loopback0
!
router bgp 100
 address-family ipv4 unicast
  redistribute connected
 !
 address-family vpnv4 unicast
 !
 address-family vpnv6 unicast
 !
!
 neighbor 5.5.5.5
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
  !
  address-family vpnv4 unicast
  !
  address-family vpnv6 unicast
  !
 !
 vrf 251
  rd 251:251
```

```
  address-family ipv4 unicast
   redistribute connected
  !
 !
mpls traffic-eng
 interface GigabitEthernet0/0/2/0
 !
!
mpls ldp
 router-id 1.1.1.1
 graceful-restart
 mldp
  logging internal
 !
 <all core-facing interfaces>
 !
!
multicast-routing
 address-family ipv4
  nsf
  mdt source Loopback0
  interface all enable
  accounting per-prefix
 !
 vrf 251
  address-family ipv4
   mdt partitioned mldp ipv4 mp2mp
   interface all enable
  !
 !
router pim
 vrf 251
  address-family ipv4
   rpf topology route-policy mldp-251
   rp-address 10.11.1.1
  !
 !
```

Partitioned-MDT MP2MP with BGP-AD

```
vrf 301
 address-family ipv4 unicast
  import route-target
   301:301
  !
  export route-target
   301:301
  !
 !
!
interface Loopback0
 ipv4 address 1.1.1.1 255.255.255.255
!
interface Loopback301
 vrf 301
 ipv4 address 10.11.51.1 255.255.255.255
!
route-policy mldp-301
  set core-tree mldp-partitioned-mp2mp
end-policy
!
router ospf 1
 address-family ipv4 unicast
 area 0
  mpls traffic-eng
  interface Loopback0
  !
  interface Loopback1
  !
  interface GigabitEthernet0/0/2/0
  !
```

```
  interface GigabitEthernet0/3/2/1
  !
  interface GigabitEthernet0/3/2/2
  !
 !
 mpls traffic-eng router-id Loopback0
!
router bgp 100
 address-family ipv4 unicast
  redistribute connected
 !
 address-family vpnv4 unicast
 !
 address-family vpnv6 unicast
 !
 address-family ipv4 mvpn
 !
 neighbor 5.5.5.5
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
  !
  address-family vpnv4 unicast
  !
  address-family vpnv6 unicast
  !
  address-family ipv4 mvpn
  !
 !
 vrf 301
  rd 301:301
  address-family ipv4 unicast
   redistribute connected
  !
  address-family ipv4 mvpn
  !
 !
mpls traffic-eng
 interface GigabitEthernet0/0/2/0
 !
!
mpls ldp
 router-id 1.1.1.1
 graceful-restart
 mldp
  logging internal
 !
 <all core-facing interfaces>
 !
!

multicast-routing
 address-family ipv4
  nsf
  mdt source Loopback0
  interface all enable
  accounting per-prefix
 !
 vrf 301
  address-family ipv4
   bgp auto-discovery mldp
   mdt partitioned mldp ipv4 mp2mp
   interface all enable
  !
 !
router pim
 vrf 301
  address-family ipv4
   rpf topology route-policy mldp-301
   rp-address 10.11.51.1
  !
 !
```

Multidirectional Selective Provider Multicast Service Instance mLDP-P2MP with BGP-Advertisement

```
vrf 401
 address-family ipv4 unicast
  import route-target
   401:401
  !
  export route-target
   401:401
  !
 !
!
interface Loopback0
 ipv4 address 1.1.1.1 255.255.255.255
!
interface Loopback401
 vrf 401
 ipv4 address 10.11.151.1 255.255.255.255
!
route-policy mldp-401
  set core-tree mldp-partitioned-p2mp
end-policy
!
router ospf 1
 address-family ipv4 unicast
 area 0
  mpls traffic-eng
  interface Loopback0
  !
  interface Loopback1
  !
  interface GigabitEthernet0/0/2/0
  !
  interface GigabitEthernet0/3/2/1
  !
  interface GigabitEthernet0/3/2/2
  !
 !
 mpls traffic-eng router-id Loopback0
!
router bgp 100
 address-family ipv4 unicast
  redistribute connected
 !
 address-family vpnv4 unicast
 !
 address-family vpnv6 unicast
 !
 address-family ipv4 mvpn
 !
 neighbor 5.5.5.5
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
  !
  address-family vpnv4 unicast
  !
  address-family vpnv6 unicast
  !
  address-family ipv4 mvpn
  !
 !
 vrf 401
  rd 401:401
  address-family ipv4 unicast
   redistribute connected
  !
  address-family ipv4 mvpn
  !
 !
mpls traffic-eng
 interface GigabitEthernet0/0/2/0
```

```
 !
!
mpls ldp
 router-id 1.1.1.1
 graceful-restart
 mldp
  logging internal
 !
 <all core-facing interfaces>
 !
!
multicast-routing
 address-family ipv4
  nsf
  mdt source Loopback0
  interface all enable
  accounting per-prefix
 !
 vrf 401
  address-family ipv4
   bgp auto-discovery mldp
   mdt partitioned mldp ipv4 p2mp
   interface all enable
  !
 !
router pim
 vrf 401
  address-family ipv4
   rpf topology route-policy mldp-401
   rp-address 10.11.151.1
  !
```

Rosen-GRE with BGP-Advertisement

```
vrf 501
  address-family ipv4 unicast
  import route-target
   501:501
  !
  export route-target
   501:501
  !
 !
!
interface Loopback0
 ipv4 address 1.1.1.1 255.255.255.255
!
interface Loopback501
 vrf 501
 ipv4 address 10.111.1.1 255.255.255.255
!

<no route policy?>

 vrf 501
  rd 501:501
  address-family ipv4 unicast
   redistribute connected
  !
  address-family ipv4 mvpn
  !
 !
router ospf 1
 address-family ipv4 unicast
 area 0
  mpls traffic-eng
  interface Loopback0
  !
  interface Loopback1
  !
  interface GigabitEthernet0/0/2/0
```

```
  !
  interface GigabitEthernet0/3/2/1
  !
  interface GigabitEthernet0/3/2/2
  !
 !
 mpls traffic-eng router-id Loopback0
!
router bgp 100
 address-family ipv4 unicast
  redistribute connected
 !
 address-family vpnv4 unicast
 !
 address-family vpnv6 unicast
 !
 address-family ipv4 mvpn
 !
 neighbor 5.5.5.5
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
  !
  address-family vpnv4 unicast
  !
  address-family vpnv6 unicast
  !
  address-family ipv4 mvpn
  !
 !
 vrf 501
  rd 501:501
  address-family ipv4 unicast
   redistribute connected
  !
  address-family ipv4 mvpn
  !
 !
mpls traffic-eng
 interface GigabitEthernet0/0/2/0
 !
!
mpls ldp
 router-id 1.1.1.1
 graceful-restart
 mldp
  logging internal
 !
 <all core-facing interfaces>
 !
!
multicast-routing
 address-family ipv4
  nsf
  mdt source Loopback0
  interface all enable
  accounting per-prefix
 !
 vrf 501
  address-family ipv4
   bgp auto-discovery pim
   mdt default ipv4 232.1.1.1
   interface all enable
  !
 !
router pim
 vrf 501
  address-family ipv4
   rp-address 10.111.1.1
  !
 !
```

Additional References

Related Documents

Related Topic	Document Title
Multicast command reference document	*Cisco ASR 9000 Series Aggregation Services Router Multicast Command Reference*
Getting started material	*Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide*
Modular quality of service command reference document	*Cisco ASR 9000 Series Aggregation Services Router Modular Quality of Service Command Reference*
Routing command reference and configuration documents	*Cisco ASR 9000 Series Aggregation Services Router Routing Command Reference* *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide*
Information about user groups and task IDs	*Cisco ASR 9000 Series Aggregation Services Router System Security Configuration Guide*