Cisco IOS IP Multicast Configuration Guide

Release 12.4

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

Cisco IOS IP Multicast Configuration Guide

About Cisco IOS and Cisco IOS XE Software Documentation

Last updated: August 6, 2008

This document describes the objectives, audience, conventions, and organization used in Cisco IOS and Cisco IOS XE software documentation, collectively referred to in this document as Cisco IOS documentation. Also included are resources for obtaining technical assistance, additional documentation, and other information from Cisco. This document is organized into the following sections:

Documentation Objectives

Cisco IOS documentation describes the tasks and commands available to configure and maintain Cisco networking devices.

Audience

The Cisco IOS documentation set is i ntended for users who configure and maintain Cisco networking devices (such as routers and switches) but who may not be familiar with the configuration and maintenance tasks, the relationship among tasks, or the Cisco IOS commands necessary to perform particular tasks. The Cisco IOS documentation set is also intended for those users experienced with Cisco IOS who need to know about new features, new configuration options, and new software characteristics in the current Cisco IOS release.

Documentation Conventions

In Cisco IOS documentation, the term *router* may be used to refer to various Cisco products; for example, routers, access servers, and switches. These and other networking devices that support Cisco IOS software are shown interchangeably in examples and are used only for illustrative purposes. An example that shows one product does not necessarily mean that other products are not supported.

This section includes the following topics:

Typographic Conventions

Cisco IOS documentation uses the following typographic conventions:

Convention	Description
^ or Ctrl	Both the ^ symbol and Ctrl represent the Control (Ctrl) key on a keyboard. For example, the key combination **^D** or **Ctrl-D** means that you hold down the Control key while you press the D key. (Keys are indicated in capital letters but are not case sensitive.)
string	A string is a nonquoted set of characters shown in italics. For example, when setting a Simple Network Management Protocol (SNMP) community string to *public*, do not use quotation marks around the string; otherwise, the string will include the quotation marks.

Command Syntax Conventions

Cisco IOS documentation uses the following command syntax conventions:

Convention	Description
bold	Bold text indicates commands and keywords that you enter as shown.
italic	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional keyword or argument.
|	A vertical line, called a pipe, indicates a choice within a set of keywords or arguments.
[x | y]	Square brackets enclosing keywords or arguments separated by a pipe indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a pipe indicate a required choice.
[x {y | z}]	Braces and a pipe within square brackets indicate a required choice within an optional element.

Software Conventions

Cisco IOS uses the following program code conventions:

Convention	Description
`Courier font`	Courier font is used for information that is displayed on a PC or terminal screen.
`Bold Courier font`	Bold Courier font indicates text that the user must enter.
< >	Angle brackets enclose text that is not displayed, such as a password. Angle brackets also are used in contexts in which the italic font style is not supported; for example, ASCII text.
!	An exclamation point at the beginning of a line indicates that the text that follows is a comment, not a line of code. An exclamation point is also displayed by Cisco IOS software for certain processes.
[ ]	Square brackets enclose default responses to system prompts.

Reader Alert Conventions

The Cisco IOS documentation set uses the following conventions for reader alerts:

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Timesaver Means *the described action saves time*. You can save time by performing the action described in the paragraph.

Documentation Organization

This section describes the Cisco IOS documentation set, how it is organized, and how to access it on Cisco.com. Included are lists of configuration guides, command references, and supplementary references and resources that make up the documentation set. The following topics are included:

Cisco IOS Documentation Set

Cisco IOS documentation consists of the following:

- Release notes and caveats provide information about platform, technology, and feature support for a release and describe severity 1 (catastrophic), severity 2 (severe), and severity 3 (moderate) defects in released Cisco IOS code. Review release notes before other documents to learn whether or not updates have been made to a feature.
- Sets of configuration guides and command references organized by technology and published for each standard Cisco IOS release.
  - Configuration guides—Compilations of documents that provide informational and task-oriented descriptions of Cisco IOS features.
  - Command references—Compilations of command pages that provide detailed information about the commands used in the Cisco IOS features and processes that make up the related configuration guides. For each technology, there is a single command reference that covers all Cisco IOS releases and that is updated at each standard release.
- Lists of all the commands in a specific release and all commands that are new, modified, removed, or replaced in the release.
- Command reference book for **debug** commands. Command pages are listed in alphabetical order.
- Reference book for system messages for all Cisco IOS releases.

Cisco IOS Documentation on Cisco.com

The following sections describe the documentation organization and how to access various document types.

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

New Features List

The New Features List for each release provides a list of all features in the release with hyperlinks to the feature guides in which they are documented.

Feature Guides

Cisco IOS features are documented in feature guides. Feature guides describe one feature or a group of related features that are supported on many different software releases and platforms. Your Cisco IOS software release or platform may not support all the features documented in a feature guide. See the Feature Information table at the end of the feature guide for information about which features in that guide are supported in your software release.

Configuration Guides

Configuration guides are provided by technology and release and comprise a set of individual feature guides relevant to the release and technology.

Command References

Command reference books describe Cisco IOS commands that are supported in many different software releases and on many different platforms. The books are provided by technology. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/all_release/all_mcl.html.

Cisco IOS Supplementary Documents and Resources

Supplementary documents and resources are listed in Table 2 on page xi.

Configuration Guides, Command References, and Supplementary Resources

Table 1 lists, in alphabetical order, Cisco IOS and Cisco IOS XE software configuration guides and command references, including brief descriptions of the contents of the documents. The Cisco IOS command references are comprehensive, meaning that they include commands for both Cisco IOS software and Cisco IOS XE software, for all releases. The configuration guides and command references support many different software releases and platforms. Your Cisco IOS software release or platform may not support all these technologies.

For additional information about configuring and operating specific networking devices, go to the Product Support area of Cisco.com at http://www.cisco.com/web/psa/products/index.html.

Table 2 lists documents and resources that supplement the Cisco IOS software configuration guides and command references. These supplementary resources include release notes and caveats; master command lists; new, modified, removed, and replaced command lists; system messages; and the debug command reference.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS AppleTalk Configuration Guide *Cisco IOS XE AppleTalk Configuration Guide* *Cisco IOS AppleTalk Command Reference*	AppleTalk protocol.
Cisco IOS Asynchronous Transfer Mode Configuration Guide *Cisco IOS Asynchronous Transfer Mode Command Reference*	LAN ATM, multiprotocol over ATM (MPoA), and WAN ATM.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS Bridging and IBM Networking Configuration Guide *Cisco IOS Bridging Command Reference* *Cisco IOS IBM Networking Command Reference*	• Transparent and source-route transparent (SRT) bridging, source-route bridging (SRB), Token Ring Inter-Switch Link (TRISL), and token ring route switch module (TRRSM). • Data-link switching plus (DLSw+), serial tunnel (STUN), block serial tunnel (BSTUN); logical link control, type 2 (LLC2), synchronous data link control (SDLC); IBM Network Media Translation, including Synchronous Data Logical Link Control (SDLLC) and qualified LLC (QLLC); downstream physical unit (DSPU), Systems Network Architecture (SNA) service point, SNA frame relay access, advanced peer-to-peer networking (APPN), native client interface architecture (NCIA) client/server topologies, and IBM Channel Attach.
Cisco IOS Broadband and DSL Configuration Guide *Cisco IOS XE Broadband and DSL Configuration Guide* *Cisco IOS Broadband and DSL Command Reference*	Point-to-Point Protocol (PPP) over ATM (PPPoA) and PPP over Ethernet (PPPoE).
Cisco IOS Carrier Ethernet Configuration Guide *Cisco IOS Carrier Ethernet Command Reference*	Connectivity fault management (CFM), Ethernet Local Management Interface (ELMI), IEEE 802.3ad link bundling, Link Layer Discovery Protocol (LLDP), media endpoint discovery (MED), and operations, administration, and maintenance (OAM).
Cisco IOS Configuration Fundamentals Configuration Guide *Cisco IOS XE Configuration Fundamentals Configuration Guide* *Cisco IOS Configuration Fundamentals Command Reference*	Autoinstall, Setup, Cisco IOS command-line interface (CLI), Cisco IOS file system (IFS), Cisco IOS web browser user interface (UI), basic file transfer services, and file management.
Cisco IOS DECnet Configuration Guide *Cisco IOS XE DECnet Configuration Guide* *Cisco IOS DECnet Command Reference*	DECnet protocol.
Cisco IOS Dial Technologies Configuration Guide *Cisco IOS XE Dial Technologies Configuration Guide* *Cisco IOS Dial Technologies Command Reference*	Asynchronous communications, dial backup, dialer technology, dial-in terminal services and AppleTalk remote access (ARA), large scale dialout, dial-on-demand routing, dialout, modem and resource pooling, ISDN, multilink PPP (MLP), PPP, virtual private dialup network (VPDN).
Cisco IOS Flexible NetFlow Configuration Guide *Cisco IOS Flexible NetFlow Command Reference*	Flexible NetFlow.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS H.323 Configuration Guide	Gatekeeper enhancements for managed voice services, Gatekeeper Transaction Message Protocol, gateway codec order preservation and shutdown control, H.323 dual tone multifrequency relay, H.323 version 2 enhancements, Network Address Translation (NAT) support of H.323 v2 Registration, Admission, and Status (RAS) protocol, tokenless call authorization, and VoIP gateway trunk and carrier-based routing.
Cisco IOS High Availability Configuration Guide *Cisco IOS XE High Availability Configuration Guide* *Cisco IOS High Availability Command Reference*	A variety of High Availability (HA) features and technologies that are available for different network segments (from enterprise access to service provider core) to facilitate creation of end-to-end highly available networks. Cisco IOS HA features and technologies can be categorized in three key areas: system-level resiliency, network-level resiliency, and embedded management for resiliency.
Cisco IOS Integrated Session Border Controller Command Reference	A VoIP-enabled device that is deployed at the edge of networks. An SBC is a toolkit of functions, such as signaling interworking, network hiding, security, and quality of service (QoS).
Cisco IOS Intelligent Service Gateway Configuration Guide *Cisco IOS Intelligent Service Gateway Command Reference*	Subscriber identification, service and policy determination, session creation, session policy enforcement, session life-cycle management, accounting for access and service usage, session state monitoring.
Cisco IOS Interface and Hardware Component Configuration Guide *Cisco IOS XE Interface and Hardware Component Configuration Guide* *Cisco IOS Interface and Hardware Component Command Reference*	LAN interfaces, logical interfaces, serial interfaces, virtual interfaces, and interface configuration.
Cisco IOS IP Addressing Services Configuration Guide *Cisco IOS XE Addressing Services Configuration Guide* *Cisco IOS IP Addressing Services Command Reference*	Address Resolution Protocol (ARP), Network Address Translation (NAT), Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP), and Next Hop Address Resolution Protocol (NHRP).
Cisco IOS IP Application Services Configuration Guide *Cisco IOS XE IP Application Services Configuration Guide* *Cisco IOS IP Application Services Command Reference*	Enhanced Object Tracking (EOT), Gateway Load Balancing Protocol (GLBP), Hot Standby Router Protocol (HSRP), IP Services, Server Load Balancing (SLB), Stream Control Transmission Protocol (SCTP), TCP, Web Cache Communication Protocol (WCCP), User Datagram Protocol (UDP), and Virtual Router Redundancy Protocol (VRRP).
Cisco IOS IP Mobility Configuration Guide *Cisco IOS IP Mobility Command Reference*	Mobile ad hoc networks (MANet) and Cisco mobile networks.
Cisco IOS IP Multicast Configuration Guide *Cisco IOS XE IP Multicast Configuration Guide* *Cisco IOS IP Multicast Command Reference*	Protocol Independent Multicast (PIM) sparse mode (PIM-SM), bidirectional PIM (bidir-PIM), Source Specific Multicast (SSM), Multicast Source Discovery Protocol (MSDP), Internet Group Management Protocol (IGMP), and Multicast VPN (MVPN).

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS IP Routing Protocols Configuration Guide *Cisco IOS XE IP Routing Protocols Configuration Guide* *Cisco IOS IP Routing Protocols Command Reference*	Border Gateway Protocol (BGP), multiprotocol BGP, multiprotocol BGP extensions for IP multicast, bidirectional forwarding detection (BFD), Enhanced Interior Gateway Routing Protocol (EIGRP), Interior Gateway Routing Protocol (IGRP), Intermediate System-to-Intermediate System (IS-IS), on-demand routing (ODR), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP).
Cisco IOS IP SLAs Configuration Guide *Cisco IOS XE IP SLAs Configuration Guide* *Cisco IOS IP SLAs Command Reference*	Cisco IOS IP Service Level Agreements (IP SLAs).
Cisco IOS IP Switching Configuration Guide *Cisco IOS XE IP Switching Configuration Guide* *Cisco IOS IP Switching Command Reference*	Cisco Express Forwarding, fast switching, and Multicast Distributed Switching (MDS).
Cisco IOS IPv6 Configuration Guide *Cisco IOS XE IPv6 Configuration Guide* *Cisco IOS IPv6 Command Reference*	For IPv6 features, protocols, and technologies, go to the IPv6 "Start Here" document at the following URL: http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-roadmap.html
Cisco IOS ISO CLNS Configuration Guide *Cisco IOS XE ISO CLNS Configuration Guide* *Cisco IOS ISO CLNS Command Reference*	ISO connectionless network service (CLNS).
Cisco IOS LAN Switching Configuration Guide *Cisco IOS XE LAN Switching Configuration Guide* *Cisco IOS LAN Switching Command Reference*	VLANs, Inter-Switch Link (ISL) encapsulation, IEEE 802.10 encapsulation, IEEE 802.1Q encapsulation, and multilayer switching (MLS).
Cisco IOS Mobile Wireless Gateway GPRS Support Node Configuration Guide *Cisco IOS Mobile Wireless Gateway GPRS Support Node Command Reference*	Cisco IOS Gateway GPRS Support Node (GGSN) in a 2.5-generation general packet radio service (GPRS) and 3-generation universal mobile telecommunication system (UMTS) network.
Cisco IOS Mobile Wireless Home Agent Configuration Guide *Cisco IOS Mobile Wireless Home Agent Command Reference*	Cisco Mobile Wireless Home Agent, an anchor point for mobile terminals for which mobile IP or proxy mobile IP services are provided.
Cisco IOS Mobile Wireless Packet Data Serving Node Configuration Guide *Cisco IOS Mobile Wireless Packet Data Serving Node Command Reference*	Cisco Packet Data Serving Node (PDSN), a wireless gateway that is between the mobile infrastructure and standard IP networks and that enables packet data services in a code division multiple access (CDMA) environment.
Cisco IOS Mobile Wireless Radio Access Networking Configuration Guide *Cisco IOS Mobile Wireless Radio Access Networking Command Reference*	Cisco IOS radio access network products.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS Multiprotocol Label Switching Configuration Guide *Cisco IOS XE Multiprotocol Label Switching Configuration Guide* *Cisco IOS Multiprotocol Label Switching Command Reference*	MPLS Label Distribution Protocol (LDP), MPLS Layer 2 VPNs, MPLS Layer 3 VPNs, MPLS Traffic Engineering (TE), and MPLS Embedded Management (EM) and MIBs.
Cisco IOS Multi-Topology Routing Configuration Guide *Cisco IOS Multi-Topology Routing Command Reference*	Unicast and multicast topology configurations, traffic classification, routing protocol support, and network management support.
Cisco IOS NetFlow Configuration Guide *Cisco IOS XE NetFlow Configuration Guide* *Cisco IOS NetFlow Command Reference*	Network traffic data analysis, aggregation caches, export features.
Cisco IOS Network Management Configuration Guide *Cisco IOS XE Network Management Configuration Guide* *Cisco IOS Network Management Command Reference*	Basic system management; system monitoring and logging; troubleshooting, logging, and fault management; Cisco Discovery Protocol; Cisco IOS Scripting with Tool Control Language (Tcl); Cisco networking services (CNS); DistributedDirector; Embedded Event Manager (EEM); Embedded Resource Manager (ERM); Embedded Syslog Manager (ESM); HTTP; Remote Monitoring (RMON); SNMP; and VPN Device Manager Client for Cisco IOS Software (XSM Configuration).
Cisco IOS Novell IPX Configuration Guide *Cisco IOS XE Novell IPX Configuration Guide* *Cisco IOS Novell IPX Command Reference*	Novell Internetwork Packet Exchange (IPX) protocol.
Cisco IOS Optimized Edge Routing Configuration Guide *Cisco IOS Optimized Edge Routing Command Reference*	Optimized edge routing (OER) monitoring, policy configuration, routing control, logging and reporting, and VPN IPsec/generic routing encapsulation (GRE) tunnel interface optimization.
Cisco IOS Quality of Service Solutions Configuration Guide *Cisco IOS XE Quality of Service Solutions Configuration Guide* *Cisco IOS Quality of Service Solutions Command Reference*	Class-based weighted fair queuing (CBWFQ), custom queuing, distributed traffic shaping (DTS), generic traffic shaping (GTS), IP- to-ATM class of service (CoS), low latency queuing (LLQ), modular QoS CLI (MQC), Network-Based Application Recognition (NBAR), priority queuing, Security Device Manager (SDM), Multilink PPP (MLPPP) for QoS, header compression, AutoQoS, QoS features for voice, Resource Reservation Protocol (RSVP), weighted fair queuing (WFQ), and weighted random early detection (WRED).
Cisco IOS Security Configuration Guide *Cisco IOS XE Security Configuration Guide* *Cisco IOS Security Command Reference*	Access control lists (ACLs), authentication, authorization, and accounting (AAA), firewalls, IP security and encryption, neighbor router authentication, network access security, network data encryption with router authentication, public key infrastructure (PKI), RADIUS, TACACS+, terminal access security, and traffic filters.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS Service Selection Gateway Configuration Guide *Cisco IOS Service Selection Gateway Command Reference*	Subscriber authentication, service access, and accounting.
Cisco IOS Software Activation Configuration Guide *Cisco IOS Software Activation Command Reference*	An orchestrated collection of processes and components to activate Cisco IOS software feature sets by obtaining and validating Cisco software licenses.
Cisco IOS Software Modularity Installation and Configuration Guide *Cisco IOS Software Modularity Command Reference*	Installation and basic configuration of software modularity images, including installations on single and dual route processors, installation rollbacks, software modularity binding, software modularity processes and patches.
Cisco IOS Terminal Services Configuration Guide *Cisco IOS Terminal Services Command Reference* *Cisco IOS XE Terminal Services Command Reference*	DEC, local-area transport (LAT), and X.25 packet assembler/disassembler (PAD).
Cisco IOS Virtual Switch Command Reference	Virtual switch redundancy, high availability, and packet handling; converting between standalone and virtual switch modes; virtual switch link (VSL); Virtual Switch Link Protocol (VSLP). **Note** For information about virtual switch configuration, refer to the product-specific software configuration information for the Cisco Catalyst 6500 series switch or for the Metro Ethernet 6500 series switch.
Cisco IOS Voice Configuration Library *Cisco IOS Voice Command Reference*	Cisco IOS support for voice call control protocols, interoperability, physical and virtual interface management, and troubleshooting. The library includes documentation for IP telephony applications.
Cisco IOS VPDN Configuration Guide *Cisco IOS XE VPDN Configuration Guide* *Cisco IOS VPDN Command Reference*	Layer 2 Tunneling Protocol (L2TP) dial-out load balancing and redundancy, L2TP extended failover, L2TP security VPDN, multihop by Dialed Number Identification Service (DNIS), timer and retry enhancements for L2TP and Layer 2 Forwarding (L2F), RADIUS Attribute 82: tunnel assignment ID, shell-based authentication of VPDN users, tunnel authentication via RADIUS on tunnel terminator.
Cisco IOS Wide-Area Networking Configuration Guide *Cisco IOS XE Wide-Area Networking Configuration Guide* *Cisco IOS Wide-Area Networking Command Reference*	Frame Relay, Layer 2 Tunneling Protocol Version 3 (L2TPv3), Link Access Procedure, Balanced (LAPB), Switched Multimegabit Data Service (SMDS), and X.25.
Cisco IOS Wireless LAN Configuration Guide *Cisco IOS Wireless LAN Command Reference*	Broadcast key rotation, IEEE 802.11x support, IEEE 802.1x authenticator, IEEE 802.1x local authentication service for Extensible Authentication Protocol-Flexible Authentication via Secure Tunneling (EAP-FAST), Multiple Basic Service Set ID (BSSID), Wi-Fi Multimedia (WMM) required elements, and Wi-Fi Protected Access (WPA).

Table 2 Cisco IOS Supplementary Documents and Resources

Document Title	Description
Cisco IOS Master Command List, All Releases	Alphabetical list of all the commands documented in all Cisco IOS releases.
Cisco IOS New, Modified, Removed, and Replaced Commands	List of all the new, modified, removed, and replaced commands for a Cisco IOS release.
Cisco IOS Software System Messages	List of Cisco IOS system messages and descriptions. System messages may indicate problems with your system; be informational only; or may help diagnose problems with communications lines, internal hardware, or the system software.
Cisco IOS Debug Command Reference	Alphabetical list of **debug** commands including brief descriptions of use, command syntax, and usage guidelines.
Release Notes and Caveats	Information about new and changed features, system requirements, and other useful information about specific software releases; information about defects in specific Cisco IOS software releases.
MIBs	Files used for network monitoring. To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator at the following URL: http://www.cisco.com/go/mibs
RFCs	Standards documents maintained by the Internet Engineering Task Force (IETF) that Cisco IOS documentation references where applicable. The full text of referenced RFCs may be obtained at the following URL: http://www.rfc-editor.org/

Additional Resources and Documentation Feedback

What's New in Cisco Product Documentation is published monthly and describes all new and revised Cisco technical documentation. The *What's New in Cisco Product Documentation* publication also provides information about obtaining the following resources:

- Technical documentation
- Cisco product security overview
- Product alerts and field notices
- Technical assistance

Cisco IOS technical documentation includes embedded feedback forms where you can rate documents and provide suggestions for improvement. Your feedback helps us improve our documentation.

Using the Command-Line Interface in Cisco IOS and Cisco IOS XE Software

Last updated: August 6, 2008

This document provides basic information about the command-line interface (CLI) in Cisco IOS and Cisco IOS XE software and how you can use some of the CLI features. This document contains the following sections:

For more information about using the CLI, see the "Using the Cisco IOS Command-Line Interface" section of the *Cisco IOS Configuration Fundamentals Configuration Guide*.

For information about the software documentation set, see the "About Cisco IOS and Cisco IOS XE Software Documentation" document.

Initially Configuring a Device

Initially configuring a device varies by platform. For information about performing an initial configuration, see the hardware installation documentation that is provided with the original packaging of the product or go to the Product Support area of Cisco.com at http://www.cisco.com/web/psa/products/index.html.

After you have performed the initial configuration and connected the device to your network, you can configure the device by using the console port or a remote access method, such as Telnet or Secure Shell (SSH), to access the CLI or by using the configuration method provided on the device, such as Security Device Manager.

Changing the Default Settings for a Console or AUX Port

There are only two changes that you can make to a console port and an AUX port:

- Change the port speed with the **config-register 0x** command. Changing the port speed is not recommended. The well-known default speed is 9600.
- Change the behavior of the port; for example, by adding a password or changing the timeout value.

Note The AUX port on the Route Processor (RP) installed in a Cisco ASR1000 series router does not serve any useful customer purpose and should be accessed only under the advisement of a customer support representative.

Using the CLI

This section describes the following topics:

Understanding Command Modes

The CLI command mode structure is hierarchical, and each mode supports a set of specific commands. This section describes the most common of the many modes that exist.

Table 1 lists common command modes with associated CLI prompts, access and exit methods, and a brief description of how each mode is used.

Table 1 CLI Command Modes

Command Mode	Access Method	Prompt	Exit Method	Mode Usage
User EXEC	Log in.	`Router>`	Issue the **logout** or **exit** command.	• Change terminal settings. • Perform basic tests. • Display device status.
Privileged EXEC	From user EXEC mode, issue the **enable** command.	`Router#`	Issue the **disable** command or the **exit** command to return to user EXEC mode.	• Issue **show** and **debug** commands. • Copy images to the device. • Reload the device. • Manage device configuration files. • Manage device file systems.
Global configuration	From privileged EXEC mode, issue the **configure terminal** command.	`Router(config)#`	Issue the **exit** command or the **end** command to return to privileged EXEC mode.	Configure the device.
Interface configuration	From global configuration mode, issue the **interface** command.	`Router(config-if)#`	Issue the **exit** command to return to global configuration mode or the **end** command to return to privileged EXEC mode.	Configure individual interfaces.
Line configuration	From global configuration mode, issue the **line vty** or **line console** command.	`Router(config-line)#`	Issue the **exit** command to return to global configuration mode or the **end** command to return to privileged EXEC mode.	Configure individual terminal lines.

Table 1 CLI Command Modes (continued)

Command Mode	Access Method	Prompt	Exit Method	Mode Usage
ROM monitor	From privileged EXEC mode, issue the **reload** command. Press the **Break** key during the first 60 seconds while the system is booting.	`rommon # >` The # symbol represents the line number and increments at each prompt.	Issue the **continue** command.	• Run as the default operating mode when a valid image cannot be loaded. • Access the fall-back procedure for loading an image when the device lacks a valid image and cannot be booted. • Perform password recovery when a CTRL-Break sequence is issued within 60 seconds of a power-on or reload event.
Diagnostic (available only on the Cisco ASR1000 series router)	The router boots or enters diagnostic mode in the following scenarios. When a Cisco IOS process or processes fail, in most scenarios the router will reload. • A user-configured access policy was configured using the **transport-map** command, which directed the user into diagnostic mode. • The router was accessed using an RP auxiliary port. • A break signal (**Ctrl-C**, **Ctrl-Shift-6**, or the **send break** command) was entered, and the router was configured to enter diagnostic mode when the break signal was received.	`Router(diag)#`	If a Cisco IOS process failure is the reason for entering diagnostic mode, the failure must be resolved and the router must be rebooted to exit diagnostic mode. If the router is in diagnostic mode because of a transport-map configuration, access the router through another port or using a method that is configured to connect to the Cisco IOS CLI. If the RP auxiliary port was used to access the router, use another port for access. Accessing the router through the auxiliary port is not useful for customer purposes.	• Inspect various states on the router, including the Cisco IOS state. • Replace or roll back the configuration. • Provide methods of restarting the Cisco IOS software or other processes. • Reboot hardware, such as the entire router, an RP, an ESP, a SIP, a SPA, or possibly other hardware components. • Transfer files into or off of the router using remote access methods such as FTP, TFTP, and SCP.

EXEC commands are not saved when the software reboots. Commands that you issue in a configuration mode can be saved to the startup configuration. If you save the running configuration to the startup configuration, these commands will execute when the software is rebooted. Global configuration mode is the highest level of configuration mode. From global configuration mode, you can enter a variety of other configuration modes, including protocol-specific modes.

ROM monitor mode is a separate mode that is used when the software cannot load properly. If a valid software image is not found when the software boots or if the configuration file is corrupted at startup, the software might enter ROM monitor mode. Use the question symbol (?) to view the commands that you can use while the device is in ROM monitor mode.

```
rommon 1 > ?
alias               set and display aliases command
boot                boot up an external process
confreg             configuration register utility
cont                continue executing a downloaded image
context             display the context of a loaded image
cookie              display contents of cookie PROM in hex
.
.
.
rommon 2 >
```

The following example shows how the command prompt changes to indicate a different command mode:

```
Router> enable
Router# configure terminal
Router(config)# interface ethernet 1/1
Router(config-if)# ethernet
Router(config-line)# exit
Router(config)# end
Router#
```

Note A keyboard alternative to the **end** command is Ctrl-Z.

Using the Interactive Help Feature

The CLI includes an interactive Help feature. Table 2 describes how to use the Help feature.

Table 2 CLI Interactive Help Commands

Command	Purpose
help	Provides a brief description of the help feature in any command mode.
?	Lists all commands available for a particular command mode.
*partial command***?**	Provides a list of commands that begin with the character string (no space between the command and the question mark).
partial command<**Tab**>	Completes a partial command name (no space between the command and <Tab>).
command **?**	Lists the keywords, arguments, or both associated with the command (space between the command and the question mark).
command keyword **?**	Lists the arguments that are associated with the keyword (space between the keyword and the question mark).

The following examples show how to use the help commands:

help

```
Router> help

Help may be requested at any point in a command by entering a question mark '?'.  If
nothing matches, the help list will be empty and you must backup until entering a '?'
shows the available options.

Two styles of help are provided:

1. Full help is available when you are ready to enter a command argument (e.g. 'show ?')
and describes each possible argument.

2. Partial help is provided when an abbreviated argument is entered and you want to know
what arguments match the input (e.g. 'show pr?'.)
```

?

```
Router# ?
Exec commands:
  access-enable        Create a temporary access-List entry
  access-profile       Apply user-profile to interface
  access-template      Create a temporary access-List entry
  alps                 ALPS exec commands
  archive              manage archive files
<snip>
```

***partial command*?**

```
Router(config)# zo?
zone  zone-pair
```

***partial command*<Tab>**

```
Router(config)# we<Tab> webvpn
```

***command* ?**

```
Router(config-if)# pppoe ?
  enable        Enable pppoe
  max-sessions  Maximum PPPOE sessions
```

***command keyword* ?**

```
Router(config-if)# pppoe enable ?
  group  attach a BBA group
  <cr>
```

Understanding Command Syntax

Command syntax is the format in which a command should be entered in the CLI. Commands include the name of the command, keywords, and arguments. Keywords are alphanumeric strings that are used literally. Arguments are placeholders for values that a user must supply. Keywords and arguments may be required or optional.

Specific conventions convey information about syntax and command elements. Table 3 describes these conventions.

Table 3 CLI Syntax Conventions

Symbol/Text	Function	Notes
< > (angle brackets)	Indicate that the option is an argument.	Sometimes arguments are displayed without angle brackets.
A.B.C.D.	Indicates that you must enter a dotted decimal IP address.	Angle brackets (< >) are not always used to indicate that an IP address is an argument.
WORD (all capital letters)	Indicates that you must enter one word.	Angle brackets (< >) are not always used to indicate that a WORD is an argument.
LINE (all capital letters)	Indicates that you must enter more than one word.	Angle brackets (< >) are not always used to indicate that a LINE is an argument.
<cr> (carriage return)	Indicates the end of the list of available keywords and arguments, and also indicates when keywords and arguments are optional. When <cr> is the only option, you have reached the end of the branch or the end of the command if the command has only one branch.	—

The following examples show syntax conventions:

```
Router(config)# ethernet cfm domain ?
  WORD  domain name
Router(config)# ethernet cfm domain dname ?
  level
Router(config)# ethernet cfm domain dname level ?
  <0-7>  maintenance level number
Router(config)# ethernet cfm domain dname level 7 ?
  <cr>
Router(config)# snmp-server file-transfer access-group 10 ?
  protocol  protocol options
  <cr>
Router(config)# logging host ?
  Hostname or A.B.C.D  IP address of the syslog server
  ipv6                 Configure IPv6 syslog server
Router(config)# snmp-server file-transfer access-group 10 ?
  protocol  protocol options
  <cr>
```

Understanding Enable and Enable Secret Passwords

Some privileged EXEC commands are used for actions that impact the system, and it is recommended that you set a password for these commands to prevent unauthorized use. Two types of passwords, enable (not encrypted) and enable secret (encrypted), can be set. The following commands set these passwords and are issued in global configuration mode:

- **enable** *password*
- **enable secret** *password*

Using an enable secret password is recommended because it is encrypted and more secure than the enable password. When you use an enable secret password, text is encrypted (unreadable) before it is written to the config.text file. When you use an enable password, the text is written as entered (readable) to the config.text file.

Each type of password is case sensitive, can contain from 1 to 25 uppercase and lowercase alphanumeric characters, and can start with a number. Spaces are also valid password characters; for example, "two words" is a valid password. Leading spaces are ignored, but trailing spaces are recognized.

Note Both password commands have numeric keywords that are single integer values. If you choose a number for the first character of your password followed by a space, the system will read the number as if it were the numeric keyword and not as part of your password.

When both passwords are set, the enable secret password takes precedence over the enable password.

To remove a password, use the **no** form of the commands: **no enable** *password* or **no enable secret** *password*.

For more information about password recovery procedures for Cisco products, see http://www.cisco.com/en/US/products/sw/iosswrel/ps1831/products_tech_note09186a00801746e6.shtml.

Using the Command History Feature

The CLI command history feature saves the commands you enter during a session in a command history buffer. The default number of commands saved is 10, but the number is configurable within the range of 0 to 256. This command history feature is particularly useful for recalling long or complex commands.

To change the number of commands saved in the history buffer for a terminal session, issue the **terminal history size** command:

```
Router# terminal history size num
```

A command history buffer is also available in line configuration mode with the same default and configuration options. To set the command history buffer size for a terminal session in line configuration mode, issue the **history** command:

```
Router(config-line)# history [size num]
```

To recall commands from the history buffer, use the following methods:

- Press Ctrl-P or the up arrow key—Recalls commands beginning with the most recent command. Repeat the key sequence to recall successively older commands.

- Press Ctrl-N or the down arrow key—Recalls the most recent commands in the history buffer after they have been recalled using Ctrl-P or the up arrow key. Repeat the key sequence to recall successively more recent commands.

Note The arrow keys function only on ANSI-compatible terminals such as the VT100.

- Issue the **show history** command in user EXEC or privileged EXEC mode—Lists the most recent commands that you entered. The number of commands that are displayed is determined by the setting of the **terminal history size** and **history** commands.

  The CLI command history feature is enabled by default. To disable this feature for a terminal session, issue the **terminal no history** command in user EXEC or privileged EXEC mode or the **no history** command in line configuration mode.

Abbreviating Commands

Typing a complete command name is not always required for the command to execute. The CLI recognizes an abbreviated command when the abbreviation contains enough characters to uniquely identify the command. For example, the **show version** command can be abbreviated as **sh ver**. It cannot be abbreviated as **s ver** because **s** could mean **show**, **set**, or **systat**. The **sh v** abbreviation also is not valid because the **show** command has **vrrp** as a keyword in addition to **version**. (Command and keyword examples from Cisco IOS Release 12.4(13)T.)

Using Aliases for CLI Commands

To save time and the repetition of entering the same command multiple times, you can use a command alias. An alias can be configured to do anything that can be done at the command line, but an alias cannot move between modes, type in passwords, or perform any interactive functions.

Table 4 shows the default command aliases.

Table 4 Default Command Aliases

Command Alias	Original Command
h	help
lo	logout
p	ping
s	show
u or **un**	undebug
w	where

To create a command alias, issue the **alias** command in global configuration mode. The syntax of the command is **alias** *mode command-alias original-command*. Following are some examples:

- Router(config)# **alias exec prt partition**—privileged EXEC mode
- Router(config)# **alias configure sb source-bridge**—global configuration mode
- Router(config)# **alias interface rl rate-limit**—interface configuration mode

To view both default and user-created aliases, issue the **show alias** command.

For more information about the **alias** command, see http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html.

Using the no and default Forms of Commands

Most configuration commands have a **no** form that is used to reset a command to its default value or disable a feature or function. For example, the **ip routing** command is enabled by default. To disable this command, you would issue the **no ip routing** command. To re-enable IP routing, you would issue the **ip routing** command.

Configuration commands may also have a **default** form, which returns the command settings to their default values. For commands that are disabled by default, using the **default** form has the same effect as using the **no** form of the command. For commands that are enabled by default and have default settings, the **default** form enables the command and returns the settings to their default values.

The **no** and **default** forms of commands are described in the command pages of command references.

Using the debug Command

A **debug** command produces extensive output that helps you troubleshoot problems in your network. These commands are available for many features and functions within Cisco IOS and Cisco IOS XE software. Some **debug** commands are **debug all**, **debug aaa accounting**, and **debug mpls packets**. To use **debug** commands during a Telnet session with a device, you must first enter the **terminal monitor** command. To turn off debugging completely, you must enter the **undebug all** command.

For more information about **debug** commands, see the *Cisco IOS Debug Command Reference* at http://www.cisco.com/en/US/docs/ios/debug/command/reference/db_book.html.

Caution Debugging is a high priority and high CPU utilization process that can render your device unusable. Use **debug** commands only to troubleshoot specific problems. The best times to run debugging are during periods of low network traffic and when few users are interacting with the network. Debugging during these periods decreases the likelihood that the **debug** command processing overhead will affect network performance or user access or response times.

Filtering Output Using Output Modifiers

Many commands produce lengthy output that may use several screens to display. Using output modifiers, you can filter this output to show only the information that you want to see.

Three output modifiers are available and are described as follows:

- **begin** *regular expression*—Displays the first line in which a match of the regular expression is found and all lines that follow.
- **include** *regular expression*—Displays all lines in which a match of the regular expression is found.
- **exclude** *regular expression*—Displays all lines except those in which a match of the regular expression is found.

To use one of these output modifiers, type the command followed by the pipe symbol (|), the modifier, and the regular expression that you want to search for or filter. A regular expression is a case-sensitive alphanumeric pattern. It can be a single character or number, a phrase, or a more complex string.

The following example illustrates how to filter output of the **show interface** command to display only lines that include the expression "protocol."

```
Router# show interface | include protocol

FastEthernet0/0 is up, line protocol is up
Serial4/0 is up, line protocol is up
Serial4/1 is up, line protocol is up
Serial4/2 is administratively down, line protocol is down
Serial4/3 is administratively down, line protocol is down
```

Understanding CLI Error Messages

You may encounter some error messages while using the CLI. Table 5 shows the common CLI error messages.

Table 5 Common CLI Error Messages

Error Message	Meaning	How to Get Help
% Ambiguous command: "show con"	You did not enter enough characters for the command to be recognized.	Reenter the command followed by a space and a question mark (?). The keywords that you are allowed to enter for the command appear.
% Incomplete command.	You did not enter all the keywords or values required by the command.	Reenter the command followed by a space and a question mark (?). The keywords that you are allowed to enter for the command appear.
% Invalid input detected at "^" marker.	You entered the command incorrectly. The caret (^) marks the point of the error.	Enter a question mark (?) to display all the commands that are available in this command mode. The keywords that you are allowed to enter for the command appear.

For more system error messages, see the following documents:

- *Cisco IOS Release 12.2SR System Message Guide*
- *Cisco IOS System Messages, Volume 1 of 2* (Cisco IOS Release 12.4)
- *Cisco IOS System Messages, Volume 2 of 2* (Cisco IOS Release 12.4)

Saving Changes to a Configuration

To save changes that you made to the configuration of a device, you must issue the **copy running-config startup-config** command or the **copy system:running-config nvram:startup-config** command. When you issue these commands, the configuration changes that you made are saved to the startup configuration and saved when the software reloads or power to the device is turned off or interrupted. The following example shows the syntax of the **copy running-config startup-config** command:

```
Router# copy running-config startup-config
Destination filename [startup-config]?
```

You press Enter to accept the startup-config filename (the default), or type a new filename and then press Enter to accept that name. The following output is displayed indicating that the configuration was saved:

```
Building configuration...
[OK]
Router#
```

On most platforms, the configuration is saved to NVRAM. On platforms with a Class A flash file system, the configuration is saved to the location specified by the CONFIG_FILE environment variable. The CONFIG_FILE variable defaults to NVRAM.

Additional Information

- "Using the Cisco IOS Command-Line Interface" section of the *Cisco IOS Configuration Fundamentals Configuration Guide*:

  http://www.cisco.com/en/US/docs/ios/fundamentals/configuration/guide/cf_cli-basics.html

  or

  "Using Cisco IOS XE Software" chapter of the *Cisco ASR1000 Series Aggregation Services Routers Software Configuration Guide*:

  http://www.cisco.com/en/US/docs/routers/asr1000/configuration/guide/chassis/using_cli.html
- Cisco Product Support Resources

  http://www.cisco.com/web/psa/products/index.html
- Support area on Cisco.com (also search for documentation by task or product)

  http://www.cisco.com/en/US/support/index.html
- *White Paper: Cisco IOS Reference Guide*

  http://www.cisco.com/en/US/products/sw/iosswrel/ps1828/products_white_paper09186a008018305e.shtml
- Software Download Center (downloads; tools; licensing, registration, advisory, and general information) (requires Cisco.com User ID and password)

  http://www.cisco.com/kobayashi/sw-center/
- Error Message Decoder, a tool to help you research and resolve error messages for Cisco IOS software

  http://www.cisco.com/pcgi-bin/Support/Errordecoder/index.cgi

- Command Lookup Tool, a tool to help you find detailed descriptions of Cisco IOS commands (requires Cisco.com user ID and password)

  http://tools.cisco.com/Support/CLILookup

- Output Interpreter, a troubleshooting tool that analyzes command output of supported **show** commands

  https://www.cisco.com/pcgi-bin/Support/OutputInterpreter/home.pl\

IP Multicast Features Roadmap

First Published: February 11, 2008
Last Updated: June 24, 2008

This feature roadmap lists the Cisco IOS features documented in the *Cisco IOS IP Multicast Configuration Guide* and maps them to the documents in which they appear. The roadmap is organized so that you can select your release train and see the features in that release. Find the feature name you are searching for and click on the URL in the "Where Documented" column to access the document containing that feature.

Feature and Release Support

Table 1 lists IP multicast feature support for the following Cisco IOS software release trains:

- Cisco IOS Releases 12.0S
- Cisco IOS Releases 12.2S
- Cisco IOS Releases 12.2SB
- Cisco IOS Releases 12.2SR
- Cisco IOS Releases 12.2SX
- Cisco IOS Releases 12.2T, 12.3, 12.3T, 12.4, and 12.4T

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 lists the most recent release of each software train first and the features in alphabetical order within the release.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Table 1 Supported IP Multicast Features

Release	Feature Name	Feature Description	Where Documented
Cisco IOS Releases 12.0S			
12.0(18)S	IGMPv3—Explicit Tracking Host, Group, and Channel	This IGMPv3—Explicit Tracking Host, Group, and Channel feature enables a multicast router to explicitly track the membership of all multicast hosts in a particular multiaccess network. This enhancement to the Cisco IOS implementation of Internet Group Management Protocol Version 3 (IGMPv3) enables the router to track each individual host that is joined to a particular group or channel.	Customizing IGMP
12.0(19)S	Extended ACL Support for IGMP to Support SSM in IPv4	The Extended ACL Support for IGMP to Support SSM in IPv4 feature enables IGMPv3 to accommodate extended access lists. IGMPv3 support of extended access lists allows you to leverage an important advantage of Source Specific Multicast (SSM) in IPv4, that of filtering IGMPv3 reports based on source address, group address, or both.	Customizing IGMP
12.0(22)S	Multicast Subsecond Convergence	The Multicast Subsecond Convergence feature comprises a comprehensive set of features and protocol enhancements that provide for improved scalability and convergence in multicast-based services. This feature set provides for the ability to scale to larger services levels and to recover multicast forwarding after service failure in subsecond time frames.	Multicast Subsecond Convergence
12.0(23)S	Multicast-VPN—IP Multicast Support of MPLS VPNs	The Multicast VPN feature provides the ability to support multicast over a Layer 3 Virtual Private Network (VPN). As enterprises extend the reach of their multicast applications, service providers can accommodate these enterprises over their Multiprotocol Label Switching (MPLS) core network. IP multicast is used to stream video, voice, and data to an MPLS VPN network core.	Configuring Multicast VPN
12.0(27)S	MSDP Compliance with IETF RFC 3618	The MSDP Compliance with IETF RFC 3618 feature enables you to configure Multicast Source Discover Protocol (MSDP) to comply with the peer-RPF forwarding rules defined in the IETF RFC 3618 specifications.	Using MSDP to Interconnect Multiple PIM-SM Domains

Table 1 Supported IP Multicast Features (continued)

Release	Feature Name	Feature Description	Where Documented
12.0(28)S	PIM Dense Mode Fallback Prevention in a Network Following RP Information Loss	The PIM Dense Mode Fallback Prevention in a Network Following RP Information Loss feature enables you to prevent PIM-DM fallback when all rendezvous points (RPs) fail. Preventing the use of dense mode is very important to multicast networks whose reliability is critical. This feature provides a mechanism to keep the multicast groups in sparse mode. This feature also allows you to block multicast traffic for groups not specifically configured.	IP Multicast Technology Overview Configuring Basic IP Multicast
12.0(29)S	BGP Multicast Inter-AS VPN	The BGP Multicast Inter-AS VPN feature introduces the IPv4 Multicast Distribution Tree (MDT) Subaddress Family Identifier (SAFI) in BGP. The MDT SAFI is a transitive multicast capable connector attribute that is defined as an IPv4 address family in BGP. The MDT SAFI is designed to support inter-AS VPN peering sessions.	Configuring Multicast VPN Inter-AS Support
12.0(29)S	Multicast VPN MIB	The Multicast VPN MIB feature introduces the capability for Simple Network Management Protocol (SNMP) monitoring of a Multicast VPN (MVPN) using the MVPN MIB (CISCO-MVPN-MIB).	Multicast VPN MIB
12.0(30)S	Multicast VPN Inter-AS Support	The Multicast VPN Inter-AS Support feature enables Multicast Distribution Trees (MDTs) used for MVPNs to span multiple autonomous systems. Benefits include increased multicast coverage to customers that require multicast to span multiple service providers in a Multiprotocol Label Switching (MPLS) Layer 3 Virtual Private Network (VPN) service with the flexibility to support all options described in RFC 4364. Additionally, the Multicast VPN Inter-AS Support feature can be used to consolidate an existing MVPN service with another MVPN service, such as the case with a company merger or acquisition.	Configuring Multicast VPN Inter-AS Support
12.0(30)S	PIM RPF Vector	The PIM RPF Vector feature enables core routers to perform RPF checks on an IP address of the exit router instead of on the source router. The address on the exit router is the RPF Vector and it is inserted in PIM join messages.	Configuring Multicast VPN Inter-AS Support
Cisco IOS Releases 12.2S			
12.2(14)S	IGMP State Limit	The IGMP State Limit feature introduces the capability to limit the number of mroute states resulting from IGMP membership states per interface, per subinterface, or globally.	Customizing IGMP

Table 1 Supported IP Multicast Features (continued)

Release	Feature Name	Feature Description	Where Documented
12.2(14)S	IGMPv3—Explicit Tracking of Hosts, Groups, and Channels	This IGMPv3—Explicit Tracking Host, Group, and Channel feature enables a multicast router to explicitly track the membership of all multicast hosts in a particular multiaccess network. This enhancement to the Cisco IOS implementation of IGMPv3 enables the router to track each individual host that is joined to a particular group or channel.	Customizing IGMP
12.2(14)S	Multicast Subsecond Convergence	The Multicast Subsecond Convergence feature comprises a comprehensive set of features and protocol enhancements that provide for improved scalability and convergence in multicast-based services. This feature set provides for the ability to scale to larger services levels and to recover multicast forwarding after service failure in subsecond time frames.	Multicast Subsecond Convergence
12.2(14)S	Multicast-VPN—IP Multicast Support of MPLS VPNs	The Multicast VPN feature provides the ability to support multicast over a Layer 3 Virtual Private Network (VPN). As enterprises extend the reach of their multicast applications, service providers can accommodate these enterprises over their Multiprotocol Label Switching (MPLS) core network. IP multicast is used to stream video, voice, and data to an MPLS VPN network core.	Configuring Multicast VPN
12.2(18)S	Source Specific Multicast (SSM) Mapping	The Source Specific Multicast (SSM) Mapping feature extends the Cisco IOS suite of SSM transition tools, which also includes URL Rendezvous Directory (URD) and Internet Group Management Protocol Version 3 Lite (IGMP v3lite). SSM mapping supports SSM transition in cases where neither URD nor IGMP v3lite is available, or when supporting SSM on the end system is impossible or unwanted due to administrative or technical reasons. SSM mapping enables you to leverage SSM for video delivery to legacy set-top boxes (STBs) that do not support IGMPv3 or for applications that do not take advantage of the IGMPv3 host stack.	Source Specific Multicast (SSM) Mapping
12.2(25)S	Extended ACL Support for IGMP to Support SSM in IPv4	The Extended ACL Support for IGMP to Support SSM in IPv4 feature enables IGMPv3 to accommodate extended access lists. IGMPv3 support of extended access lists allows you to leverage an important advantage of SSM in IPv4, that of filtering IGMPv3 reports based on source address, group address, or both.	Customizing IGMP

Table 1 Supported IP Multicast Features (continued)

Release	Feature Name	Feature Description	Where Documented
12.2(25)S	MSDP Compliance with IETF RFC 3618	The MSDP Compliance with IETF RFC 3618 feature enables you to configure MSDP to comply with the peer-RPF forwarding rules defined in the IETF RFC 3618 specifications.	Using MSDP to Interconnect Multiple PIM-SM Domains
Cisco IOS Releases 12.2SB			
12.2(27)SBC	Extended ACL Support for IGMP to Support SSM in IPv4	The Extended ACL Support for IGMP to Support SSM in IPv4 feature enables IGMPv3 to accommodate extended access lists. IGMPv3 support of extended access lists allows you to leverage an important advantage of SSM in IPv4, that of filtering IGMPv3 reports based on source address, group address, or both.	Customizing IGMP
12.2(27)SBC	MSDP Compliance with IETF RFC 3618	The MSDP Compliance with IETF RFC 3618 feature enables you to configure MSDP to comply with the peer-RPF forwarding rules defined in the IETF RFC 3618 specifications.	Using MSDP to Interconnect Multiple PIM-SM Domains
12.2(27)SBC	Source Specific Multicast (SSM) Mapping	The Source Specific Multicast (SSM) Mapping feature extends the Cisco IOS suite of SSM transition tools, which also includes URD and IGMP v3lite. SSM mapping supports SSM transition in cases where neither URD nor IGMP v3lite is available, or when supporting SSM on the end system is impossible or unwanted due to administrative or technical reasons. SSM mapping enables you to leverage SSM for video delivery to legacy STBs that do not support IGMPv3 or for applications that do not take advantage of the IGMPv3 host stack.	Source Specific Multicast (SSM) Mapping
12.2(31)SB2	BGP Multicast Inter-AS VPN	The BGP Multicast Inter-AS VPN feature introduces the IPv4 MDT SAFI in BGP. The MDT SAFI is a transitive multicast capable connector attribute that is defined as an IPv4 address family in BGP. The MDT SAFI is designed to support inter-AS VPN peering sessions.	Configuring Multicast VPN Inter-AS Support
12.2(31)SB2	Multicast VPN Extranet Support	The Multicast VPN Extranet Support feature enables service providers to distribute IP multicast content originated from one enterprise site to other enterprise sites. This feature enables service providers to offer the next generation of flexible extranet services, helping to enable business partnerships between different enterprise VPN customers.	Configuring Multicast VPN Extranet Support

Table 1 Supported IP Multicast Features (continued)

Release	Feature Name	Feature Description	Where Documented
12.2(31)SB2	Multicast VPN Extranet VRF Select	The Multicast VPN Extranet VRF Select feature provides the capability for RPF lookups to be performed to the same source address in different VRFs using the group address as the VRF selector. This feature enhances extranet MVPNs by enabling service providers to distribute content streams coming in from different MVPNs and redistributing them from there.	Configuring Multicast VPN Extranet Support
12.2(31)SB2	Multicast VPN Inter-AS Support	The Multicast VPN Inter-AS support feature enables MDTs used for MVPNs to span multiple autonomous systems. Benefits include increased multicast coverage to customers that require multicast to span multiple service providers in an MPLS Layer 3 VPN service with the flexibility to support all options described in RFC 4364. Additionally, the Multicast VPN Inter-AS Support feature may be used to consolidate an existing MVPN service with another MVPN service, such as the case with a company merger or acquisition.	Configuring Multicast VPN Inter-AS Support
12.2(31)SB2	PIM RPF Vector	The PIM RPF Vector feature enables core routers to perform RPF checks on an IP address of the exit router instead of on the source router. The address on the exit router is the RPF Vector and it is inserted in PIM join messages.	Configuring Multicast VPN Inter-AS Support
Cisco IOS Releases 12.2SR			
12.2(33)SRA	BGP Multicast Inter-AS VPN	The BGP Multicast Inter-AS VPN feature introduces the IPv4 MDT SAFI in BGP. The MDT SAFI is a transitive multicast capable connector attribute that is defined as an IPv4 address family in BGP. The MDT SAFI is designed to support inter-AS VPN peering sessions.	Configuring Multicast VPN Inter-AS Support
12.2(33)SRA	Extended ACL Support for IGMP to Support SSM in IPv4	The Extended ACL Support for IGMP to Support SSM in IPv4 feature enables IGMPv3 to accommodate extended access lists. IGMPv3 support of extended access lists allows you to leverage an important advantage of SSM in IPv4, that of filtering IGMPv3 reports based on source address, group address, or both.	Customizing IGMP
12.2(33)SRA	MSDP Compliance with IETF RFC 3618	The MSDP Compliance with IETF RFC 3618 feature enables you to configure MSDP to comply with the peer-RPF forwarding rules defined in the IETF RFC 3618 specifications.	Using MSDP to Interconnect Multiple PIM-SM Domains

Table 1 Supported IP Multicast Features (continued)

Release	Feature Name	Feature Description	Where Documented
12.2(33)SRA	Multicast VPN Inter-AS Support	The Multicast VPN Inter-AS support feature enables MDTs used for MVPNs to span multiple autonomous systems. Benefits include increased multicast coverage to customers that require multicast to span multiple service providers in an MPLS Layer 3 VPN service with the flexibility to support all options described in RFC 4364. Additionally, the Multicast VPN Inter-AS Support feature may be used to consolidate an existing MVPN service with another MVPN service, such as the case with a company merger or acquisition.	Configuring Multicast VPN Inter-AS Support
12.2(33)SRA	Multicast VPN MIB	The Multicast VPN MIB feature introduces the capability for SNMP monitoring of an MVPN using the MVPN MIB (CISCO-MVPN-MIB).	Multicast VPN MIB
12.2(33)SRA	PIM Dense Mode Fallback Prevention in a Network Following RP Information Loss	The PIM Dense Mode Fallback Prevention in a Network Following RP Information Loss feature enables you to prevent PIM-DM fallback when all RPs fail. Preventing the use of dense mode is very important to multicast networks whose reliability is critical. This feature provides a mechanism to keep the multicast groups in sparse mode. This feature also allows you to block multicast traffic for groups not specifically configured.	IP Multicast Technology Overview Configuring Basic IP Multicast
12.2(33)SRA	PIM RPF Vector	The PIM RPF Vector feature enables core routers to perform RPF checks on an IP address of the exit router instead of on the source router. The address on the exit router is the RPF Vector and it is inserted in PIM join messages.	Configuring Multicast VPN Inter-AS Support
12.2(33)SRB	Bandwidth-Based CAC for IP Multicast	The Bandwidth-Based CAC for IP Multicast feature enhances the Per Interface Mroute State Limit feature by implementing a way to count per interface mroute state limiters using cost multipliers. This feature can be used to provide bandwidth-based CAC on a per interface basis in network environments where the multicast flows utilize different amounts of bandwidth.	Per Interface Mroute State Limit with Bandwidth-Based Call Admission Control (CAC) for IP Multicast

Table 1 Supported IP Multicast Features (continued)

Release	Feature Name	Feature Description	Where Documented
12.2(33)SRB	IP Multicast Load Splitting—Equal Cost Multipath (ECMP) Using S, G and Next Hop	The IP Multicast Load Splitting—Equal Cost Multipath (ECMP) Using S, G and Next Hop feature introduces more flexible support for ECMP multicast load splitting by adding support for load splitting based on source and group address and on source, group, and next-hop address. This feature enables multicast traffic from devices that send many streams to groups or that broadcast many channels, such as IPTV servers or MPEG video servers, to be more effectively load split across equal-cost paths. Prior to the introduction of this feature, the Cisco IOS software only supported ECMP multicast load splitting based on source address, which restricted multicast traffic sent by a single source to multiple groups from being load split across equal-cost paths.	Load Splitting IP Multicast Traffic over ECMP
12.2(33)SRB	Per Interface Mroute State Limit	The Per Interface Mroute State Limit feature provides the capability to limit the number of mroute states on an interface for different ACL-classified sets of multicast traffic. This feature can be used to prevent DOS attacks, or to provide a multicast CAC mechanism in network environments where all the multicast flows roughly utilize the same amount of bandwidth.	Per Interface Mroute State Limit with Bandwidth-Based Call Admission Control (CAC) for IP Multicast
12.2(33)SRC	Multicast VPN Extranet Support	The Multicast VPN Extranet Support feature enables service providers to distribute IP multicast content originated from one enterprise site to other enterprise sites. This feature enables service providers to offer the next generation of flexible extranet services, helping to enable business partnerships between different enterprise VPN customers.	Configuring Multicast VPN Extranet Support
Cisco IOS Releases 12.2SX			
12.2(18)SXD 3	Source Specific Multicast (SSM) Mapping	The Source Specific Multicast (SSM) Mapping feature extends the Cisco IOS suite of SSM transition tools, which also includes URD and IGMP v3lite. SSM mapping supports SSM transition in cases where neither URD nor IGMP v3lite is available, or when supporting SSM on the end system is impossible or unwanted due to administrative or technical reasons. SSM mapping enables you to leverage SSM for video delivery to legacy STBs that do not support IGMPv3 or for applications that do not take advantage of the IGMPv3 host stack.	Source Specific Multicast (SSM) Mapping

Table 1 Supported IP Multicast Features (continued)

Release	Feature Name	Feature Description	Where Documented
12.2(18)SXF 5	IGMP Static Group Range Support	The IGMP Static Group Range Support feature introduces the capability to configure group ranges in class maps and attach class maps to the **ip igmp static-group** command. This feature is an enhancement that simplifies the administration of networks with devices that require many interfaces to be configured with many different **ip igmp static-group** command configurations.	IGMP Static Group Range Support
12.2(33)SXH	BGP Multicast Inter-AS VPN	The BGP Multicast Inter-AS VPN feature introduces the IPv4 MDT SAFI in BGP. The MDT SAFI is a transitive multicast capable connector attribute that is defined as an IPv4 address family in BGP. The MDT SAFI is designed to support inter-AS VPN peering sessions.	Configuring Multicast VPN Inter-AS Support
12.2(33)SXH	Extended ACL Support for IGMP to Support SSM in IPv4	The Extended ACL Support for IGMP to Support SSM in IPv4 feature enables IGMPv3 to accommodate extended access lists. IGMPv3 support of extended access lists allows you to leverage an important advantage of SSM in IPv4, that of filtering IGMPv3 reports based on source address, group address, or both.	Customizing IGMP
12.2(33)SXH	Hardware Acceleration for Multicast VPN Extranet Support	The Hardware Acceleration for Multicast VPN Extranet Support introduces the linking of forwarding entries and the replication of packets in hardware for extranet MVPN services on Catalyst 6500 series switches.	Configuring Multicast VPN Extranet Support
12.2(33)SXH	MSDP Compliance with IETF RFC 3618	The MSDP Compliance with IETF RFC 3618 feature enables you to configure MSDP to comply with the peer-RPF forwarding rules defined in the IETF RFC 3618 specifications.	Using MSDP to Interconnect Multiple PIM-SM Domains
12.2(33)SXH	MSDP MD5 Password Authentication	The MSDP MD5 password authentication feature is an enhancement to support Message Digest 5 (MD5) signature protection on a TCP connection between two MSDP peers. This feature provides added security by protecting MSDP against the threat of spoofed TCP segments being introduced into the TCP connection stream.	Using MSDP to Interconnect Multiple PIM-SM Domains

Table 1 Supported IP Multicast Features (continued)

Release	Feature Name	Feature Description	Where Documented
12.2(33)SXH	Multicast VPN Extranet Support	The Multicast VPN Extranet Support feature enables service providers to distribute IP multicast content originated from one enterprise site to other enterprise sites. This feature enables service providers to offer the next generation of flexible extranet services, helping to enable business partnerships between different enterprise VPN customers.	Configuring Multicast VPN Extranet Support
12.2(33)SXH	Multicast VPN Inter-AS Support	The Multicast VPN Inter-AS support feature enables MDTs used for MVPNs to span multiple autonomous systems. Benefits include increased multicast coverage to customers that require multicast to span multiple service providers in an MPLS Layer 3 VPN service with the flexibility to support all options described in RFC 4364. Additionally, the Multicast VPN Inter-AS Support feature may be used to consolidate an existing MVPN service with another MVPN service, such as the case with a company merger or acquisition.	Configuring Multicast VPN Inter-AS Support
12.2(33)SXH	Multicast VPN MIB	The Multicast VPN MIB feature introduces the capability for SNMP monitoring of an MVPN using the MVPN MIB (CISCO-MVPN-MIB).	Multicast VPN MIB
12.2(33)SXH	PIM Dense Mode Fallback Prevention in a Network Following RP Information Loss	The PIM Dense Mode Fallback Prevention in a Network Following RP Information Loss feature enables you to prevent PIM-DM fallback when all RPs fail. Preventing the use of dense mode is very important to multicast networks whose reliability is critical. This feature provides a mechanism to keep the multicast groups in sparse mode. This feature also allows you to block multicast traffic for groups not specifically configured.	IP Multicast Technology Overview Configuring Basic IP Multicast
12.2(33)SXH	PIM RPF Vector	The PIM RPF Vector feature enables core routers to perform RPF checks on an IP address of the exit router instead of on the source router. The address on the exit router is the RPF Vector and it is inserted in PIM join messages.	Configuring Multicast VPN Inter-AS Support

Table 1 Supported IP Multicast Features (continued)

Release	Feature Name	Feature Description	Where Documented
12.2(33)SXH	PIM Triggered Joins	The PIM Triggered Joins feature is a high availability (HA) multicast enhancement that improves the reconvergence of multicast routes (mroutes) after a supervisor engine switchover on a Catalyst 6500 series switch. In the event of a supervisor engine switchover, this feature utilizes the Generation ID (GenID) value as a mechanism to trigger adjacent Protocol Independent Multicast (PIM) neighbors on an interface to send PIM join messages for all (*, G) and (S, G) mroutes that use that interface as a reverse path forwarding (RPF) interface, immediately reestablishing those states on the newly active supervisor engine.	PIM Triggered Joins
Cisco IOS Releases 12.2T, 12.3, 12.3T, 12.4, and 12.4T			
12.2(4)T	PIM MIB Extensions	Protocol Independent Multicast (PIM) is an IP multicast routing protocol used for routing multicast data packets to multicast groups. RFC 2934 defines the PIM for IPv4 MIB, which describes managed objects that enable users to remotely monitor and configure PIM using Simple Network Management Protocol (SNMP).	Monitoring and Maintaining IP Multicast
12.2(8)T	IGMPv3—Explicit Tracking of Hosts, Groups, and Channels	This IGMPv3—Explicit Tracking Host, Group, and Channel feature enables a multicast router to explicitly track the membership of all multicast hosts in a particular multiaccess network. This enhancement to the Cisco IOS implementation of IGMPv3 enables the router to track each individual host that is joined to a particular group or channel.	Customizing IGMP
12.2(13)T	Multicast-VPN—IP Multicast Support of MPLS VPNs	The Multicast VPN feature provides the ability to support multicast over a Layer 3 Virtual Private Network (VPN). As enterprises extend the reach of their multicast applications, service providers can accommodate these enterprises over their Multiprotocol Label Switching (MPLS) core network. IP multicast is used to stream video, voice, and data to an MPLS VPN network core.	Configuring Multicast VPN
12.2(15)T	IGMP State Limit	The IGMP State Limit feature introduces the capability to limit the number of mroute states resulting from IGMP membership states per interface, per subinterface, or globally.	Customizing IGMP

Table 1 Supported IP Multicast Features (continued)

Release	Feature Name	Feature Description	Where Documented
12.2(15)T	Multicast Subsecond Convergence	The Multicast Subsecond Convergence feature comprises a comprehensive set of features and protocol enhancements that provide for improved scalability and convergence in multicast-based services. This feature set provides for the ability to scale to larger services levels and to recover multicast forwarding after service failure in subsecond time frames.	Multicast Subsecond Convergence
12.3(2)T	Source Specific Multicast (SSM) Mapping	The Source Specific Multicast (SSM) Mapping feature extends the Cisco IOS suite of SSM transition tools, which also includes URD and IGMP v3lite. SSM mapping supports SSM transition in cases where neither URD nor IGMP v3lite is available, or when supporting SSM on the end system is impossible or unwanted due to administrative or technical reasons. SSM mapping enables you to leverage SSM for video delivery to legacy STBs that do not support IGMPv3 or for applications that do not take advantage of the IGMPv3 host stack.	Source Specific Multicast (SSM) Mapping
12.3(4)T	MSDP Compliance with IETF RFC 3618	The MSDP Compliance with IETF RFC 3618 feature enables you to configure MSDP to comply with the peer-RPF forwarding rules defined in the IETF RFC 3618 specifications.	Using MSDP to Interconnect Multiple PIM-SM Domains
12.3(4)T	PIM Dense Mode Fallback Prevention in a Network Following RP Information Loss	The PIM Dense Mode Fallback Prevention in a Network Following RP Information Loss feature enables you to prevent PIM-DM fallback when all RPs fail. Preventing the use of dense mode is very important to multicast networks whose reliability is critical. This feature provides a mechanism to keep the multicast groups in sparse mode. This feature also allows you to block multicast traffic for groups not specifically configured.	IP Multicast Technology Overview Configuring Basic IP Multicast
12.3(7)T	Extended ACL Support for IGMP to Support SSM in IPv4	The Extended ACL Support for IGMP to Support SSM in IPv4 feature enables IGMPv3 to accommodate extended access lists. IGMPv3 support of extended access lists allows you to leverage an important advantage of SSM in IPv4, that of filtering IGMPv3 reports based on source address, group address, or both.	Customizing IGMP
12.3(11)T	SSM Channel-Based Filtering for Multicast Boundaries	The SSM Channel Based Filtering for Multicast Boundaries feature enables the application of SSM filtering policies based on SSM channels.	SSM Channel-Based Filtering for Multicast Boundaries

Table 1 Supported IP Multicast Features (continued)

Release	Feature Name	Feature Description	Where Documented
12.3(14)T	Multicast VPN MIB	The Multicast VPN MIB feature introduces the capability for SNMP monitoring of an MVPN using the MVPN MIB (CISCO-MVPN-MIB).	Multicast VPN MIB
12.3(14)T	Per Interface Mroute State Limit	The Per Interface Mroute State Limit feature provides the capability to limit the amount of multicast route (mroute) states on an interface for different access control list (ACL)-classified sets of multicast traffic. This feature can be used to prevent denial-of-service (DoS) attacks, or to provide a multicast Call Admission Control (CAC) mechanism in network environments where all the multicast flows roughly utilize the same amount of bandwidth.	Per Interface Mroute State Limit with Bandwidth-Based Call Admission Control (CAC) for IP Multicast
12.3(14)T	IGMPv3 Host Stack	The IGMPv3 Host Stack feature enables routers and switches to function as multicast network endpoints or hosts. The feature adds INCLUDE mode capability to the IGMPv3 host stack for SSM groups. Enabling the IGMPv3 host stack ensures that hosts on a LAN can leverage SSM by enabling the router to initiate IGMPv3 joins, such as in environments where fast channel change is required in a SSM deployments.	Customizing IGMP

Table 1 Supported IP Multicast Features (continued)

Release	Feature Name	Feature Description	Where Documented
12.4(2)T	MSDP MD5 Password Authentication	The MSDP MD5 password authentication feature is an enhancement to support MD5 signature protection on a TCP connection between two MSDP peers. This feature provides added security by protecting MSDP against the threat of spoofed TCP segments being introduced into the TCP connection stream.	Using MSDP to Interconnect Multiple PIM-SM Domains
12.4(4)T	Multicast Service Reflection	The Multicast Service Reflection feature provides the capability for users to translate externally received multicast destination addresses to addresses that conform to their organization's internal addressing policy. Using this feature, users do not need to redistribute routes at the translation boundary into their network infrastructure for Reverse Path Forwarding (RPF) to work properly, and users can receive identical feeds from two ingress points in the network and route them independently.	Implementing Multicast Service Reflection
12.4(15)T	Bandwidth-Based CAC for IP Multicast	The Bandwidth-Based CAC for IP Multicast feature enhances the Per Interface Mroute State Limit feature by implementing a way to count per interface mroute state limiters using cost multipliers. This feature can be used to provide bandwidth-based CAC on a per interface basis in network environments where the multicast flows utilize different amounts of bandwidth.	Per Interface Mroute State Limit with Bandwidth-Based Call Admission Control (CAC) for IP Multicast

IP Multicast Technology Overview

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

This module contains a technical overview of IP multicast. IP multicast is an efficient way to use network resources, especially for bandwidth-intensive services such as audio and video. Before beginning to configure IP multicast, it is important that you understand the information presented in this module.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for IP Multicast Technology Overview" section on page 24.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Information About IP Multicast Technology

Before you configure IP multicast, you should understand the following concepts:

Role of IP Multicast in Information Delivery

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

IP multicast routing enables a host (source) to send packets to a group of hosts (receivers) anywhere within the IP network by using a special form of IP address called the IP multicast group address. The sending host inserts the multicast group address into the IP destination address field of the packet and IP multicast routers and multilayer switches forward incoming IP multicast packets out all interfaces that lead to the members of the multicast group. 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.

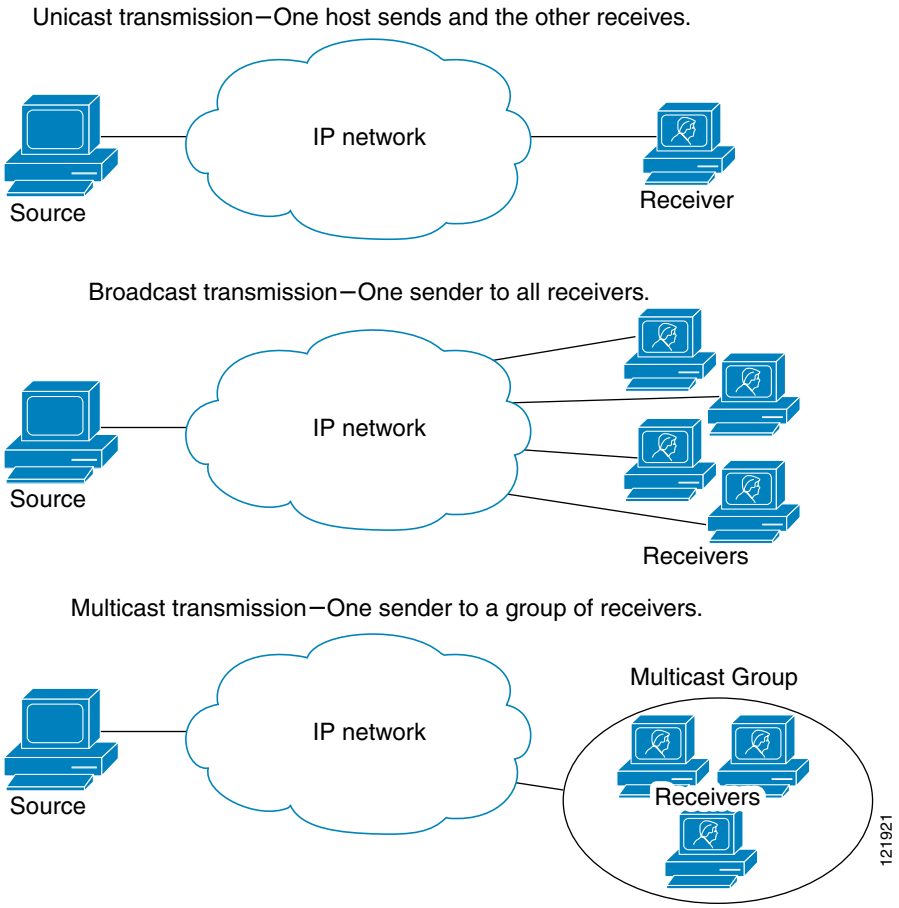
Multicast Group Transmission Scheme

IP communication consists of hosts that act as senders and receivers of traffic as shown in Figure 1. Senders are called sources. Traditional IP communication is accomplished by a single host source sending packets to another single host (*unicast transmission*) or to all hosts (*broadcast transmission*). IP multicast provides a third scheme, allowing a host to send packets to a subset of all hosts (*multicast transmission*). This subset of receiving hosts is called a multicast group. The hosts that belong to a multicast group are called group members.

Multicast is based on this group concept. A multicast group is an arbitrary number of receivers that join a group in order to receive a particular data stream. This multicast group has no physical or geographical boundaries—the hosts can be located anywhere on the Internet or on any private internetwork. Hosts that are interested in receiving data from a source to a particular group must join that group. Joining a group is accomplished by a host receiver by way of the Internet Group Management Protocol (IGMP).

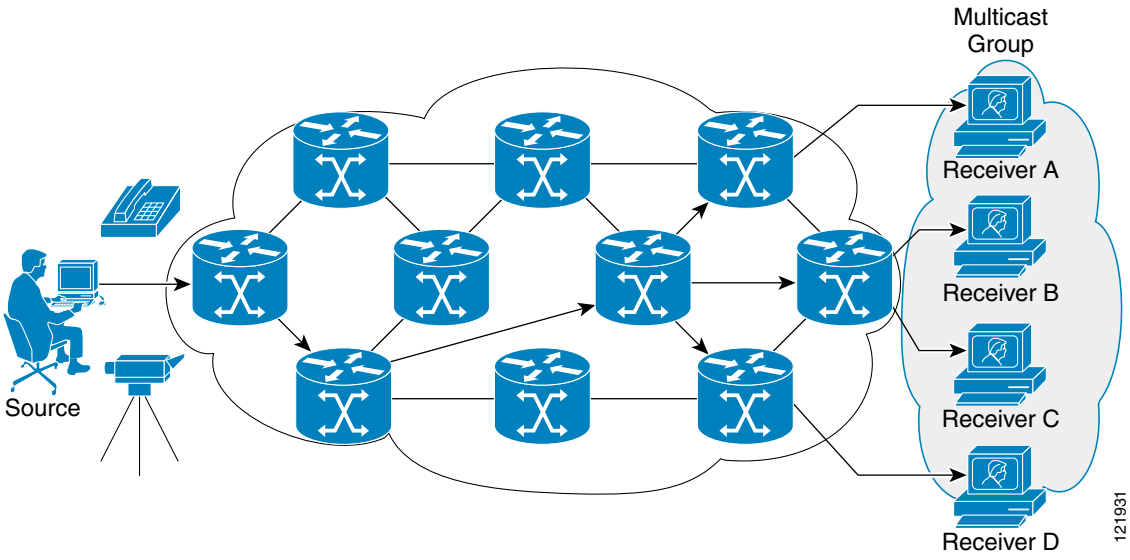
In a multicast environment, any host, regardless of whether it is a member of a group, can send to a group. However, only the members of a group can receive packets sent to that group. Multicast packets are delivered to a group using best-effort reliability, just like IP unicast packets.

Figure 1 IP Transmission Schemes



In Figure 2, the receivers (the designated multicast group) are interested in receiving the video data stream from the source. The receivers indicate their interest by sending an IGMP host report to the routers in the network. The routers are then responsible for delivering the data from the source to the receivers. The routers use Protocol Independent Multicast (PIM) (see the "Protocol Independent Multicast" section) to dynamically create a multicast distribution tree. The video data stream will then be delivered only to the network segments that are in the path between the source and the receivers.

Figure 2 Multicast Transmission



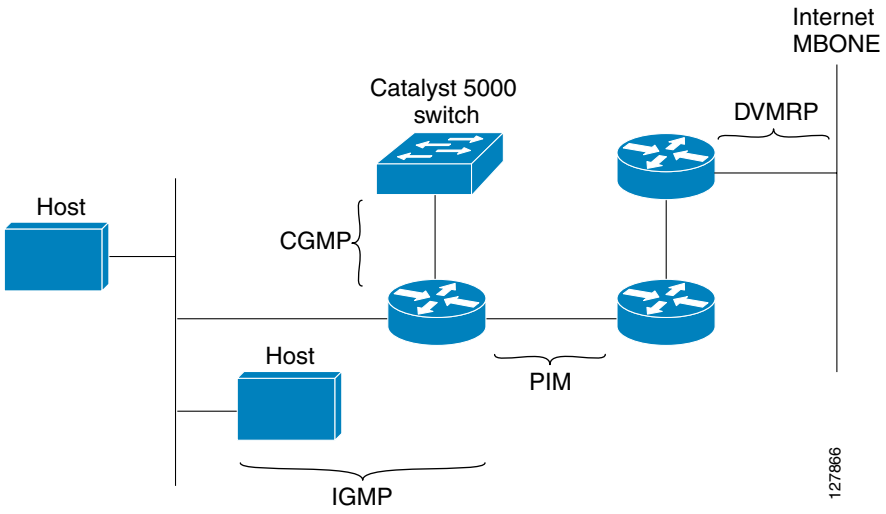
IP Multicast Routing Protocols

The Cisco IOS software supports the following protocols to implement IP 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 (PIM) is used between routers so that they can track which multicast packets to forward to each other and to their directly connected LANs.
- Distance Vector Multicast Routing Protocol (DVMRP) is used on the MBONE (the multicast backbone of the Internet). The Cisco IOS software supports PIM-to-DVMRP interaction.
- Cisco Group Management Protocol (CGMP) is used on routers connected to Catalyst switches to perform tasks similar to those performed by IGMP.

Figure 3 shows where these protocols operate within the IP multicast environment.

Figure 3 IP Multicast Routing Protocols



IP Multicast Group Addressing

A multicast group is identified by its multicast group address. Multicast packets are delivered to that multicast group address. Unlike unicast addresses that uniquely identify a single host, multicast IP addresses do not identify a particular host. To receive the data sent to a multicast address, a host must join the group that address identifies. The data is sent to the multicast address and received by all the hosts that have joined the group indicating that they wish to receive traffic sent to that group. The multicast group address is assigned to a group at the source. Network administrators who assign multicast group addresses must make sure the addresses conform to the multicast address range assignments reserved by the Internet Assigned Numbers Authority (IANA).

IP Class D Addresses

IP multicast addresses have been assigned to the IPv4 Class D address space by IANA. The high-order four bits of a Class D address are 1110. Therefore, host group addresses can be in the range 224.0.0.0 to 239.255.255.255. A multicast address is chosen at the source (sender) for the receivers in a multicast group.

Note The Class D address range is used only for the group address or destination address of IP multicast traffic. The source address for multicast datagrams is always the unicast source address.

IP Multicast Address Scoping

The multicast address range is subdivided to provide predictable behavior for various address ranges and for address reuse within smaller domains. Table 1 is a summary of the multicast address ranges. A brief summary description of each range follows.

Table 1 Multicast Address Range Assignments

Name	Range	Description
Reserved Link-Local Addresses	224.0.0.0 to 224.0.0.255	Reserved for use by network protocols on a local network segment.
Globally Scoped Addresses	224.0.1.0 to 238.255.255.255	Reserved to send multicast data between organizations and across the Internet.
Source Specific Multicast	232.0.0.0 to 232.255.255.255	Reserved for use with the SSM datagram delivery model where data is forwarded only to receivers that have explicitly joined the group.
GLOP Addresses	233.0.0.0 to 233.255.255.255	Reserved for statically defined addresses by organizations that already have an assigned autonomous system (AS) domain number.
Limited Scope Address	239.0.0.0 to 239.255.255.255	Reserved as administratively or limited scope addresses for use in private multicast domains.

Reserved Link-Local Addresses

The IANA has reserved the range 224.0.0.0 to 224.0.0.255 for use by network protocols on a local network segment. Packets with an address in this range are local in scope and are not forwarded by IP routers. Packets with link local destination addresses are typically sent with a time-to-live (TTL) value of 1 and are not forwarded by a router.

Within this range, reserved link-local addresses provide network protocol functions for which they are reserved. Network protocols use these addresses for automatic router discovery and to communicate important routing information. For example, Open Shortest Path First (OSPF) uses the IP addresses 224.0.0.5 and 224.0.0.6 to exchange link-state information.

IANA assigns single multicast address requests for network protocols or network applications out of the 224.0.1.*xxx* address range. Multicast routers forward these multicast addresses.

Globally Scoped Addresses

Addresses in the range 224.0.1.0 to 238.255.255.255 are called globally scoped addresses. These addresses are used to send multicast data between organizations across the Internet. Some of these addresses have been reserved by IANA for use by multicast applications. For example, the IP address 224.0.1.1 is reserved for Network Time Protocol (NTP).

Source Specific Multicast Addresses

Addresses in the range 232.0.0.0/8 are reserved for Source Specific Multicast (SSM) by IANA. In Cisco IOS software, you can use the **ip pim ssm** command to configure SSM for arbitrary IP multicast addresses also. SSM is an extension of Protocol Independent Multicast (PIM) that allows for an efficient data delivery mechanism in one-to-many communications. SSM is described in the "IP Multicast Delivery Modes" section.

GLOP Addresses

GLOP addressing (as proposed by RFC 2770, *GLOP Addressing in 233/8*) proposes that the 233.0.0.0/8 range be reserved for statically defined addresses by organizations that already have an AS number reserved. This practice is called GLOP addressing. The AS number of the domain is embedded into the second and third octets of the 233.0.0.0/8 address range. For example, AS 62010 is written in hexadecimal format as F23A. Separating the two octets F2 and 3A results in 242 and 58 in decimal format. These values result in a subnet of 233.242.58.0/24 that would be globally reserved for AS 62010 to use.

Limited Scope Addresses

The range 239.0.0.0 to 239.255.255.255 is reserved as administratively or limited scoped addresses for use in private multicast domains. These addresses are constrained to a local group or organization. Companies, universities, and other organizations can use limited scope addresses to have local multicast applications that will not be forwarded outside their domain. Routers typically are configured with filters to prevent multicast traffic in this address range from flowing outside an autonomous system (AS) or any user-defined domain. Within an AS or domain, the limited scope address range can be further subdivided so that local multicast boundaries can be defined.

Note Network administrators may use multicast addresses in this range, inside a domain, without conflicting with others elsewhere in the Internet.

Layer 2 Multicast Addresses

Historically, network interface cards (NICs) on a LAN segment could receive only packets destined for their burned-in MAC address or the broadcast MAC address. In IP multicast, several hosts need to be able to receive a single data stream with a common destination MAC address. Some means had to be devised so that multiple hosts could receive the same packet and still be able to differentiate between several multicast groups. One method to accomplish this is to map IP multicast Class D addresses directly to a MAC address. Using this method, NICs can receive packets destined to many different MAC address.

Cisco Group Management Protocol (CGMP) is used on routers connected to Catalyst switches to perform tasks similar to those performed by IGMP. CGMP is necessary for those Catalyst switches that cannot distinguish between IP multicast data packets and IGMP report messages, both of which are addressed to the same group address at the MAC level.

IP Multicast Delivery Modes

IP multicast delivery modes differ only for the receiver hosts, not for the source hosts. A source host sends IP multicast packets with its own IP address as the IP source address of the packet and a group address as the IP destination address of the packet.

Any Source Multicast

For the Any Source Multicast (ASM) delivery mode, an IP multicast receiver host can use any version of IGMP to join a multicast group. This group is notated as G in the routing table state notation. By joining this group, the receiver host is indicating that it wants to receive IP multicast traffic sent by any source to group G. The network will deliver IP multicast packets from any source host with the destination address G to all receiver hosts in the network that have joined group G.

ASM requires group address allocation within the network. At any given time, an ASM group should only be used by a single application. When two applications use the same ASM group simultaneously, receiver hosts of both applications will receive traffic from both application sources. This may result in unexpected excess traffic in the network. This situation may cause congestion of network links and malfunction of the application receiver hosts.

Source Specific Multicast

Source Specific Multicast (SSM) is a datagram delivery model that best supports one-to-many applications, also known as broadcast applications. SSM is a core network technology for the Cisco implementation of IP multicast targeted for audio and video broadcast application environments.

For the SSM delivery mode, an IP multicast receiver host must use IGMP Version 3 (IGMPv3) to subscribe to channel (S,G). By subscribing to this channel, the receiver host is indicating that it wants to receive IP multicast traffic sent by source host S to group G. The network will deliver IP multicast packets from source host S to group G to all hosts in the network that have subscribed to the channel (S, G).

SSM does not require group address allocation within the network, only within each source host. Different applications running on the same source host must use different SSM groups. Different applications running on different source hosts can arbitrarily reuse SSM group addresses without causing any excess traffic on the network.

Protocol Independent Multicast

The Protocol Independent Multicast (PIM) protocol maintains the current IP multicast service mode of receiver-initiated membership. PIM is not dependent on a specific unicast routing protocol; it is IP routing protocol independent and can leverage whichever unicast routing protocols are used to populate the unicast routing table, including Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First (OSPF), Border Gateway Protocol (BGP), and static routes. PIM uses unicast routing information to perform the multicast forwarding function.

Although PIM is called a multicast routing protocol, it actually uses the unicast routing table to perform the reverse path forwarding (RPF) check function instead of building up a completely independent multicast routing table. Unlike other routing protocols, PIM does not send and receive routing updates between routers.

PIM is defined in RFC 2362, *Protocol-Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification*.

PIM can operate in dense mode or sparse mode. The router can handle both sparse groups and dense groups at the same time. The mode determines how the router populates its multicast routing table and how the router forwards multicast packets it receives from its directly connected LANs.

For information about PIM forwarding (interface) modes, see the following sections:

- "PIM Dense Mode" section on page 8
- "PIM Sparse Mode" section on page 9
- "Sparse-Dense Mode" section on page 10
- "Bidirectional PIM" section on page 10

PIM Dense Mode

PIM dense mode (PIM-DM) uses a push model to flood multicast traffic to every corner of the network. This push model is a method for delivering data to the receivers without the receivers requesting the data. This method is efficient in certain deployments in which there are active receivers on every subnet in the network.

In dense mode, a router assumes that all other routers want to forward multicast packets for a group. If a router receives a multicast packet and has no directly connected members or PIM neighbors present, a prune message is sent back to the source. Subsequent multicast packets are not flooded to this router on this pruned branch. PIM builds source-based multicast distribution trees.

PIM-DM initially floods multicast traffic throughout the network. Routers that have no downstream neighbors prune back the unwanted traffic. This process repeats every 3 minutes.

Routers accumulate state information by receiving data streams through the flood and prune mechanism. These data streams contain the source and group information so that downstream routers can build up their multicast forwarding table. PIM-DM supports only source trees—that is, (S,G) entries—and cannot be used to build a shared distribution tree.

Note Dense mode is not often used and its use is not recommended. For this reason it is not specified in the configuration tasks in related modules.

PIM Sparse Mode

PIM sparse mode (PIM-SM) uses a pull model to deliver multicast traffic. Only network segments with active receivers that have explicitly requested the data will receive the traffic.

Unlike dense mode interfaces, sparse mode interfaces are added to the multicast routing table only when periodic Join messages are received from downstream routers, or when a directly connected member is on the interface. When forwarding from a LAN, sparse mode operation occurs if an RP is known for the group. If so, the packets are encapsulated and sent toward the RP. When no RP is known, the packet is flooded in a dense mode fashion. If the multicast traffic from a specific source is sufficient, the first hop router of the receiver may send Join messages toward the source to build a source-based distribution tree.

PIM-SM distributes information about active sources by forwarding data packets on the shared tree. Because PIM-SM uses shared trees (at least, initially), it requires the use of a rendezvous point (RP). The RP must be administratively configured in the network. See the "Rendezvous Points" section for more information.

In sparse mode, a router assumes that other routers do not want to forward multicast packets for a group, unless there is an explicit request for the traffic. When hosts join a multicast group, the directly connected routers send PIM Join messages toward the RP. The RP keeps track of multicast groups. Hosts that send multicast packets are registered with the RP by the first hop router of that host. The RP then sends Join messages toward the source. At this point, packets are forwarded on a shared distribution tree. If the multicast traffic from a specific source is sufficient, the first hop router of the host may send Join messages toward the source to build a source-based distribution tree.

Sources register with the RP and then data is forwarded down the shared tree to the receivers. The edge routers learn about a particular source when they receive data packets on the shared tree from that source through the RP. The edge router then sends PIM (S,G) Join messages toward that source. Each router along the reverse path compares the unicast routing metric of the RP address to the metric of the source address. If the metric for the source address is better, it will forward a PIM (S,G) Join message toward the source. If the metric for the RP is the same or better, then the PIM (S,G) Join message will be sent in the same direction as the RP. In this case, the shared tree and the source tree would be considered congruent.

If the shared tree is not an optimal path between the source and the receiver, the routers dynamically create a source tree and stop traffic from flowing down the shared tree. This behavior is the default behavior in Cisco IOS software. Network administrators can force traffic to stay on the shared tree by using the Cisco IOS **ip pim spt-threshold infinity** command.

PIM-SM scales well to a network of any size, including those with WAN links. The explicit join mechanism prevents unwanted traffic from flooding the WAN links.

Sparse-Dense Mode

If you configure either sparse mode or dense mode on an interface, then sparseness or denseness is applied to the interface as a whole. However, some environments might require PIM to run in a single region in sparse mode for some groups and in dense mode for other groups.

An alternative to enabling only dense mode or only sparse mode is to enable sparse-dense mode. In this case, the interface is treated as dense mode if the group is in dense mode; the interface is treated in sparse mode if the group is in sparse mode. You must have an RP if the interface is in sparse-dense mode and you want to treat the group as a sparse group.

If you configure sparse-dense mode, the idea of sparseness or denseness is applied to the groups for which the router is a member.

Another benefit of sparse-dense mode is that Auto-RP information can be distributed in a dense mode; yet, multicast groups for user groups can be used in a sparse mode manner. Therefore there is no need to configure a default RP at the leaf routers.

When an interface is treated in dense mode, it is populated in the outgoing interface list of a multicast routing table when either of the following conditions is true:

- Members or DVMRP neighbors are on the interface.
- There are PIM neighbors and the group has not been pruned.

When an interface is treated in sparse mode, it is populated in the outgoing interface list of a multicast routing table when either of the following conditions is true:

- Members or DVMRP neighbors are on the interface.
- An explicit Join message has been received by a PIM neighbor on the interface.

Bidirectional PIM

Bidirectional PIM (bidir-PIM) is an enhancement of the PIM protocol that was designed for efficient many-to-many communications within an individual PIM domain. Multicast groups in bidirectional mode can scale to an arbitrary number of sources with only a minimal amount of additional overhead.

The shared trees that are created in PIM sparse mode are unidirectional. This means that a source tree must be created to bring the data stream to the RP (the root of the shared tree) and then it can be forwarded down the branches to the receivers. Source data cannot flow up the shared tree toward the RP—this would be considered a bidirectional shared tree.

In bidirectional mode, traffic is routed only along a bidirectional shared tree that is rooted at the RP for the group. In bidir-PIM, 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 need not be a router address, but can be any unassigned IP address on a network that is reachable throughout the PIM domain.

Bidir-PIM is derived from the mechanisms of PIM sparse mode (PIM-SM) and shares many of the shared tree operations. Bidir-PIM 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.

Multicast Group Modes

In PIM, packet traffic for a multicast group is routed according to the rules of the mode configured for that multicast group. The Cisco IOS implementation of PIM supports four modes for a multicast group:

- PIM Bidirectional mode
- PIM Sparse mode
- PIM Dense mode
- PIM Source Specific Multicast (SSM) mode

A router can simultaneously support all four modes or any combination of them for different multicast groups.

Bidirectional Mode

In bidirectional mode, traffic is routed only along a bidirectional shared tree that is rooted at the rendezvous point (RP) for the group. In bidir-PIM, 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 need not be a router, but can be any unassigned IP address on a network that is reachable throughout the PIM domain. This technique is the preferred configuration method for establishing a redundant RP configuration for bidir-PIM.

Membership to a bidirectional group is signalled via explicit Join messages. Traffic from sources is unconditionally sent up the shared tree toward the RP and passed down the tree toward the receivers on each branch of the tree.

Sparse Mode

Sparse mode operation centers around a single unidirectional shared tree whose root node is called the rendezvous point (RP). Sources must register with the RP to get their multicast traffic to flow down the shared tree by way of the RP. This registration process actually triggers a shortest path tree (SPT) Join by the RP toward the source when there are active receivers for the group in the network.

A sparse mode group uses the explicit join model of interaction. Receiver hosts join a group at a rendezvous point (RP). Different groups can have different RPs.

Multicast traffic packets flow down the shared tree to only those receivers that have explicitly asked to receive the traffic.

Dense Mode

Dense mode operates using the broadcast (flood) and prune model.

In populating the multicast routing table, dense mode interfaces are always added to the table. Multicast traffic is forwarded out all interfaces in the outgoing interface list to all receivers. Interfaces are removed from the outgoing interface list in a process called pruning. In dense mode, interfaces are pruned for various reasons including that there are no directly connected receivers.

A pruned interface can be reestablished, that is, grafted back so that restarting the flow of multicast traffic can be accomplished with minimal delay.

Rendezvous Points

A rendezvous point (RP) is a role that a router performs when operating in PIM-SM mode. An RP is required only in networks running PIM-SM. In PIM-SM, only network segments with active receivers that have explicitly requested multicast data will be forwarded the traffic. This method of delivering multicast data is in contrast to the PIM dense mode (PIM-DM) model. In PIM-DM, multicast traffic is initially flooded to all segments of the network. Routers that have no downstream neighbors or directly connected receivers prune back the unwanted traffic.

An RP acts as the meeting place for sources and receivers of multicast data. In a PIM-SM network, sources must send their traffic to the RP. This traffic is then forwarded to receivers down a shared distribution tree. By default, when the first hop router of the receiver learns about the source, it will send a Join message directly to the source, creating a source-based distribution tree from the source to the receiver. This source tree does not include the RP unless the RP is located within the shortest path between the source and receiver.

In most cases, the placement of the RP in the network is not a complex decision. By default, the RP is needed only to start new sessions with sources and receivers. Consequently, the RP experiences little overhead from traffic flow or processing. In PIM version 2, the RP performs less processing than in PIM version 1 because sources must only periodically register with the RP to create state.

Auto-RP

In the first version of PIM-SM, all leaf routers (routers directly connected to sources or receivers) were required to be manually configured with the IP address of the RP. This type of configuration is also known as static RP configuration. Configuring static RPs is relatively easy in a small network, but it can be laborious in a large, complex network.

Following the introduction of PIM-SM version 1, Cisco implemented a version of PIM-SM with the Auto-RP feature. Auto-RP automates the distribution of group-to-RP mappings in a PIM network. Auto-RP has the following benefits:

- Configuring the use of multiple RPs within a network to serve different groups is easy.
- Auto-RP allows load splitting among different RPs and arrangement of RPs according to the location of group participants.
- Auto-RP avoids inconsistent, manual RP configurations that can cause connectivity problems.

Multiple RPs can be used to serve different group ranges or serve as backups to each other. For Auto-RP to work, a router must be designated as an RP-mapping agent, which receives the RP-announcement messages from the RPs and arbitrates conflicts. The RP-mapping agent then sends the consistent group-to-RP mappings to all other routers. Thus, all routers automatically discover which RP to use for the groups they support.

Note If you configure PIM in sparse mode or sparse-dense mode and do not configure Auto-RP, you must statically configure an RP.

Note If 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.

To make Auto-RP work, a router must be designated as an RP mapping agent, which receives the RP announcement messages from the RPs and arbitrates conflicts. The RP mapping agent then sends the consistent group-to-RP mappings to all other routers by dense mode flooding. Thus, all routers automatically discover which RP to use for the groups they support. The Internet Assigned Numbers Authority (IANA) has assigned two group addresses, 224.0.1.39 and 224.0.1.40, for Auto-RP. One advantage of Auto-RP is that any change to the RP designation must be configured only on the routers that are RPs and not on the leaf routers. Another advantage of Auto-RP is that it offers the ability to scope the RP address within a domain. Scoping can be achieved by defining the time-to-live (TTL) value allowed for the Auto-RP advertisements.

Each method for configuring an RP has its own strengths, weaknesses, and level of complexity. In conventional IP multicast network scenarios, we recommend using Auto-RP to configure RPs because it is easy to configure, well-tested, and stable. The alternative ways to configure an RP are static RP, Auto-RP, and bootstrap router.

Sparse-Dense Mode for Auto-RP

A prerequisite of Auto-RP is that all interfaces must be configured in sparse-dense mode using the **ip pim sparse-dense-mode** interface configuration command. An interface configured in sparse-dense mode is treated in either sparse mode or dense mode of operation, depending on which mode the multicast group operates. If a multicast group has a known RP, the interface is treated in sparse mode. If a group has no known RP, by default the interface is treated in dense mode and data will be flooded over this interface. (You can prevent dense-mode fallback; see the module "Configuring Basic IP Multicast.")

To successfully implement Auto-RP and prevent any groups other than 224.0.1.39 and 224.0.1.40 from operating in dense mode, we recommend configuring a "sink RP" (also known as "RP of last resort"). A sink RP is a statically configured RP that may or may not actually exist in the network. Configuring a sink RP does not interfere with Auto-RP operation because, by default, Auto-RP messages supersede static RP configurations. We recommend configuring a sink RP for all possible multicast groups in your network, because it is possible for an unknown or unexpected source to become active. If no RP is configured to limit source registration, the group may revert to dense mode operation and be flooded with data.

Bootstrap Router

Another RP selection model called bootstrap router (BSR) was introduced after Auto-RP in PIM-SM version 2. BSR performs similarly to Auto-RP in that it uses candidate routers for the RP function and for relaying the RP information for a group. RP information is distributed through BSR messages, which are carried within PIM messages. PIM messages are link-local multicast messages that travel from PIM router to PIM router. Because of this single hop method of disseminating RP information, TTL scoping cannot be used with BSR. A BSR performs similarly as an RP, except that it does not run the risk of reverting to dense mode operation, and it does not offer the ability to scope within a domain.

Multicast Source Discovery Protocol

In the PIM sparse mode model, multicast sources and receivers must register with their local rendezvous point (RP). Actually, the router closest to a source or a receiver registers with the RP, but the key point to note is that the RP "knows" about all the sources and receivers for any particular group. RPs in other domains have no way of knowing about sources that are located in other domains. Multicast Source Discovery Protocol (MSDP) is an elegant way to solve this problem.

MSDP is a mechanism that allows RPs to share information about active sources. RPs know about the receivers in their local domain. When RPs in remote domains hear about the active sources, they can pass on that information to their local receivers. Multicast data can then be forwarded between the domains. A useful feature of MSDP is that it allows each domain to maintain an independent RP that does not rely on other domains, but it does enable RPs to forward traffic between domains. PIM-SM is used to forward the traffic between the multicast domains.

The RP in each domain establishes an MSDP peering session using a TCP connection with the RPs in other domains or with border routers leading to the other domains. When the RP learns about a new multicast source within its own domain (through the normal PIM register mechanism), the RP encapsulates the first data packet in a Source-Active (SA) message and sends the SA to all MSDP peers. Each receiving peer uses a modified Reverse Path Forwarding (RPF) check to forward the SA, until the SA reaches every MSDP router in the interconnected networks—theoretically the entire multicast internet. If the receiving MSDP peer is an RP, and the RP has a (*, G) entry for the group in the SA (there is an interested receiver), the RP creates (S,G) state for the source and joins to the shortest path tree for the source. The encapsulated data is decapsulated and forwarded down the shared tree of that RP. When the last hop router (the router closest to the receiver) receives the multicast packet, it may join the shortest path tree to the source. The MSDP speaker periodically sends SAs that include all sources within the domain of the RP.

MSDP was developed for peering between Internet service providers (ISPs). ISPs did not want to rely on an RP maintained by a competing ISP to provide service to their customers. MSDP allows each ISP to have its own local RP and still forward and receive multicast traffic to the Internet.

Anycast RP

Anycast RP is a useful application of MSDP. Originally developed for interdomain multicast applications, MSDP used for Anycast RP is an intradomain feature that provides redundancy and load-sharing capabilities. Enterprise customers typically use Anycast RP for configuring a Protocol Independent Multicast sparse mode (PIM-SM) network to meet fault tolerance requirements within a single multicast domain.

In Anycast RP, two or more RPs are configured with the same IP address on loopback interfaces. The Anycast RP loopback address should be configured with a 32-bit mask, making it a host address. All the downstream routers should be configured to “know” that the Anycast RP loopback address is the IP address of their local RP. IP routing automatically will select the topologically closest RP for each source and receiver. Assuming that the sources are evenly spaced around the network, an equal number of sources will register with each RP. That is, the process of registering the sources will be shared equally by all the RPs in the network.

Because a source may register with one RP and receivers may join to a different RP, a method is needed for the RPs to exchange information about active sources. This information exchange is done with MSDP.

In Anycast RP, all the RPs are configured to be MSDP peers of each other. When a source registers with one RP, an SA message will be sent to the other RPs informing them that there is an active source for a particular multicast group. The result is that each RP will know about the active sources in the area of the other RPs. If any of the RPs were to fail, IP routing would converge and one of the RPs would become the active RP in more than one area. New sources would register with the backup RP. Receivers would join toward the new RP and connectivity would be maintained.

Note The RP is normally needed only to start new sessions with sources and receivers. The RP facilitates the shared tree so that sources and receivers can directly establish a multicast data flow. If a multicast data flow is already directly established between a source and the receiver, then an RP failure will not affect that session. Anycast RP ensures that new sessions with sources and receivers can begin at any time.

Multicast Forwarding

Forwarding of multicast traffic is accomplished by multicast-capable routers. These routers create distribution trees that control the path that IP multicast traffic takes through the network in order to deliver traffic to all receivers.

Multicast traffic flows from the source to the multicast group over a distribution tree that connects all of the sources to all of the receivers in the group. This tree may be shared by all sources (a shared tree) or a separate distribution tree can be built for each source (a source tree). The shared tree may be one-way or bidirectional.

Before describing the structure of source and shared trees, it is helpful to explain the notations that are used in multicast routing tables. These notations include the following:

- (S,G) = (unicast source for the multicast group G, multicast group G)
- (*,G) = (any source for the multicast group G, multicast group G)

The notation of (S,G), pronounced “S comma G,” enumerates a shortest path tree where S is the IP address of the source and G is the multicast group address.

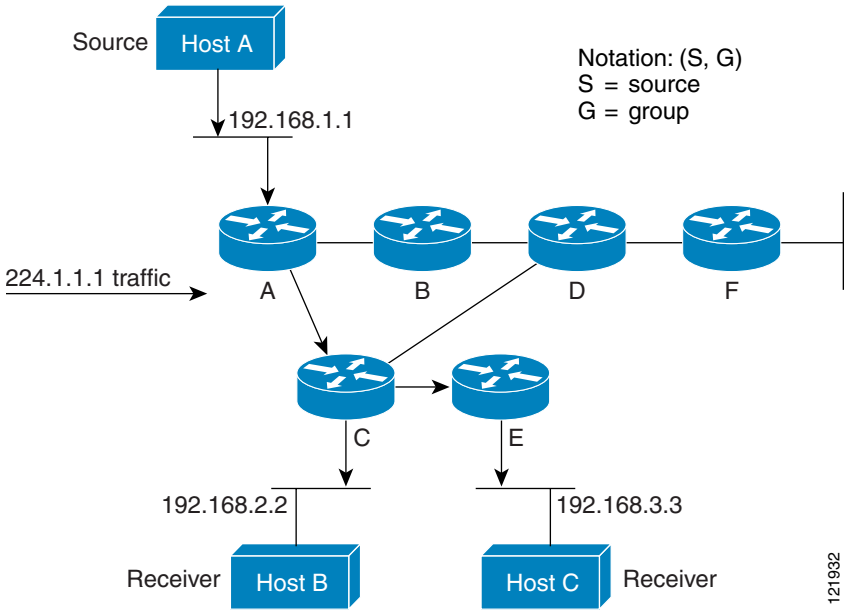
Shared trees are (*,G) and the source trees are (S,G) and always routed at the sources.

Multicast Distribution Source Tree (Shortest Path Tree)

The simplest form of a multicast distribution tree is a source tree. A source tree has its root at the source host and has branches forming a spanning tree through the network to the receivers. Because this tree uses the shortest path through the network, it is also referred to as a shortest path tree (SPT).

Figure 4 shows an example of an SPT for group 224.1.1.1 rooted at the source, Host A, and connecting two receivers, Hosts B and C.

Figure 4 Source Tree



Using standard notation, the SPT for the example shown in Figure 4 would be (192.168.1.1, 224.1.1.1).

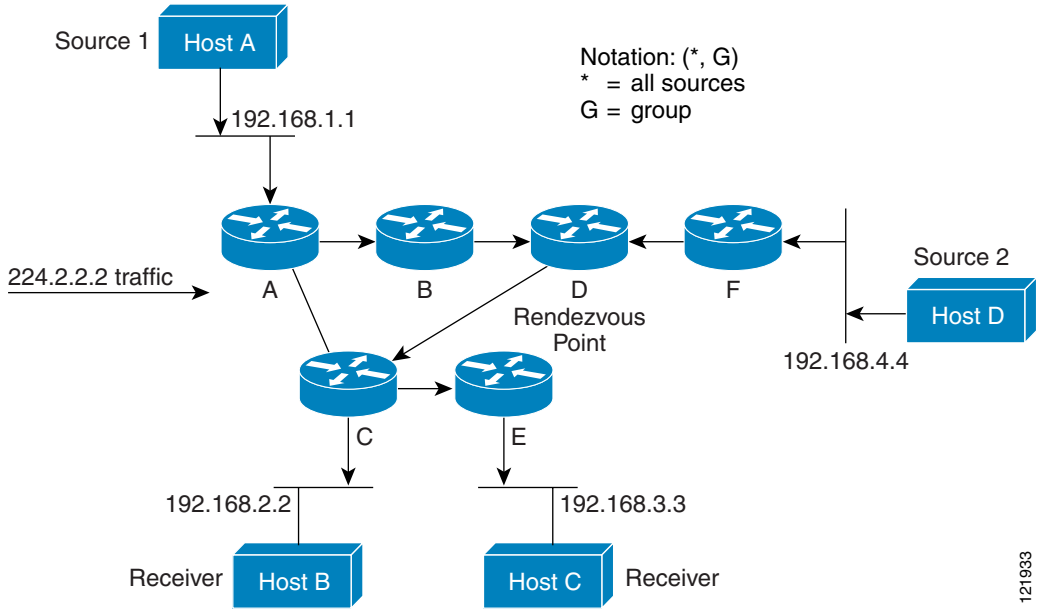
The (S,G) notation implies that a separate SPT exists for each individual source sending to each group—which is correct.

Multicast Distribution Shared Tree

Unlike source trees that have their root at the source, shared trees use a single common root placed at some chosen point in the network. This shared root is called a rendezvous point (RP).

Figure 5 shows a shared tree for the group 224.2.2.2 with the root located at Router D. This shared tree is unidirectional. Source traffic is sent towards the RP on a source tree. The traffic is then forwarded down the shared tree from the RP to reach all of the receivers (unless the receiver is located between the source and the RP, in which case it will be serviced directly).

Figure 5 Shared Distribution Tree



In this example, multicast traffic from the sources, Hosts A and D, travels to the root (Router D) and then down the shared tree to the two receivers, Hosts B and C. Because all sources in the multicast group use a common shared tree, a wildcard notation written as (*, G), pronounced "star comma G," represents the tree. In this case, * means all sources, and G represents the multicast group. Therefore, the shared tree shown in Figure 5 would be written as (*, 224.2.2.2).

Both source trees and shared trees are loop-free. Messages are replicated only where the tree branches. Members of multicast groups can join or leave at any time; therefore the distribution trees must be dynamically updated. When all the active receivers on a particular branch stop requesting the traffic for a particular multicast group, the routers prune that branch from the distribution tree and stop forwarding traffic down that branch. If one receiver on that branch becomes active and requests the multicast traffic, the router will dynamically modify the distribution tree and start forwarding traffic again.

Source Tree Advantage

Source trees have the advantage of creating the optimal path between the source and the receivers. This advantage guarantees the minimum amount of network latency for forwarding multicast traffic. However, this optimization comes at a cost. The routers must maintain path information for each source. In a network that has thousands of sources and thousands of groups, this overhead can quickly become a resource issue on the routers. Memory consumption from the size of the multicast routing table is a factor that network designers must take into consideration.

Shared Tree Advantage

Shared trees have the advantage of requiring the minimum amount of state in each router. This advantage lowers the overall memory requirements for a network that only allows shared trees. The disadvantage of shared trees is that under certain circumstances the paths between the source and receivers might not be the optimal paths, which might introduce some latency in packet delivery. For example, in Figure 5 the shortest path between Host A (source 1) and Host B (a receiver) would be Router A and Router C.

Because we are using Router D as the root for a shared tree, the traffic must traverse Routers A, B, D and then C. Network designers must carefully consider the placement of the rendezvous point (RP) when implementing a shared tree-only environment.

In unicast routing, traffic is routed through the network along a single path from the source to the destination host. A unicast router does not consider the source address; it considers only the destination address and how to forward the traffic toward that destination. The router scans through its routing table for the destination address and then forwards a single copy of the unicast packet out the correct interface in the direction of the destination.

In multicast forwarding, the source is sending traffic to an arbitrary group of hosts that are represented by a multicast group address. The multicast router must determine which direction is the upstream direction (toward the source) and which one is the downstream direction (or directions) toward the receivers. If there are multiple downstream paths, the router replicates the packet and forwards it down the appropriate downstream paths (best unicast route metric)—which is not necessarily all paths. Forwarding multicast traffic away from the source, rather than to the receiver, is called Reverse Path Forwarding (RPF). RPF is described in the following section.

Reverse Path Forwarding (RPF)

In unicast routing, traffic is routed through the network along a single path from the source to the destination host. A unicast router does not consider the source address; it considers only the destination address and how to forward the traffic toward that destination. The router scans through its routing table for the destination network and then forwards a single copy of the unicast packet out the correct interface in the direction of the destination.

In multicast forwarding, the source is sending traffic to an arbitrary group of hosts that are represented by a multicast group address. The multicast router must determine which direction is the upstream direction (toward the source) and which one is the downstream direction (or directions) toward the receivers. If there are multiple downstream paths, the router replicates the packet and forwards it down the appropriate downstream paths (best unicast route metric)—which is not necessarily all paths. Forwarding multicast traffic away from the source, rather than to the receiver, is called Reverse Path Forwarding (RPF). RPF is an algorithm used for forwarding multicast datagrams.

Protocol Independent Multicast (PIM) uses the unicast routing information to create a distribution tree along the reverse path from the receivers towards the source. The multicast routers then forward packets along the distribution tree from the source to the receivers. RPF is a key concept in multicast forwarding. It enables routers to correctly forward multicast traffic down the distribution tree. RPF makes use of the existing unicast routing table to determine the upstream and downstream neighbors. A router will forward a multicast packet only if it is received on the upstream interface. This RPF check helps to guarantee that the distribution tree will be loop-free.

RPF Check

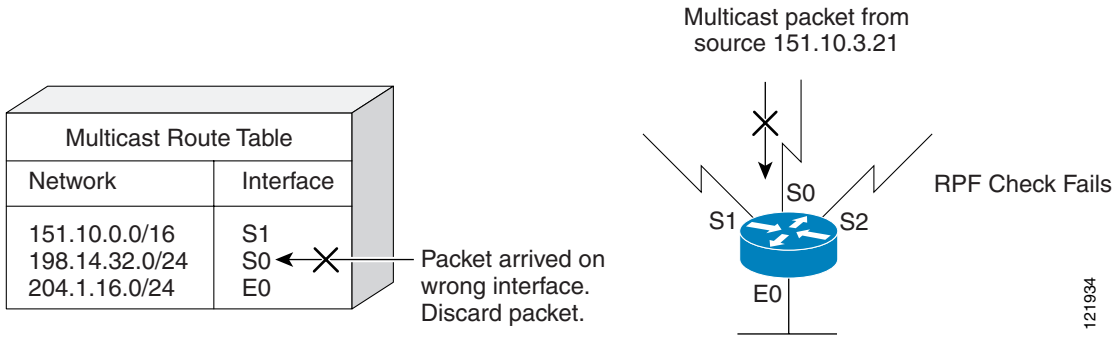
When a multicast packet arrives at a router, the router performs an RPF check on the packet. If the RPF check succeeds, the packet is forwarded. Otherwise, it is dropped.

For traffic flowing down a source tree, the RPF check procedure works as follows:

1. The router looks up the source address in the unicast routing table to determine if the packet has arrived on the interface that is on the reverse path back to the source.
2. If the packet has arrived on the interface leading back to the source, the RPF check succeeds and the packet is forwarded out the interfaces present in the outgoing interface list of a multicast routing table entry.
3. If the RPF check in Step 2 fails, the packet is dropped.

Figure 6 shows an example of an unsuccessful RPF check.

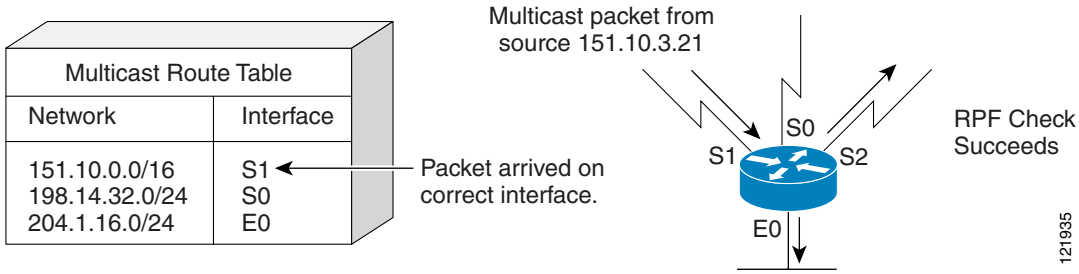
Figure 6 RPF Check Fails



As Figure 6 illustrates, a multicast packet from source 151.10.3.21 is received on serial interface 0 (S0). A check of the unicast route table shows that S1 is the interface this router would use to forward unicast data to 151.10.3.21. Because the packet has arrived on interface S0, the packet is discarded.

Figure 7 shows an example of a successful RPF check.

Figure 7 RPF Check Succeeds



In this example, the multicast packet has arrived on interface S1. The router refers to the unicast routing table and finds that S1 is the correct interface. The RPF check passes, and the packet is forwarded.

PIM Dense Mode Fallback

If you use IP multicast in mission-critical networks, you should avoid the use of PIM-DM (dense mode).

Dense mode fallback describes the event of the PIM mode changing (falling back) from sparse mode (which requires an RP) to dense mode (which does not use an RP). Dense mode fallback occurs when RP information is lost.

By default, if all interfaces in a multicast VPN routing or forwarding instance are configured with the **ip pim sparse-mode** command, there is no dense mode fallback because dense mode groups cannot be formed over interfaces configured for sparse mode.

Cause and Effect of Dense Mode Fallback

PIM determines whether a multicast group operates in PIM-DM or PIM sparse-dense mode based solely on the existence of RP information in the group-to-RP mapping cache. If Auto-RP is configured or a bootstrap router (BSR) is used to distribute RP information, there is a risk that RP information can be lost if all RPs, Auto-RP, or the BSR for a group fails due to network congestion. This failure can lead to the network either partially or fully falling back into PIM-DM.

If a network falls back into PIM-DM, and if interfaces are configured for Auto-RP, dense mode flooding will occur. Routers that lose RP information will switch all existing states into dense mode and any new states that must be created for the failed group will be created in dense mode.

Effects of Preventing Dense Mode Fallback

The PIM Dense Mode Fallback Prevention in a Network Following RP Information Loss feature provides a method for doing the following:

- Preventing dense mode fallback.
- Blocking multicast traffic for groups not specifically configured. When a group is not specifically configured (that is, there is no RP for that group), multicast traffic does not flow across the network. The first hop router that receives data for such a group creates a (*, G) entry for that group with an RP address of 0.0.0.0 and drops these packets. There are no (S,G) entries created on the routers.

When the feature is configured, sparse mode groups operate with an RP address of 0.0.0.0, which causes the following conditions to apply:

- (S,G) state is maintained unchanged and existing flows are unchanged.
- No PIM Join or Prune messages for (*, G) or (S,G, RPbit) are sent.
- Received (*, G) or (S,G, RPbit) Joins or Prune messages are ignored.
- No registers are sent and traffic at the first hop is dropped.
- Received registers are answered with register stop.
- Asserts are unchanged.
- The (*, G) outgoing interface (OIF) list is maintained only for the Internet Group Management Protocol (IGMP) state.
- Multicast Source Discovery Protocol (MSDP) source active (SA) messages for RP 0.0.0.0 groups are still accepted and forwarded.
- IGMP messages are processed as for any other PIM-SM groups. IGMPv3 (S,G) memberships are expanded into (*, G) IGMP memberships.
- Any group for which no mode information is available (such as PIM sparse mode, bidirectional PIM, or Source Specific Multicast [SSM]) are created in a default mode in which no traffic is forwarded, and no PIM states are established in the network apart from the network first hop.

Benefits of Preventing PIM Dense Mode Fallback in a Network Following RP Information Loss

The benefits of preventing PIM dense mode fallback are the following:

- You can block multicast traffic for groups not specifically configured. When a group is not specifically configured (that is, there is no RP for that group), multicast traffic does not flow across the network. The first hop router that receives data for such a group creates a (*, G) entry for that group with an RP address of 0.0.0.0 and drops these packets. There are no (S,G) entries created on the routers.

- If interfaces are configured with **ip pim sparse-dense-mode** (such as when Auto-RP is in use), fallback into PIM dense mode can cause undesirable dense mode flooding of multicast packets and outages of multicast traffic flows. These events can be avoided by preventing dense mode fallback.

Prevention of Dense Mode Fallback

By default, PIM dense mode fallback is enabled. That is, a multicast group in the absence of rendezvous point (RP) information will fall to dense mode, regardless of the interface mode configuration.

However, if all of the interfaces in a VRF are configured with PIM sparse mode, no dense mode fallback is achieved by default. If all of the interfaces are already configured as sparse, even though the group mode falls to dense mode, the traffic does not get flooded (due to the sparse characteristic of the interface). But the established flows might be torn down and the state of the network could become indeterministic. The main advantage of no dense mode fallback in this case would be deterministic behavior.

If you have interfaces that are configured with PIM sparse-dense mode, you can disable dense mode fallback by configuring the **no ip pim dm-fallback** command in global configuration mode.

Guidelines for Choosing a PIM Mode

Before beginning the configuration process, you must decide which PIM mode needs to be used. This determination is based on the applications you intend to support on your network.

Basic guidelines include the following:

- In general, if the application is one-to-many or many-to-many in nature, then PIM-SM can be used successfully.
- For optimal one-to-many application performance, SSM is appropriate but requires IGMP version 3 support.
- For optimal many-to-many application performance, bidirectional PIM is appropriate but hardware support is limited to Cisco IOS devices and the Catalyst 6000 series switches with Sup720.

Where to Go Next

- To configure basic IP multicast, see the "Configuring Basic IP Multicast" module.

Additional References

The following sections provide references related to IP multicast.

Related Documents

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

MIBs

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

RFCs

RFC	Title
RFC 1112	*Host Extensions for IP Multicasting*
RFC 2113	*IP Router Alert Option*
RFC 2362	*Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification*
RFC 3180	*GLOP Addressing in 233/8*

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 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/public/support/tac/home.shtml

Glossary

basic multicast—Interactive intra-domain multicast. Supports multicast applications within an enterprise campus. Also provides an additional integrity in the network with the inclusion of a reliable multicast transport, PGM.

bidir PIM—Bidirectional PIM is an extension to the PIM suite of protocols that implements shared sparse trees with bidirectional flow of data. In contrast to PIM-SM, bidir-PIM avoids keeping source specific state in router and thus allows trees to scale to an arbitrary number of sources.

broadcast—One-to-all transmission where the source sends one copy of the message to all nodes, whether they wish to receive it or not.

Cisco Group Management Protocol (CGMP)—Cisco-developed protocol that allows Layer 2 switches to leverage IGMP information on Cisco routers to make Layer 2 forwarding decisions. It allows the switches to forward multicast traffic to only those ports that are interested in the traffic.

dense mode (DM) (Internet Draft Spec)—Actively attempts to send multicast data to all potential receivers (flooding) and relies upon their self-pruning (removal from group) to achieve desired distribution.

designated router (DR)—The router in a PIM-SM tree that instigates the Join/Prune message cascade upstream to the RP in response to IGMP membership information it receives from IGMP hosts.

distribution tree—Multicast traffic flows from the source to the multicast group over a distribution tree that connects all of the sources to all of the receivers in the group. This tree may be shared by all sources (a shared-tree), or a separate distribution tree can be built for each source (a source-tree). The shared-tree may be one-way or bidirectional.

IGMP messages—IGMP messages are encapsulated in standard IP datagrams with an IP protocol number of 2 and the IP Router Alert option (RFC 2113).

IGMP snooping—IGMP snooping requires the LAN switch to examine, or "snoop," some Layer 3 information in the IGMP packet sent from the host to the router. When the switch hears an IGMP report from a host for a particular multicast group, the switch adds the host's port number to the associated multicast table entry. When it hears an IGMP Leave Group message from a host, it removes the host's port from the table entry.

IGMP unidirectional link routing—Cisco's other UDLR solution is to use IP multicast routing with IGMP, which has been enhanced to accommodate UDLR. This solution scales very well for many satellite links.

Internet Group Management Protocol v2 (IGMP)—Used by IP routers and their immediately connected hosts to communicate multicast group membership states.

Internet Group Management Protocol v3 (IGMP)—IGMP is the protocol used by IPv4 systems to report their IP multicast group memberships to neighboring multicast routers. Version 3 of IGMP adds support for "source filtering," that is, the ability for a system to report interest in receiving packets only from specific source addresses, or from all but specific source addresses, sent to a particular multicast address.

multicast—A routing technique that allows IP traffic to be sent from one source or multiple sources and delivered to multiple destinations. Instead of sending individual packets to each destination, a single packet is sent to a group of destinations known as a multicast group, which is identified by a single IP destination group address. Multicast addressing supports the transmission of a single IP datagram to multiple hosts.

multicast routing monitor (MRM)—A management diagnostic tool that provides network fault detection and isolation in a large multicast routing infrastructure. It is designed to notify a network administrator of multicast routing problems in near real time.

Multicast Source Discovery Protocol (MSDP)—A mechanism to connect multiple PIM sparse mode (PIM-SM) domains. MSDP allows multicast sources for a group to be known to all rendezvous point(s) (RPs) in different domains. Each PIM-SM domain uses its own RPs and need not depend on RPs in other domains. An RP runs MSDP over TCP to discover multicast sources in other domains. MSDP is also used to announce sources sending to a group. These announcements must originate at the domain's RP. MSDP depends heavily on MBGP for interdomain operation.

Protocol Independent Multicast (PIM)—A multicast routing architecture defined by the IETF that enables IP multicast routing on existing IP networks. Its key point is its independence from any underlying unicast protocol such as OSPF or BGP.

prune—Multicast routing terminology indicating that the multicast-enabled router has sent the appropriate multicast messages to remove itself from the multicast tree for a particular multicast group. It will stop receiving the multicast data addressed to that group and, therefore, cannot deliver the data to any connected hosts until it rejoins the group.

query—IGMP messages originating from the router(s) to elicit multicast group membership information from its connected hosts.

rendezvous point (RP)—The multicast router that is the root of the PIM-SM shared multicast distribution tree.

report—IGMP messages originating from the hosts that are joining, maintaining, or leaving their membership in a multicast group.

source tree—A multicast distribution path that directly connects the source's and receivers' designated router (or the rendezvous point) to obtain the shortest path through the network. Results in most efficient routing of data between source and receivers, but may result in unnecessary data duplication throughout the network if built by anything other than the RP.

sparse mode (SM) (RFC 2362)—Relies upon an explicitly joining method before attempting to send multicast data to receivers of a multicast group.

UDLR tunnel—Uses a back channel (another link) so the routing protocols believe the one-way link is bidirectional. The back channel itself is a special, unidirectional, generic route encapsulation (GRE) tunnel through which control traffic flows in the opposite direction of the user data flow. This feature allows IP and its associated unicast and multicast routing protocols to believe the unidirectional link is logically bidirectional. This solution accommodates all IP unicast and multicast routing protocols without changing them. However, it does not scale and no more than 20 tunnels should feed into the upstream router. The purpose of the unidirectional GRE tunnel is to move control packets from a downstream node to an upstream node.

Unicast—Point-to-point transmission requiring the source to send an individual copy of a message to each requester.

unidirectional Link Routing Protocol (UDLR)—A routing protocol that provides a way to forward multicast packets over a physical unidirectional interface (such as a satellite link of high bandwidth) to stub networks that have a back channel.

URL rendezvous directory (URD)—URD is a multicast-lite solution that directly provides the network with information about the specific source of a content stream. It enables the network to quickly establish the most direct distribution path from the source to the receiver, thus significantly reducing the time and effort required in receiving the streaming media. URD allows an application to identify the source of the content stream through a web page link or web directly. When that information is sent back to the application it is then conveyed back to the network using URD.

In this feature, a URD-capable web page provides information about the source, the group, and the application (via media-type) on a web page. An interested host will click on the web page pulling across the information in an HTTP transaction. The last-hop router to receiver would intercept this transaction and send it to a special port allocated by IANA. The last-hop router is also URD capable and uses the information to initiate the PIM source, group (S,G) join on behalf of the host.

Note Refer to *Internetworking Terms and Acronyms* for terms not included in this glossary.

Feature Information for IP Multicast Technology Overview

Table 2 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Releases 12.2(1) or later appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for specific commands was introduced, see the command reference documents.

If you are looking for information on a feature in this technology that is not documented here, see the "IP Multicast Features Roadmap."

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator (http://www.cisco.com/go/fn). You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click Cancel at the login dialog box and follow the instructions that appear.

Note Table 2 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 2 Feature Information for IP Multicast Technology Overview

Feature Names	Releases	Feature Configuration Information
PIM Dense Mode Fallback Prevention in a Network Following RP Information Loss	12.3(4)T	This feature enables you to prevent PIM dense mode fallback when all RPs fail. Preventing the use of PIM dense mode is very important to multicast networks whose reliability is critical. The feature provides a mechanism to keep the multicast groups in sparse mode. The feature also allows you to block multicast traffic for groups not specifically configured. The following sections provide information about this feature: • PIM Dense Mode Fallback, page 19

Configuring Basic IP Multicast

First Published: May 2, 2005
Last Updated: August 21, 2007

IP multicast is a bandwidth-conserving technology that reduces traffic by delivering a single stream of information simultaneously to potentially thousands of corporate businesses and homes. Applications that take advantage of multicast include video conferencing, corporate communications, distance learning, and distribution of software, stock quotes, and news. This module describes the tasks used to configure basic IP multicast.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Configuring Basic IP Multicast" section on page 38.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Configuring Basic IP Multicast

- Before performing the tasks in this module, you should be familiar with the concepts explained in the "IP Multicast Technology Overview" module.
- To determine which of the tasks contained in this module you will have to perform, you must decide which Protocol Independent Multicast (PIM) mode will be used. This determination is based on the applications you intend to support on your network.
- All access lists you intend to use with the tasks in this module should be configured prior to beginning the configuration task. For information about how to configure an access list, see the "Creating an IP Access List and Applying It to an Interface" module.

How to Configure Basic IP Multicast

The tasks described in this section configure the basic IP multicast modes. No single task in this section is required; however, at least one of the tasks must be performed to configure IP multicast in a network. More than one of the tasks may be needed. This section contains the following tasks:

- Configuring Sparse Mode with Auto-RP, page 2
- Configuring Sparse Mode with Anycast RP, page 8
- Configuring Sparse Mode with a Bootstrap Router, page 12
- Configuring Sparse Mode with a Single Static RP, page 16
- Configuring Source Specific Multicast, page 19
- Configuring Bidirectional PIM, page 25

Configuring Sparse Mode with Auto-RP

This section contains information about and instructions on how to configure auto-rendezvous point (Auto-RP). Auto-RP can also be optionally used with anycast RP, which is described in the "Configuring Sparse Mode with Anycast RP"section.

Note The simultaneous deployment of Auto-RP and bootstrap router (BSR) is not supported.

The Role of Auto-RP in a PIM Network

Auto-RP automates the distribution of group-to-rendezvous point (RP) mappings in a PIM network. To make Auto-RP work, a router must be designated as an RP mapping agent, which receives the RP announcement messages from the RPs and arbitrates conflicts. The RP mapping agent then sends the consistent group-to-RP mappings to all other routers by way of dense mode flooding.

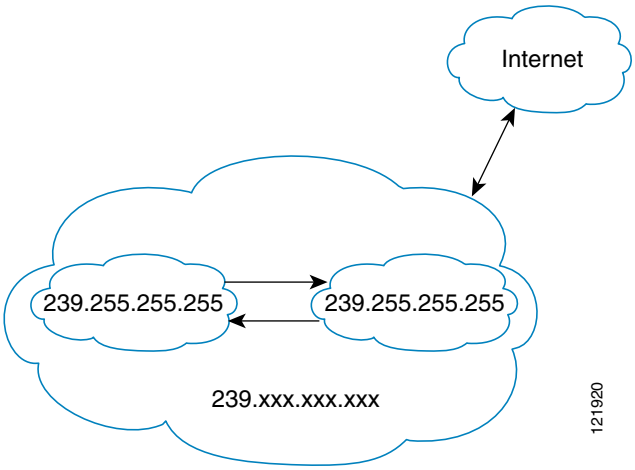
Thus, all routers automatically discover which RP to use for the groups they support. The Internet Assigned Numbers Authority (IANA) has assigned two group addresses, 224.0.1.39 and 224.0.1.40, for Auto-RP.

The mapping agent receives announcements of intention to become the RP from Candidate-RPs. The mapping agent then announces the winner of the RP election. This announcement is made independently of the decisions by the other mapping agents.

IP Multicast Boundary

As shown in Figure 1, address scoping defines domain boundaries so that domains with RPs that have the same IP address do not leak into each other. Scoping is performed on the subnet boundaries within large domains and on the boundaries between the domain and the Internet.

Figure 1 Address Scoping at Boundaries



You can set up an administratively scoped boundary on an interface for multicast group addresses using the **ip multicast boundary** command with the *access-list* argument. A standard access list defines the range of addresses affected. When a boundary is set up, no multicast data packets are allowed to flow across the boundary from either direction. The boundary allows the same multicast group address to be reused in different administrative domains.

The Internet Assigned Numbers Authority (IANA) has designated the multicast address range 239.0.0.0 to 239.255.255.255 as the administratively scoped addresses. This range of addresses can be reused in domains administered by different organizations. They would be considered local, not globally unique.

You can configure the **filter-autorp** keyword to examine and filter Auto-RP discovery and announcement messages at the administratively scoped boundary. Any Auto-RP group range announcements from the Auto-RP packets that are denied by the boundary access control list (ACL) are removed. An Auto-RP group range announcement is permitted and passed by the boundary only if all addresses in the Auto-RP group range are permitted by the boundary ACL. If any address is not permitted, the entire group range is filtered and removed from the Auto-RP message before the Auto-RP message is forwarded.

Benefits of Auto-RP in a PIM Network

- Auto-RP allows any change to the RP designation to be configured only on the routers that are RPs, not on the leaf routers.
- Auto-RP offers the ability to scope the RP address within a domain. Scoping can be achieved by using the **ip multicast boundary** command with the **filter-autorp** keyword.

Prerequisites

- When configuring Auto-RP, you must either configure the Auto-RP listener feature using the **ip pim autorp listener** command (Step 5) and specify sparse mode using the **ip pim sparse-mode** command (Step 7) or specify sparse-dense mode (Step 8) using the **ip pim sparse-dense mode** command.

Note When you configure sparse-dense mode, dense mode failover may result in a network dense-mode flood. To avoid this condition, use PIM sparse mode with the Auto-RP listener feature.

- An interface configured in sparse-dense mode is treated in either sparse mode or dense mode of operation, depending on the mode in which the multicast group operates. You must decide how to configure your interfaces.
- All access lists that are needed when Auto-RP is configured should be configured prior to beginning the configuration task. For information about how to configure an access list, see the "Creating an IP Access List and Applying It to an Interface" module.

Restrictions

If a group has no known RP and the interface is configured to be sparse-dense mode, the interface is treated as if it were in dense mode, and data is flooded over the interface. To avoid this data flooding, configure the Auto-RP listener using the **ip pim autorp listener** command and then configure the interface as sparse mode using the **ip pim sparse mode** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast-routing** [**distributed**]
4. Either perform Steps 5 through 7 or perform Steps 6 and 8.
5. **ip pim autorp listener**
6. **interface** *type number*
7. **ip pim sparse-mode**
8. **ip pim sparse-dense-mode**
9. **exit**
10. Repeat Steps 1 through 9 on all PIM interfaces.
11. **ip pim send-rp-announce** {*interface-type interface-number* | *ip-address*} **scope** *ttl-value* [**group-list** *access-list*] [**interval** *seconds*] [**bidir**]
12. **ip pim send-rp-discovery** [*interface-type interface-number*] **scope** *ttl-value* [**interval** *seconds*]
13. **ip pim rp-announce-filter rp-list** *access-list* **group-list** *access-list*
14. **no ip pim dm-fallback**
15. **interface** *type number*
16. **ip multicast boundary** *access-list* [**filter-autorp**]

17. **end**
18. **show ip pim autorp**
19. **show ip pim rp** [**mapping**] [*rp-address*]
20. **show ip igmp groups** [*group-name* | *group-address* | *interface-type interface-number*] [**detail**]
21. **show ip mroute** [*group-address* | *group-name*] [*source-address* | *source-name*] [*interface-type interface-number*] [**summary**] [**count**] [**active** *kbps*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast-routing** [**distributed**] **Example:** `Router(config)# ip multicast-routing`	Enables IP multicast routing. • Use the **distributed** keyword to enabled Multicast Distributed Switching.
Step 4	Either perform Steps 5 through 7 or perform Steps 6 and 8.	—
Step 5	**ip pim autorp listener** **Example:** `Router(config)# ip pim autorp listener`	Causes IP multicast traffic for the two Auto-RP groups 224.0.1.39 and 224.0.1.40 to be PIM dense mode flooded across interfaces operating in PIM sparse mode. • Skip this step if you are configuring sparse-dense mode in Step 8.
Step 6	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 7	**ip pim sparse-mode** **Example:** `Router(config-if)# ip pim sparse-mode`	Enables PIM sparse mode on an interface. When configuring Auto-RP in sparse mode, you must also configure the Auto-RP listener in the next step. • Skip this step if you are configuring sparse-dense mode in Step 8.
Step 8	**ip pim sparse-dense-mode** **Example:** `Router(config-if)# ip pim sparse-dense-mode`	Enables PIM sparse-dense mode on an interface. • Skip this step if you configured sparse mode in Step 7.

	Command or Action	Purpose
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	Repeat Steps 1 through 9 on all PIM interfaces.	—
Step 11	**ip pim send-rp-announce** {*interface-type interface-number* | *ip-address*} **scope** *ttl-value* [**group-list** *access-list*] [**interval** *seconds*] [**bidir**] **Example:** `Router(config)# ip pim send-rp-announce loopback0 scope 31 group-list 5`	Sends RP announcements out all PIM-enabled interfaces. • Perform this step on the RP router only. • Use the *interface-type* and *interface-number* arguments to define which IP address is to be used as the RP address. • Use the *ip-address* argument to specify a directly connected IP address as the RP address. **Note** If the *ip-address* argument is configured for this command, the RP-announce message will be sourced by the interface to which this IP address is connected (that is, the source address in the IP header of the RP-announce message is the IP address of that interface). • This example shows that the interface is enabled with a maximum of 31 hops. The IP address by which the router wants to be identified as RP is the IP address associated with loopback interface 0. Access list 5 describes the groups for which this router serves as RP.
Step 12	**ip pim send-rp-discovery** [*interface-type interface-number*] **scope** *ttl-value* [**interval** *seconds*] **Example:** `Router(config)# ip pim send-rp-discovery loopback 1 scope 31`	Configures the router to be an RP mapping agent. • Perform this step on the RP router only. • Use the optional *interface-type* and *interface-number* arguments to define which IP address is to be used as the source address of the RP mapping agent. • Use the **scope** keyword and *ttl-value* argument to specify the Time-to-Live (TTL) value in the IP header of Auto-RP discovery messages. • Use the optional **interval** keyword and *seconds* argument to specify the interval at which Auto-RP discovery messages are sent. **Note** Lowering the interval at which Auto-RP discovery messages are sent from the default value of 60 seconds results in more frequent floodings of the group-to-RP mappings. In some network environments, the disadvantages of lowering the interval (more control packet overhead) may outweigh the advantages (more frequent group-to-RP mapping updates). • The example shows limiting the Auto-RP discovery messages to 31 hops on loopback interface 1.

	Command or Action	Purpose
Step 13	**ip pim rp-announce-filter rp-list** *access-list* **group-list** *access-list* **Example:** `Router(config)# ip pim rp-announce-filter rp-list 1 group-list 2`	Filters incoming Auto-RP announcement messages coming from the RP. • Perform this step on the RP router only. • Two example access lists that apply to this step could be: `access-list 1 permit 10.0.0.1` `access-list 1 permit 10.0.0.2` `access-list 2 permit 224.0.0.0 15.255.255.255`
Step 14	**no ip pim dm-fallback** **Example:** `Router(config)# no ip pim dm-fallback`	(Optional) Prevents PIM dense mode fallback. • Configure this command on all routers in a PIM sparse-mode domain.
Step 15	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 16	**ip multicast boundary** *access-list* [**filter-autorp**] **Example:** `Router(config-if)# ip multicast boundary 10 filter-autorp`	Configures an administratively scoped boundary. • Perform this step on the interfaces that are boundaries to other routers. • The access list is not shown in this task. • An access list entry that uses the **deny** keyword creates a multicast boundary for packets that match that entry.
Step 17	**end** **Example:** `Router(config-if)# end`	Returns to global configuration mode.
Step 18	**show ip pim autorp** **Example:** `Router# show ip pim autorp`	(Optional) Displays the Auto-RP information.
Step 19	**show ip pim rp** [**mapping**] [*rp-address*] **Example:** `Router# show ip pim rp mapping`	(Optional) Displays RPs known in the network and shows how the router learned about each RP.

	Command or Action	Purpose
Step 20	**show ip igmp groups** [*group-name* | *group-address* | *interface-type interface-number*] [**detail**] **Example:** `Router# show ip igmp groups`	(Optional) Displays the multicast groups having receivers that are directly connected to the router and that were learned through Internet Group Management Protocol (IGMP). • A receiver must be active on the network at the time that this command is issued in order for receiver information to be present on the resulting display.
Step 21	**show ip mroute** [*group-address* | *group-name*] [*source-address* | *source-name*] [*interface-type interface-number*] [**summary**] [**count**] [**active** *kbps*] **Example:** `Router# show ip mroute cbone-audio`	(Optional) Displays the contents of the IP multicast routing (mroute) table.

What to Do Next

Proceed to the "Verifying IP Multicast Operation" module.

Configuring Sparse Mode with Anycast RP

This section describes how to configure sparse mode with anycast RP for RP redundancy.

Anycast RPs are configured statically, and interfaces are configured to operate in Protocol Independent Multicast-Sparse Mode (PIM-SM). In an anycast RP configuration, two or more RPs are configured with the same IP address on loopback interfaces. The Anycast RP loopback address should be configured with a 32-bit mask, making it a host address. An Anycast RP configuration is easy to configure and troubleshoot because the same host address is used as the RP address regardless of which router it is configured on.

Anycast RP allows two or more rendezvous points (RPs) to share the load for source registration and have the ability to act as hot backup routers for each other. Multicast Source Discovery Protocol (MSDP) is the key protocol that makes anycast RP possible.

Multicast Source Discovery Protocol Overview

In the PIM sparse mode model, multicast sources and receivers register with their local rendezvous point (RP). Operationally the router closest to a source or a receiver registers with the RP. RPs in other domains have no way of knowing about sources that are located in other domains.

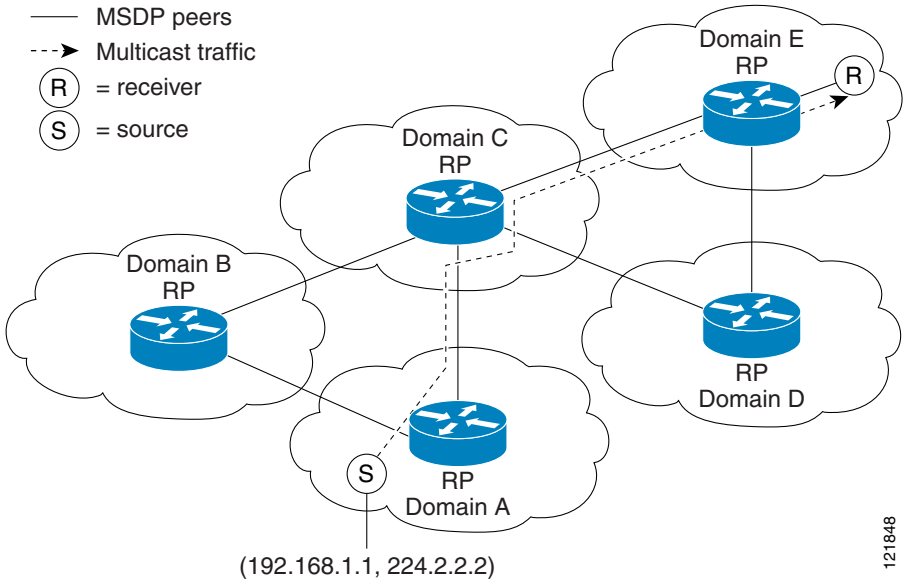
MSDP is a mechanism that allows RPs to share information about active sources. RPs register the receivers in their local domain. When RPs in remote domains hear about the active sources, they can pass on that information to their local receivers. Multicast data can then be forwarded between the domains. Another function of MSDP is that it allows each domain to maintain an independent RP that does not rely on other domains, and MSDP allows the RP to share knowledge about active sources between domains. PIM-SM is used to forward the traffic between the multicast domains.

The RP in each domain establishes an MSDP peering session using a TCP connection with the RPs in other domains or with border routers leading to the other domains. When the RP learns about a new multicast source within its own domain (through the normal PIM register mechanism), the RP encapsulates the first data packet in a Source-Active (SA) message and sends the SA to all MSDP peers.

Each receiving peer uses a modified Reverse Path Forwarding (RPF) check to forward the SA until the SA has reached every MSDP router in the interconnected networks—theoretically the entire multicast Internet. If the receiving MSDP peer is an RP, and the RP has a (*, G) entry for the group in the SA (meaning that there is an interested receiver), the RP creates (Source, Group) (S, G) state for the source and joins to the shortest path tree for the source. The encapsulated data is decapsulated and forwarded down the shared tree of that RP. When the last-hop router (the router closest to the receiver) receives the multicast packet, it may join the shortest path tree to the source. The MSDP speaker periodically sends SAs that include all sources within the domain of the RP. Figure 2 shows how data would flow from a source in domain A to a receiver in domain E.

MSDP was developed for peering among Internet service providers (ISPs). ISPs did not want to rely on an RP maintained by a competing ISP to provide service to their customers. MSDP allows each ISP to have its own local RP and still forward multicast traffic to the Internet and receive multicast traffic from the Internet.

Figure 2 MSDP Sharing Source Information Between RPs in Each Domain



Anycast RP Overview

Anycast RP is a useful application of MSDP. Originally developed for interdomain multicast applications, MSDP used for Anycast RP is an intradomain feature that provides redundancy and load-sharing capabilities. Enterprise customers typically use Anycast RP for configuring a Protocol Independent Multicast sparse mode (PIM-SM) network to meet fault tolerance requirements within a single multicast domain.

In anycast RP, two or more RPs are configured with the same IP address on loopback interfaces. The anycast RP loopback address should be configured with a 32-bit mask, making it a host address. All the downstream routers should be configured so that the anycast RP loopback address is the IP address of their local RP. IP routing will automatically select the topologically closest RP for each source and receiver. Assuming that the sources are evenly spaced around the network, an equal number of sources will register with each RP. That is, the process of registering the sources will be shared equally by all the RPs in the network.

Because a source may register with one RP and receivers may join to a different RP, a method is needed for the RPs to exchange information about active sources. This information exchange is done with MSDP.

In anycast RP, all the RPs are configured to be MSDP peers of each other. When a source registers with one RP, an SA message will be sent to the other RPs informing them that there is an active source for a particular multicast group. The result is that each RP will know about the active sources in the area of the other RPs. If any of the RPs were to fail, IP routing would converge, and one of the RPs would become the active RP in more than one area. New sources would register with the backup RP. Receivers would join the new RP and connectivity would be maintained.

The RP is normally needed only to start new sessions with sources and receivers. The RP facilitates the shared tree so that sources and receivers can establish a direct multicast data flow. If a multicast data flow is already established between a source and the receiver, an RP failure will not affect that session. Anycast RP ensures that new sessions with sources and receivers can begin at any time.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast-routing** [**distributed**]
4. **interface** *type number*
5. **ip pim sparse-mode**
6. **ip pim rp-address** *rp-address* [*access-list*] [**override**]
7. Repeat Steps 1 through 6 on two or more routers assigning the same RP address to each.
8. **interface loopback** [*interface-number*] **ip address** [*ip-address*] [*mask*]
9. **interface loopback** [*interface-number*] **ip address** [*ip-address*] [*mask*]
10. **exit**
11. **ip msdp peer** {*peer-name* | *peer-address*} [**connect-source** *interface-type interface-number*] [**remote-as** *as-number*]
12. **ip msdp originator-id loopback** [*interface*]
13. **no ip pim dm-fallback**
14. Repeat Steps 8 through 13 on the redundant RPs.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast-routing** [**distributed**] **Example:** `Router(config)# ip multicast-routing`	Enables IP multicast routing. • Use the **distributed** keyword to enable Multicast Distributed Switching.
Step 4	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 5	**ip pim sparse-mode** **Example:** `Router(config-if)# ip pim sparse-mode`	Enables sparse mode.
Step 6	**ip pim rp-address** *rp-address* [*access-list*] [**override**] **Example:** `Router(config-if)# ip pim rp-address 10.0.0.1`	Configures the address of a PIM RP for a particular group.
Step 7	Repeat Steps 1 through 6 on two or more routers assigning the same RP address to each.	—
Step 8	**interface loopback** [*interface-number*] **ip address** [*ip-address*] [*mask*] **Example:** `Router(config-if)# interface loopback 0` `ip address 10.0.0.1 255.255.255.255`	Configures the interface loopback IP address for the RP router. • Perform this step on the RP routers.
Step 9	**interface loopback** [*interface-number*] **ip address** [*ip-address*] [*mask*] **Example:** `Router(config-if)# interface loopback 1` `ip address 10.1.1.1 255.255.255.255`	Configures the interface loopback IP address for MSDP peering.
Step 10	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 11	**ip msdp peer** {*peer-name* | *peer-address*} [**connect-source** *interface-type* **interface-number**] [**remote-as** *as-number*] **Example:** `Router(config)# ip msdp peer 10.1.1.2 connect-source loopback 1`	Configures an MSDP peer. • Perform this step on the RP routers.
Step 12	**ip msdp originator-id loopback** [*interface*] **Example:** `Router(config)# ip msdp originator-id loopback 1`	Allows an MSDP speaker that originates a SA message to use the IP address of the interface as the RP address in the SA message. • Perform this step on the RP routers.
Step 13	**no ip pim dm-fallback** **Example:** `Router(config)# no ip pim dm-fallback`	(Optional) Prevents PIM dense mode fallback. • Configure this command on all routers in a PIM sparse-mode domain.
Step 14	Repeat Steps 8 through 13 on the redundant RPs.	—

What to Do Next

Proceed to the "Verifying IP Multicast Operation" module.

Configuring Sparse Mode with a Bootstrap Router

This section describes how to configure a bootstrap router (BSR), which provides a fault-tolerant, automated RP discovery and distribution mechanism so that routers learn the group-to-RP mappings dynamically.

Note The simultaneous deployment of Auto-RP and BSR is not supported.

BSR Election and Functionality

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 performed 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; they use bootstrap messages to discover which BSR has the highest priority. This router then announces to all PIM routers in the PIM domain that it is the BSR.

Following the election of the BSR, candidate RPs use unicast to announce to the BSR their willingness to be the RP. The BSR then announces the winner by way of BSR messages sent on each link between PIM routers to the PIM router link local address 224.0.0.13.

BSR lacks the ability to scope RP advertisements; however, BSR is used when vendor interoperability or open standard adherence is a requirement.

BSR Border Interface

A border interface in a PIM sparse mode domain requires precautions to prevent exchange of certain traffic with a neighboring domain reachable through that interface, especially if that domain is also running PIM sparse mode. BSR and Auto-RP messages should not be exchanged between different domains, because routers in one domain may elect RPs in the other domain, resulting in protocol malfunction or loss of isolation between the domains. Configure a BSR border interface to prevent BSR messages from being sent or received through an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast-routing** [**distributed**]
4. **interface** *type number*
5. **ip pim sparse-mode**
6. **end**
7. Repeat Steps 1 through 6 on every multicast-enabled interface on every router.
8. **ip pim bsr-candidate** *interface-type interface-number* [*hash-mask-length*] [*priority*]
9. **ip pim rp-candidate** *interface-type interface-number* [**group-list** *access-list*] [**interval** *seconds*] [**priority** *value*]
10. **no ip pim dm-fallback**
11. Repeat Steps 8 through 10 on all RP and BSR routers.
12. **interface** *type number*
13. **ip pim bsr-border**
14. **end**
15. Repeat Steps 12 through 14 on all the routers that have boundary interfaces where the messages should not be sent or received.
16. **show ip pim rp** [**mapping**] [*rp-address*]
17. **show ip pim rp-hash** [*group-address*] [*group-name*]
18. **show ip pim bsr-router**
19. **show ip igmp groups** [*group-name* | *group-address* | *interface-type interface-number*] [**detail**]
20. **show ip mroute**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast-routing** [**distributed**] **Example:** `Router(config)# ip multicast-routing`	Enables IP multicast routing. • Use the **distributed** keyword to enable Multicast Distributed Switching.
Step 4	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 5	**ip pim sparse-mode** **Example:** `Router(config-if)# ip pim sparse-mode`	Enables sparse mode.
Step 6	**end** **Example:** `Router(config-if)# end`	Returns to global configuration mode.
Step 7	Repeat Steps 1 through 6 on every multicast-enabled interface on every router.	—
Step 8	**ip pim bsr-candidate** *interface-type interface-number* [*hash-mask-length*] [*priority*] **Example:** `Router(config)# ip pim bsr-candidate ethernet 0 192`	Configures the router to announce its candidacy as a bootstrap router (BSR). • Perform this step on the RP and BSR routers. • The routers to serve as candidate BSRs should be well connected and be in the backbone portion of the network, as opposed to the dialup portion of the network. **Note** The Cisco IOS implementation of PIM BSR uses the value 0 as the default priority for candidate RPs and BSRs. This implementation predates the draft-ietf-pim-sm-bsr IETF draft, the first IETF draft to specify 192 as the default priority value. The Cisco IOS implementation, thus, deviates from the IETF draft. To comply with the default priority value specified in the draft, you must explicitly set the priority value to 192.

	Command or Action	Purpose
Step 9	**ip pim rp-candidate** *interface-type interface-number* [group-list *access-list*] [**interval** seconds] [**priority** *value*] **Example:** `Router(config)# ip pim rp-candidate ethernet 2 group-list 4 priority 192`	Configures the router to advertise itself as a PIM Version 2 candidate RP to the BSR. • Perform this step on the RP and BSR routers. • When an interval is specified, the candidate RP advertisement interval is set to the number of seconds specified. The default interval is 60 seconds. Tuning this interval down can reduce the time required to fail over to a secondary RP at the expense of generating more PIMv2 messages. • The Cisco IOS implementation of PIM BSR selects an RP from a set of candidate RPs using a method that is incompatible with the specification in RFC 2362. See the "BSR and RFC 2362 Interoperable Candidate RP: Example" section for a configuration workaround. See CSCdy56806 using the Cisco Bug Toolkit for more information. **Note** The Cisco IOS implementation of PIM BSR uses the value 0 as the default priority for candidate RPs and BSRs. This implementation predates the draft-ietf-pim-sm-bsr IETF draft, the first IETF draft to specify 192 as the default priority value. The Cisco IOS implementation, thus, deviates from the IETF draft. To comply with the default priority value specified in the draft, you must explicitly set the priority value to 192.
Step 10	**no ip pim dm-fallback** **Example:** `Router(config)# no ip pim dm-fallback`	(Optional) Prevents PIM dense mode fallback. • Configure this command on all routers in a PIM sparse-mode domain.
Step 11	Repeat Steps 8 through 10 on all RP and BSR routers.	—
Step 12	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 13	**ip pim bsr-border** **Example:** `Router(config-if)# ip pim bsr-border`	Prevents the bootstrap router (BSR) messages from being sent or received through an interface. • See the "BSR Border Interface" section for more information.
Step 14	**end** **Example:** `Router(config-if)# end`	Ends the current configuration session and returns to privileged EXEC mode.
Step 15	Repeat Steps 12 through 14 on all the routers that have boundary interfaces where the messages should not be sent or received.	—

	Command or Action	Purpose
Step 16	**show ip pim rp** [**mapping**] [*rp-address*] **Example:** `Router# show ip pim rp`	(Optional) Displays active rendezvous points (RPs) that are cached with associated multicast routing entries.
Step 17	**show ip pim rp-hash** [*group-address*] [*group-name*] **Example:** `Router# show ip pim rp-hash 239.1.1.1`	(Optional) Displays which rendezvous point (RP) is being selected for a specified group.
Step 18	**show ip pim bsr-router** **Example:** `Router# show ip pim bsr-router`	(Optional) Displays the bootstrap router (BSR) information.
Step 19	**show ip igmp groups** [*group-name* | *group-address* | *interface-type interface-number*] [**detail**] **Example:** `Router# show ip igmp groups`	(Optional) Displays the multicast groups having receivers that are directly connected to the router and that were learned through IGMP. • A receiver must be active on the network at the time that this command is issued in order for receiver information to be present on the resulting display.
Step 20	**show ip mroute** **Example:** `Router# show ip mroute cbone-audio`	(Optional) Displays the contents of the IP mroute table.

What to Do Next

Proceed to the "Verifying IP Multicast Operation" module.

Configuring Sparse Mode with a Single Static RP

A rendezvous point (RP) is required in networks running Protocol Independent Multicast sparse mode (PIM-SM). In PIM-SM, traffic will be forwarded only to network segments with active receivers that have explicitly requested multicast data.

This section describes how to configure sparse mode with a single static RP.

Static RP

If you are configuring PIM sparse mode, you must configure a PIM RP for a multicast group. An RP can either be configured statically in each device, or learned through a dynamic mechanism. This task explains how to statically configure an RP, as opposed to the router learning the RP through a dynamic mechanism such as Auto-RP.

PIM designated routers (DRs) forward data from directly connected multicast sources to the RP for distribution down the shared tree. Data is forwarded to the RP in one of two ways. It is encapsulated in register packets and unicast directly to the RP, or, if the RP has itself joined the source tree, it is multicast forwarded per the RPF forwarding algorithm. Last hop routers directly connected to receivers may, at their discretion, join themselves to the source tree and prune themselves from the shared tree.

A single RP can be configured for multiple groups that are defined by an access list. If no RP is configured for a group, the router treats the group as dense using the PIM dense mode techniques. (You can prevent this occurrence by configuring the **no ip pim dm-fallback** command.)

If a conflict exists between the RP configured with the **ip pim rp-address** command and one learned by Auto-RP, the Auto-RP information is used, unless the **override** keyword is configured.

Prerequisites

All access lists that are needed when sparse mode is configured with a single static RP should be configured prior to beginning the configuration task. For information about how to configure an access list, see the "Creating an IP Access List and Applying It to an Interface" module.

Restrictions

The same RP address cannot be used for both bidirectional and sparse mode PIM groups.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast-routing** [**distributed**]
4. **interface** *type number*
5. **ip pim sparse-mode**
6. Repeat Steps 1 through 5 on every interface that uses IP multicast.
7. **exit**
8. **ip pim rp-address** *rp-address* [*access-list*] [**override**]
9. **no ip pim dm-fallback**
10. **end**
11. **show ip pim rp** [**mapping**] [*rp-address*]
12. **show ip igmp groups** [*group-name* | *group-address* | *interface-type interface-number*] [**detail**]
13. **show ip mroute**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast-routing** [**distributed**] **Example:** `Router(config)# ip multicast-routing`	Enables IP multicast routing. • Use the **distributed** keyword to enable Multicast Distributed Switching.
Step 4	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 5	**ip pim sparse-mode** **Example:** `Router(config-if)# ip pim sparse-mode`	Enables PIM on an interface. You must use sparse mode.
Step 6	Repeat Steps 1 through 5 on every interface that uses IP multicast.	—
Step 7	**exit** **Example:** `Router(config-if)# exit`	Returns to global configuration mode.
Step 8	**ip pim rp-address** *rp-address* [*access-list*] [**override**] **Example:** `Router(config)# ip pim rp-address 192.168.0.0`	Configures the address of a PIM RP for a particular group. • Perform this step on any router. • The *access-list* argument specifies the number or name of an access list that defines for which multicast groups the RP should be used. • The **override** keyword specifies that if there is a conflict between the RP configured with this command and one learned by Auto-RP, the RP configured with this command prevails.
Step 9	**no ip pim dm-fallback** **Example:** `Router(config)# no ip pim dm-fallback`	(Optional) Prevents PIM dense mode fallback. • Configure this command on all routers in a PIM sparse-mode domain.

	Command or Action	Purpose
Step 10	**end** **Example:** `Router(config)# end`	Ends the current configuration session and returns to EXEC mode.
Step 11	**show ip pim rp** [**mapping**] [*rp-address*] **Example:** `Router# show ip pim rp mapping`	(Optional) Displays RPs known in the network and shows how the router learned about each RP.
Step 12	**show ip igmp groups** [*group-name* | *group-address* | *interface-type interface-number*] [**detail**] **Example:** `Router# show ip igmp groups`	(Optional) Displays the multicast groups having receivers that are directly connected to the router and that were learned through IGMP. • A receiver must be active on the network at the time that this command is issued in order for receiver information to be present on the resulting display.
Step 13	**show ip mroute** **Example:** `Router# show ip mroute`	(Optional) Displays the contents of the IP mroute table.

What to Do Next

Proceed to the "Verifying IP Multicast Operation" module.

Configuring Source Specific Multicast

This section contains information about and instructions on how to configure Source Specific Multicast (SSM). SSM is an extension of IP multicast where datagram traffic is forwarded to receivers from only those multicast sources that the receivers have explicitly joined. For multicast groups configured for SSM, only source-specific multicast distribution trees (not shared trees) are created.

SSM Components

SSM is a datagram delivery model that best supports one-to-many applications, also known as broadcast applications. SSM is a core networking technology for the Cisco implementation of IP multicast solutions targeted for audio and video broadcast application environments and is described in RFC 3569. The following two Cisco IOS components together support the implementation of SSM:

- Protocol Independent Multicast source-specific mode (PIM-SSM)
- Internet Group Management Protocol Version 3 (IGMPv3)

Protocol Independent Multicast (PIM) SSM, or PIM-SSM, is the routing protocol that supports the implementation of SSM and is derived from PIM sparse mode (PIM-SM). IGMP is the Internet Engineering Task Force (IETF) standards track protocol used for hosts to signal multicast group membership to routers. IGMP Version 3 supports source filtering, which is required for SSM. IGMP For SSM to run with IGMPv3, SSM must be supported in the Cisco IOS router, the host where the application is running, and the application itself.

How SSM Differs from Internet Standard Multicast

The standard IP multicast infrastructure in the Internet and many enterprise intranets is based on the PIM-SM protocol and Multicast Source Discovery Protocol (MSDP). These protocols have proved to be reliable, extensive, and efficient. However, they are bound to the complexity and functionality limitations of the Internet Standard Multicast (ISM) service model. For example, with ISM, the network must maintain knowledge about which hosts in the network are actively sending multicast traffic. With SSM, this information is provided by receivers through the source addresses relayed to the last-hop routers by IGMPv3. SSM is an incremental response to the issues associated with ISM and is intended to coexist in the network with the protocols developed for ISM. In general, SSM provides IP multicast service for applications that utilize SSM.

ISM service is described in RFC 1112. This service consists of the delivery of IP datagrams from any source to a group of receivers called the multicast host group. The datagram traffic for the multicast host group consists of datagrams with an arbitrary IP unicast source address S and the multicast group address G as the IP destination address. Systems will receive this traffic by becoming members of the host group. Membership in a host group simply requires signaling the host group through IGMP Version 1, 2, or 3.

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 will receive this traffic by becoming members of the (S, G) channel. In both SSM and ISM, no signaling is required to become a source. However, in SSM, receivers must subscribe or unsubscribe to (S, G) channels to receive or not receive traffic from specific sources. In other words, receivers can receive traffic only from (S, G) channels to which they are subscribed, whereas in ISM, receivers need not know the IP addresses of sources from which they receive their traffic. The proposed standard approach for channel subscription signaling utilizes IGMP INCLUDE mode membership reports, which are supported only in IGMP Version 3.

SSM can coexist with the ISM service by applying the SSM delivery model to a configured subset of the IP multicast group address range. The Internet Assigned Numbers Authority (IANA) has reserved the address range from 232.0.0.0 through 232.255.255.255 for SSM applications and protocols. Cisco IOS software allows SSM configuration for an arbitrary subset of the IP multicast address range from 224.0.0.0 through 239.255.255.255. When an SSM range is defined, an existing IP multicast receiver application will not receive any traffic when it tries to use addresses in the SSM range unless the application is modified to use explicit (S, G) channel subscription or is SSM-enabled through a URL Rendezvous Directory (URD).

SSM Operations

An established network in which IP multicast service is based on PIM-SM can support SSM services. SSM can also be deployed alone in a network without the full range of protocols that are required for interdomain PIM-SM. That is, SSM does not require an RP, so there is no need for an RP mechanism such as Auto-RP, MSDP, or bootstrap router (BSR).

If SSM is deployed in a network that is already configured for PIM-SM, then only the last-hop routers must be upgraded to a Cisco IOS software image that supports SSM. Routers that are not directly connected to receivers do not have to upgrade to a Cisco IOS software image that supports SSM. In general, these non-last-hop routers must only run PIM-SM in the SSM range. They may need additional access control configuration to suppress MSDP signaling, registering, or PIM-SM shared-tree operations from occurring within the SSM range.

Effects of SSM

The SSM mode of operation is enabled by configuring the SSM range using the **ip pim ssm** global configuration command. This configuration has the following effects:

- For groups within the SSM range, (S, G) channel subscriptions are accepted through IGMPv3 INCLUDE mode membership reports.
- PIM operations within the SSM range of addresses change to PIM-SSM, a mode derived from PIM-SM. In this mode, only PIM (S, G) Join and Prune messages are generated by the router. Incoming messages related to rendezvous point tree (RPT) operations are ignored or rejected, and incoming PIM register messages are immediately answered with Register-Stop messages. PIM-SSM is backward-compatible with PIM-SM unless a router is a last-hop router. Therefore, routers that are not last-hop routers can run PIM-SM for SSM groups (for example, if they do not yet support SSM).
- For groups within the SSM range, no MSDP Source-Active (SA) messages within the SSM range will be accepted, generated, or forwarded.

IGMPv3 Host Signaling

IGMPv3 is the third version of the IETF standards track protocol in which hosts signal membership to last-hop routers of multicast groups. IGMPv3 introduces the ability for hosts to signal group membership that allows filtering capabilities with respect to sources. A host can signal either that it wants to receive traffic from all sources sending to a group except for some specific sources (a mode called EXCLUDE) or that it wants to receive traffic only from some specific sources sending to the group (a mode called INCLUDE).

IGMPv3 can operate with both ISM and SSM. In ISM, both EXCLUDE and INCLUDE mode reports are accepted by the last-hop router. In SSM, only INCLUDE mode reports are accepted by the last-hop router.

Benefits of Source Specific Multicast

IP Multicast Address Management

In the ISM service, applications must acquire a unique IP multicast group address because traffic distribution is based only on the IP multicast group address used. If two applications with different sources and receivers use the same IP multicast group address, receivers of both applications will receive traffic from the senders of both applications. Even though the receivers, if programmed appropriately, can filter out the unwanted traffic, this situation would cause generally unacceptable levels of unwanted traffic.

Allocating a unique IP multicast group address for an application is problematic. Most short-lived applications use mechanisms like Session Description Protocol (SDP) and Session Announcement Protocol (SAP) to get a random address, a solution that does not work well with a rising number of applications in the Internet. The best current solution for long-lived applications is described in RFC 2770, but this solution suffers from the restriction that each autonomous system is limited to only 255 usable IP multicast addresses.

In SSM, traffic from each source is forwarded among routers in the network independently of traffic from other sources. Thus different sources can reuse multicast group addresses in the SSM range.

Inhibition of Denial of Service Attacks

In SSM, multicast traffic from each individual source will be transported across the network only if it was requested (through IGMPv3 or URD memberships) from a receiver. In contrast, ISM forwards traffic from any active source sending to a multicast group to all receivers requesting that multicast group. In

Internet broadcast applications, this ISM behavior is highly undesirable because it allows unwanted sources to easily disturb the actual Internet broadcast source by simply sending traffic to the same multicast group. This situation depletes bandwidth at the receiver side with unwanted traffic and thus disrupts the undisturbed reception of the Internet broadcast. In SSM, this type of denial-of-service (DoS) attack cannot be made by simply sending traffic to a multicast group.

Installation and Management

SSM is easy to install and provision in a network because it does not require the network to maintain information about which active sources are sending to multicast groups. This requirement exists in ISM (with IGMPv1, IGMPv2, or IGMPv3).

The current standard solutions for ISM service are PIM-SM and MSDP. Rendezvous point (RP) management in PIM-SM (including the necessity for Auto-RP or BSR) and MSDP is required only for the network to learn about active sources. This management is not necessary in SSM. SSM is therefore easier than ISM to install and manage and easier to operationally scale in deployment. Another factor that contributes to the ease of installation of SSM is the fact that it can leverage preexisting PIM-SM networks.

Internet Broadcast Applications

The three benefits listed above make SSM ideal for Internet broadcast-style applications for the following reasons:

- The ability to provide Internet broadcast services through SSM without the need for unique IP multicast addresses allows content providers to easily offer their service. IP multicast address allocation has been a serious problem for content providers in the past.
- The prevention of DoS attacks is an important factor for Internet broadcast services because, with their exposure to a large number of receivers, they are the most common targets for such attacks.
- The ease of installation and operation of SSM makes it ideal for network operators, especially in those cases where content needs to be forwarded between multiple independent PIM domains (because there is no need to manage MSDP for SSM between PIM domains).

Prerequisites

If you want to use an access list to define the SSM range, configure the access list before you reference the access list in the ip pim ssm command. For information about how to configure an access list, see the "Creating an IP Access List and Applying It to an Interface" module.

Restrictions

Address Management Restrictions

Address management is necessary to some degree when SSM is used with Layer 2 switching mechanisms. Cisco Group Management Protocol (CGMP), IGMP snooping, and Router-Port Group Management Protocol (RGMP) currently support only group-specific filtering, not (S, G) channel-specific filtering. If different receivers in a switched network request different (S, G) channels sharing the same group, they will not benefit from these existing mechanisms. Instead, both receivers will receive all (S, G) channel traffic (and filter out the unwanted traffic on input). Because SSM can reuse the group addresses in the SSM range for many independent applications, this situation can lead to unexpected traffic filtering in a switched network. It is therefore important to follow the recommendations set forth in the IETF drafts for SSM in regard to using random IP addresses in the SSM range to minimize the chance for reuse of a single address by different applications. For example, an application service providing a set of television channels should, even with SSM, use a different group

for each television (S, G) channel. This setup guarantees that multiple receivers to different channels within the same application service will never experience traffic aliasing in networks that include Layer 2 switches.

IGMP Snooping and CGMP Limitations

IGMPv3 uses new membership report messages that may not be recognized correctly by older IGMP snooping switches, in which case hosts will not receive traffic properly. IGMP uses a new link-local address for the destination of these messages. This new link-local address is 224.0.0.22.

State Maintenance Limitations

In PIM-SSM, the last-hop router will continue to send (S, G) Join messages periodically if appropriate (S, G) subscriptions are on the interfaces. Therefore, as long as receivers send (S, G) subscriptions, the shortest path tree (SPT) state from the receivers to the source will be maintained, even if the source is not sending traffic for longer periods of time (or never sends).

This case is opposite to that of PIM-SM, in which the (S, G) state is maintained only if the source is sending traffic and receivers are joining the group. If a source stops sending traffic for more than 3 minutes in PIM-SM, the (S, G) state is deleted and will be reestablished only after packets from the source arrive again through the RPT. Because no mechanism in PIM-SSM notifies a receiver that a source is active, the network must maintain the (S, G) state in PIM-SSM as long as receivers are requesting receipt of that channel.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast-routing** [**distributed**]
4. **ip pim ssm** {**default** | **range** *access-list*}
5. **interface** *type number*
6. **ip pim sparse-mode**
7. Repeat Steps 1 through 6 on every interface that uses IP multicast.
8. **ip igmp version 3**
9. Repeat Step 8 on all host-facing interfaces.
10. **end**
11. **show ip igmp groups** [*group-name* | *group-address* | *interface-type interface-number*] [**detail**]
12. **show ip mroute**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast-routing** [**distributed**] **Example:** `Router(config)# ip multicast-routing`	Enables IP multicast routing. • Use the **distributed** keyword to enable Multicast Distributed Switching.
Step 4	**ip pim ssm** {**default** | **range** *access-list*} **Example:** `Router(config)# ip pim ssm default`	Configures SSM service. • The **default** keyword defines the SSM range access list as 232/8. • The **range** keyword specifies the standard IP access list number or name that defines the SSM range.
Step 5	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Selects an interface that is connected to hosts on which IGMPv3 can be enabled.
Step 6	**ip pim sparse-mode** **Example:** `Router(config-if)# ip pim sparse-mode`	Enables PIM on an interface. You must use sparse mode.
Step 7	Repeat Steps 1 through 6 on every interface that uses IP multicast.	—
Step 8	**ip igmp version 3** **Example:** `Router(config-if)# ip igmp version 3`	Enables IGMPv3 on this interface. The default version of IGMP is set to Version 2. Version 3 is required by SSM.
Step 9	Repeat Step 8 on all host-facing interfaces.	—
Step 10	**end** **Example:** `Router(config-if)# end`	Ends the current configuration session and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 11	**show ip igmp groups** [*group-name* | *group-address* | *interface-type interface-number*] [**detail**] **Example:** `Router# show ip igmp groups`	(Optional) Displays the multicast groups having receivers that are directly connected to the router and that were learned through IGMP. • A receiver must be active on the network at the time that this command is issued in order for receiver information to be present on the resulting display.
Step 12	**show ip mroute** **Example:** `Router# show ip mroute`	(Optional) Displays the contents of the IP mroute table. • This command displays whether a multicast group is configured for SSM service or a source-specific host report has been received.

What to Do Next

Proceed to the "Verifying IP Multicast Operation" module.

Configuring Bidirectional PIM

This section describes how to configure bidirectional PIM (bidir-PIM). Bidir-PIM shares many of its shortest path tree (SPT) operations with PIM-SM. Bidir-PIM also has unconditional forwarding of source traffic toward the RP upstream on the shared tree, but has no registering process for sources as in PIM-SM. These modifications allow forwarding of traffic in all routers based solely on the (*, G) multicast routing entries. This form of forwarding eliminates any source-specific state and allows scaling capability to an arbitrary number of sources.

Benefits of Bidirectional PIM

- Bidir-PIM removes the performance cost of maintaining a routing state table for a large number of sources.
- Bidir-PIM is designed to be used for many-to-many applications within individual PIM domains. Multicast groups in bidirectional PIM mode can scale to an arbitrary number of sources without incurring overhead due to the number of sources.

Multicast Group Modes

In PIM, packet traffic for a multicast group is routed according to the rules of the mode configured for that multicast group. The Cisco IOS implementation of PIM supports four modes for a multicast group:

- PIM bidirectional mode
- PIM dense mode
- PIM sparse mode
- PIM Source Specific Mode (SSM)

A router can simultaneously support all four modes or any combination of them for different multicast groups.

Bidirectional Shared Tree

In bidirectional mode, traffic is routed only along a bidirectional shared tree that is rooted at the rendezvous point (RP) for the group. In bidir-PIM, 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 need not be a router, but can be any unassigned IP address on a network that is reachable throughout the PIM domain. This technique is the preferred configuration method for establishing a redundant RP configuration for bidir-PIM.

Membership in a bidirectional group is signaled by way of explicit Join messages. Traffic from sources is unconditionally sent up the shared tree toward the RP and passed down the tree toward the receivers on each branch of the tree.

Figure 3 and Figure 4 show the difference in state created per router for a unidirectional shared tree and source tree versus a bidirectional shared tree.

Figure 3 Unidirectional Shared Tree and Source Tree

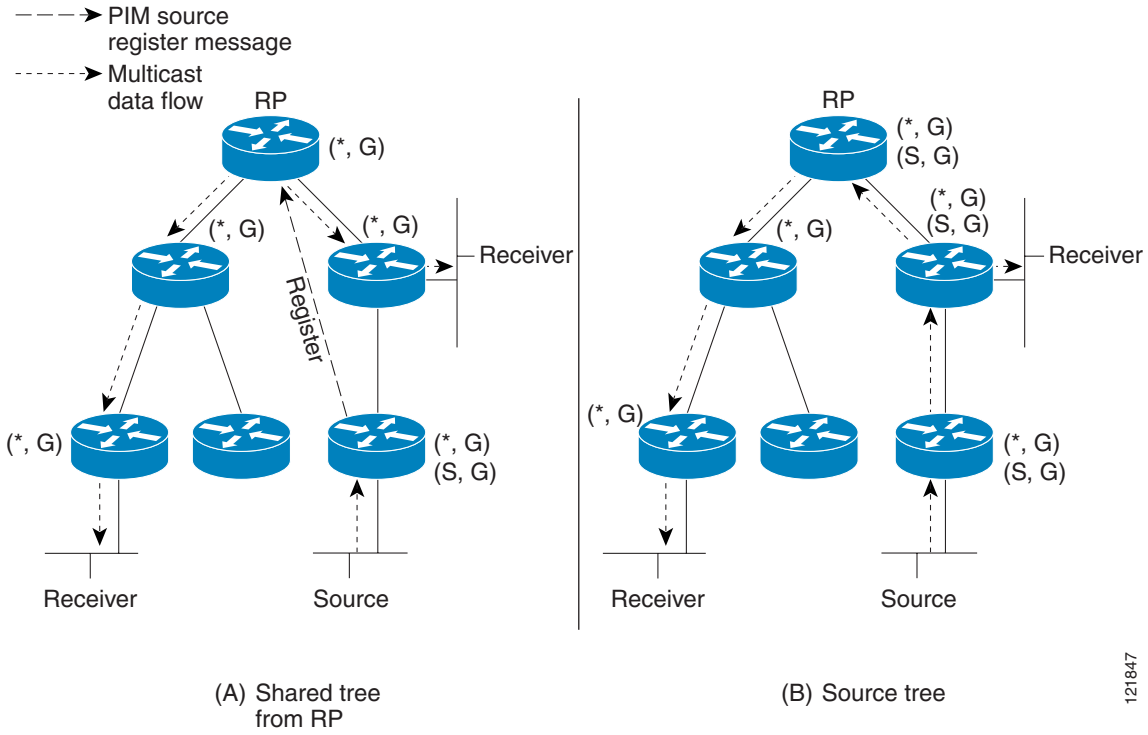
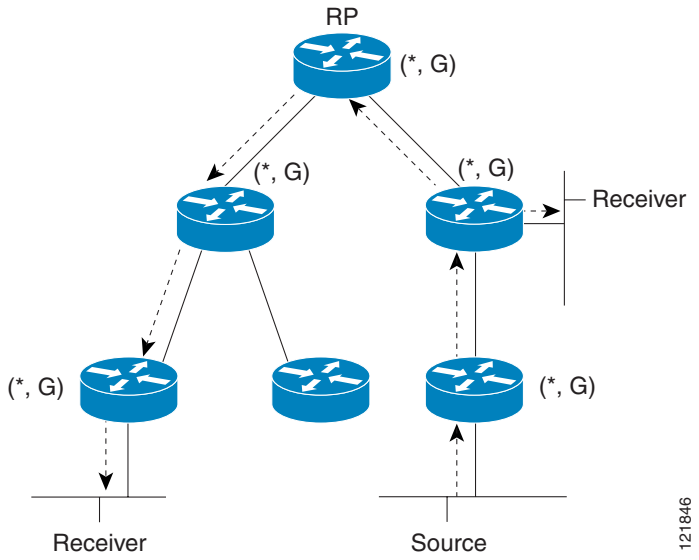


Figure 4 Bidirectional Shared Tree



For packets that are forwarded downstream from the RP toward receivers, there are no fundamental differences between bidir-PIM and PIM-SM. Bidir-PIM deviates substantially from PIM-SM for traffic that is passed from sources upstream toward the RP.

PIM-SM cannot forward traffic in the upstream direction of a tree because it accepts traffic from only one Reverse Path Forwarding (RPF) interface. This interface (for the shared tree) points toward the RP, thus allowing only downstream traffic flow. Upstream traffic is first encapsulated into unicast register messages, which are passed from the designated router (DR) of the source toward the RP. Second, the RP joins an SPT that is rooted at the source. Therefore, in PIM-SM, traffic from sources destined for the RP does not flow upstream in the shared tree, but downstream along the SPT of the source until it reaches the RP. From the RP, traffic flows along the shared tree toward all receivers.

In bidir-PIM, the packet-forwarding rules have been improved over PIM-SM, allowing traffic to be passed up the shared tree toward the RP. To avoid multicast packet looping, bidir-PIM introduces a new mechanism called designated forwarder (DF) election, which establishes a loop-free SPT rooted at the RP.

DF Election

On every network segment and point-to-point link, all PIM routers participate in a procedure called designated forwarder (DF) election. The procedure selects one router as the DF for every RP of bidirectional groups. This router is responsible for forwarding multicast packets received on that network.

The DF election is based on unicast routing metrics and uses the same tie-breaking rules employed by PIM assert processes. The router with the most preferred unicast routing metric to the RP becomes the DF. Use of this method ensures that only one copy of every packet will be sent to the RP, even if there are parallel equal-cost paths to the RP.

A DF is selected for every RP of bidirectional groups. As a result, multiple routers may be elected as DF on any network segment, one for each RP. Any particular router may be elected as DF on more than one interface.

Bidirectional Group Tree Building

The procedure for joining the shared tree of a bidirectional group is almost identical to that used in PIM-SM. One main difference is that, for bidirectional groups, the role of the DR is assumed by the DF for the RP.

On a network that has local receivers, only the router elected as the DF populates the outgoing interface list (olist) upon receiving Internet Group Management Protocol (IGMP) Join messages, and sends (*, G) Join and Leave messages upstream toward the RP. When a downstream router wishes to join the shared tree, the RPF neighbor in the PIM Join and Leave messages is always the DF elected for the interface that lead to the RP.

When a router receives a Join or Leave message, and the router is not the DF for the receiving interface, the message is ignored. Otherwise, the router updates the shared tree in the same way as in sparse mode.

In a network where all routers support bidirectional shared trees, (S, G) Join and Leave messages are ignored. There is also no need to send PIM assert messages because the DF election procedure eliminates parallel downstream paths from any RP. An RP never joins a path back to the source, nor will it send any register stops.

Packet Forwarding

A router creates (*, G) entries only for bidirectional groups. The olist of a (*, G) entry includes all the interfaces for which the router has been elected DF and that have received either an IGMP or PIM Join message. If a router is located on a sender-only branch, it will also create a (*, G) state, but the olist will not include any interfaces.

If a packet is received from the RPF interface toward the RP, the packet is forwarded downstream according to the olist of the (*, G) entry. Otherwise, only the router that is the DF for the receiving interface forwards the packet upstream toward the RP; all other routers must discard the packet.

Prerequisites

All access lists needed when configuring bidirectional PIM must be configured prior to beginning the configuration task. For information about how to configure an access list, see the "Creating an IP Access List and Applying It to an Interface" module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast-routing** [**distributed**]
4. **interface** *type number*
5. **ip pim sparse-mode**
6. **exit**
7. **ip pim bidir-enable**
8. **ip pim rp-address** *rp-address* [*access-list*] [**override**] [**bidir**]
9. **end**
10. Repeat Steps 2 through 9 on every multicast-enabled interface on every router.
11. **show ip pim rp** [mapping] [*rp-address*]

12. **show ip mroute**
13. **show ip pim interface** [*type number*] [df | count] [*rp-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast-routing** [**distributed**] **Example:** `Router(config)# ip multicast-routing`	Enables IP multicast routing. • Use the **distributed** keyword to enable Multicast Distributed Switching.
Step 4	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 5	**ip pim sparse-mode** **Example:** `Router(config-if)# ip pim sparse-mode`	Enables sparse mode.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Returns to global configuration mode.
Step 7	**ip pim bidir-enable** **Example:** `Router(config)# ip pim bidir-enable`	Enables bidir-PIM on a router. • Perform this step on every router.
Step 8	**ip pim rp-address** *rp-address* [*access-list*] [override] [bidir] **Example:** `Router(config)# ip pim rp-address 10.0.1.1 45 bidir`	Configures the address of a PIM RP for a particular group. • Perform this step on every router. • This command defines the RP as bidirectional and defines the bidirectional group by way of the access list.
Step 9	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 10	Repeat Steps 2 through 9 on every multicast-enabled interface on every router.	—

	Command or Action	Purpose
Step 11	**show ip pim rp** [**mapping**] [rp-address] **Example:** `Router# show ip pim rp`	(Optional) Displays active RPs that are cached with associated multicast routing entries.
Step 12	`show ip mroute` **Example:** `Router# show ip mroute`	(Optional) Displays the contents of the IP mroute table.
Step 13	**show ip pim interface** [*type number*] [**df** | **count**] [*rp-address*] **Example:** `Router# show ip pim interface`	(Optional) Displays information about the elected DF for each RP of an interface, along with the unicast routing metric associated with the DF.

Configuration Examples for Basic IP Multicast

This section contains the following examples:

Sparse Mode with Auto-RP: Example

The following example configures sparse mode with Auto-RP:

```
ip multicast-routing
ip pim autorp listener
ip pim send-rp-announce Loopback0 scope 16 group-list 1
ip pim send-rp-discovery Loopback1 scope 16
no ip pim dm-fallback
access-list 1 permit 239.254.2.0 0.0.0.255
access-list 1 permit 239.254.3.0 0.0.0.255
.
.
.
```

```
access-list 10 permit 224.0.1.39
access-list 10 permit 224.0.1.40
access-list 10 permit 239.254.2.0 0.0.0.255
access-list 10 permit 239.254.3.0 0.0.0.255
```

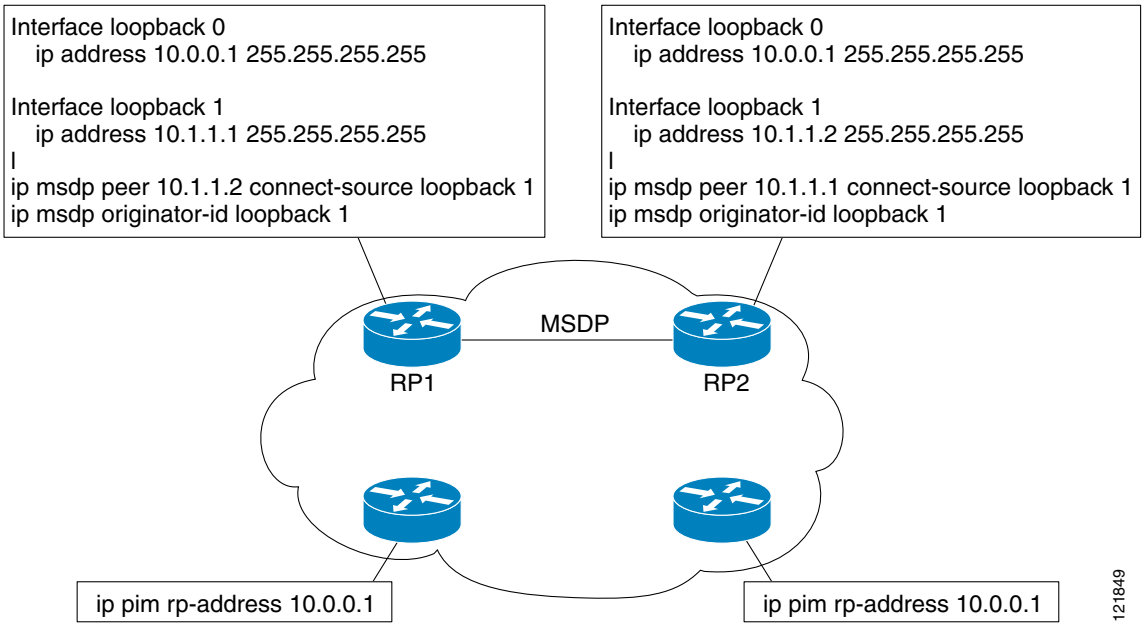
Sparse Mode with Anycast RP: Example

The main purpose of an Anycast RP implementation is that the downstream multicast routers will have just one address for an RP. The example given in Figure 5 shows how loopback interface 0 of the RPs (RP1 and RP2) is configured with the 10.0.0.1 IP address. If this 10.0.0.1 address is configured on all RPs as the address for loopback interface 0 and then configured as the RP address, IP routing will converge on the closest RP. This address must be a host route; note the 255.255.255.255 subnet mask.

The downstream routers must be informed about the 10.0.0.1 RP address. In Figure 5, the routers are configured statically with the **ip pim rp-address 10.0.0.1** global configuration command. This configuration could also be accomplished using the Auto-RP or bootstrap router (BSR) features.

The RPs in Figure 5 must also share source information using MSDP. In this example, loopback interface 1 of the RPs (RP1 and RP2) is configured for MSDP peering. The MSDP peering address must be different from the anycast RP address.

Figure 5 AnyCast RP Configuration



Many routing protocols choose the highest IP address on loopback interfaces for the router ID. A problem may arise if the router selects the anycast RP address for the router ID. It is recommended that you avoid this problem by manually setting the router ID on the RPs to the same address as the MSDP peering address (for example, the loopback 1 address in Figure 5). In Open Shortest Path First (OSPF), the router ID is configured using the **router-id** router configuration command. In Border Gateway Protocol (BGP), the router ID is configured using the **bgp router-id** router configuration command. In

many BGP topologies, the MSDP peering address and the BGP peering address must be the same in order to pass the RPF check. The BGP peering address can be set using the **neighbor update-source** router configuration command.

The anycast RP example above uses IP addresses taken from RFC 1918. These IP addresses are normally blocked at interdomain borders and therefore are not accessible to other ISPs. You must use valid IP addresses if you want the RPs to be reachable from other domains.

The following example shows how to perform an Anycast RP configuration.

On RP 1

```
ip pim rp-address 10.0.0.1
no ip pim dm-fallback
interface loopback 0
 ip address 10.0.0.1 255.255.255.255
!
interface loopback 1
 ip address 10.1.1.1. 255.255.255.255
!
 ip msdp peer 10.1.1.2 connect-source loopback 1
 ip msdp originator-id loopback 1
```

On RP 2

```
ip pim rp-address 10.0.0.1
no ip pim dm-fallback
interface loopback 0
 ip address 10.0.0.1 255.255.255.255

interface loopback 1
 ip address 10.1.1.2. 255.255.255.255
!
 ip msdp peer 10.1.1.1 connect-source loopback 1
 ip msdp originator-id loopback 1
```

All Other Routers

```
ip pim rp-address 10.0.0.1
no ip pim dm-fallback
```

Sparse Mode with Bootstrap Router: Example

The following example is a configuration for a candidate BSR, which also happens to be a candidate RP:

```
!
ip multicast-routing
!
interface Ethernet0
 ip address 172.69.62.35 255.255.255.240
 ip pim sparse-mode
!
interface Ethernet1
 ip address 172.21.24.18 255.255.255.248
 ip pim sparse-mode
!
interface Ethernet2
 ip address 172.21.24.12 255.255.255.248
 ip pim sparse-mode
!
```

```
ip pim bsr-candidate Ethernet2 30 10
ip pim rp-candidate Ethernet2 group-list 5
access-list 5 permit 239.255.2.0 0.0.0.255
no ip pim dm-fallback
```

BSR and RFC 2362 Interoperable Candidate RP: Example

When Cisco and non-Cisco routers are being operated in a single PIM domain with PIM Version 2 BSR, care must be taken when configuring candidate RPs because the Cisco IOS implementation of the BSR RP selection is not fully compatible with RFC 2362.

RFC 2362 specifies that the BSR RP be selected as follows (RFC 2362, 3.7):

1. Select the candidate RP with the highest priority (lowest configured priority value).
2. If there is a tie in the priority level, select the candidate RP with the highest hash function value.
3. If there is a tie in the hash function value, select the candidate RP with the highest IP address.

Cisco routers always select the candidate RP based on the longest match on the announced group address prefix before selecting an RP based on priority, hash function, or IP address.

Inconsistent candidate RP selection between Cisco and non-Cisco RFC 2362-compliant routers in the same domain if multiple candidate RPs with partially overlapping group address ranges are configured can occur. Inconsistent candidate RP selection can prevent connectivity between sources and receivers in the PIM domain. A source may register with one candidate RP and a receiver may connect to a different candidate RP even though it is in the same group.

The following example shows a configuration that can cause inconsistent RP selection between a Cisco and a non-Cisco router in a single PIM domain with PIM Version 2 BSR:

```
access-list 10 permit 224.0.0.0  7.255.255.255
ip pim rp-candidate ethernet1 group-list 10 priority 20

access-list 20 permit 224.0.0.0 15.255.255.255
ip pim rp-candidate ethernet2 group-list 20 priority 10
```

In this example, a candidate RP on Ethernet interface 1 announces a longer group prefix of 224.0.0.0/5 with a lower priority of 20. The candidate RP on Ethernet interface 2 announces a shorter group prefix of 224.0.0.0/4 with a higher priority of 10. For all groups that match both ranges a Cisco router will always select the candidate RP on Ethernet interface 1 because it has the longer announced group prefix. A non-Cisco fully RFC 2362-compliant router will always select the candidate RP on Ethernet interface 2 because it is configured with a higher priority.

To avoid this interoperability issue, do not configure different candidate RPs to announce partially overlapping group address prefixes. Configure any group prefixes that you want to announce from more than one candidate RP with the same group prefix length.

The following example shows how to configure the previous example so that there is no incompatibility between a Cisco router and a non-Cisco router in a single PIM domain with PIM Version 2 BSR:

```
access-list 10 permit 224.0.0.0  7.255.255.255
ip pim rp-candidate ethernet1 group-list 10 priority 20

access-list 20 permit 224.0.0.0  7.255.255.255
access-list 20 permit 232.0.0.0  7.255.255.255
ip pim rp-candidate ethernet2 group-list 20 priority 10
```

In this configuration the candidate RP on Ethernet interface 2 announces group address 224.0.0.0/5 and 232.0.0.0/5 which equal 224.0.0.0/4, but gives the interface the same group prefix length (5) as the candidate RP on Ethernet 1. As a result, both a Cisco router and an RFC 2362-compliant router will select the RP Ethernet interface 2.

Sparse Mode with a Single Static RP: Example

The following example sets the PIM RP address to 192.168.1.1 for all multicast groups and defines all groups to operate in sparse mode:

```
ip multicast-routing
interface ethernet 1
 ip pim sparse-mode
ip pim rp-address 192.168.1.1
no ip pim dm-fallback
```

Note The same RP cannot be used for both bidirectional and sparse mode groups.

The following example sets the PIM RP address to 172.16.1.1 for the multicast group 225.2.2.2 only:

```
access list 1 225.2.2.2 0.0.0.0
 ip pim rp-address 172.17.1.1
```

SSM with IGMPv3: Example

The following example shows how to configure a router (running IGMPv3) for SSM:

```
ip multicast-routing
!
interface Ethernet3/1
 ip address 172.21.200.203 255.255.255.0
 description backbone interface
    ip pim sparse-mode
!
interface Ethernet3/2
    ip address 131.108.1.2 255.255.255.0
    ip pim sparse-mode
    description ethernet connected to hosts
    ip igmp version 3
!
ip pim ssm default
```

SSM Filtering: Example

The following example shows how to configure filtering on legacy RP routers running Cisco IOS software releases that do not support SSM routing. This filtering will suppress all unwanted PIM-SM and MSDP traffic in the SSM range. Without this filtering, SSM will still operate, but there may be additional RPT traffic if legacy first-hop and last-hop routers exist in the network.

```
ip access-list extended no-ssm-range
 deny ip any 232.0.0.0 0.255.255.255
 permit ip any any
 ! Deny sources registering in SSM range
ip pim accept-register list no-ssm-range
ip access-list extended msdp-nono-list
 deny ip any 232.0.0.0 0.255.255.255
```

```
 permit ip any any
 ! Filter generated SA messages in SSM range. This configuration is needed only if there
 ! are sources directly connected to this router. The ip pim accept-register command
 ! filters remote sources. See http://www.cisco.com/warp/public/105/49.html for other SA
 ! messages that typically need to be filtered.
ip msdp redistribute list msdp-nono-list
 ! Filter received SA messages in SSM range. "Filtered on receipt" means messages are
 ! neither processed nor forwarded. This filter needs to be configured for each MSDP peer.
ip msdp sa-filter in msdp-peer1 list msdp-nono-list
.
.
.
ip msdp sa-filter in msdp-peerN list msdp-nono-list
```

Bidir-PIM: Example

By default, a bidirectional RP advertises all groups as bidirectional. An access list on the RP can be used to specify a list of groups to be advertised as bidirectional. Groups with the **deny** keyword will operate in dense mode. A different, nonbidirectional RP address is required for groups that operate in sparse mode because a single access list only allows either a **permit** or **deny** keyword.

The following example shows how to configure an RP for both sparse mode and bidirectional mode groups. The groups identified as 224/8 and 227/8 are bidirectional groups, and 226/8 is a sparse mode group. The RP must be configured to use different IP addresses for the sparse mode and bidirectional mode operations. Two loopback interfaces are used to allow this configuration. The addresses of these loopback interfaces must be routed throughout the PIM domain in such a way that the other routers in the PIM domain can communicate with the RP.

```
ip multicast-routing
!
.
.
.
!
interface loopback 0
 description One loopback address for this router's Bidir Mode RP function
 ip address 10.0.1.1 255.255.255.0
!
interface loopback 1
description One loopback address for this router's Sparse Mode RP function
 ip address 10.0.2.1 255.255.255.0
!
.
.
.
!
ip pim bidir-enable
ip pim rp-address 10.0.1.1 45 bidir
ip pim rp-address 10.0.2.1 46 bidir
!
access-list 45 permit 224.0.0.0 0.255.255.255
access-list 45 permit 227.0.0.0 0.255.255.255
access-list 46 permit 226.0.0.0 0.255.255.255
```

Additional References

The following sections provide references related to configuring basic IP multicast tasks.

Related Documents

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

Standards

Standard	Title
draft-kouvelas-pim-bidir-new-00.txt	*A New Proposal for Bi-directional PIM*

MIBs

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

RFCs

RFC	Title
RFC 1112	*Host Extensions for IP Multicasting*
RFC 1918	*Address Allocation for Private Internets*
RFC 2770	*GLOP Addressing in 233/8*
RFC 3569	*An Overview of Source-Specific Multicast (SSM)*

Technical Assistance

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

Feature Information for Configuring Basic IP Multicast

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Releases 12.2(1) or 12.0(3)S or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the "IP Multicast Features Roadmap."

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Configuring Basic IP Multicast

Feature Name	Releases	Feature Information
PIM Dense Mode Fallback Prevention in a Network Following RP Information Loss	12.3(4)T 12.0(28)S 12.2(33)SRA 12.2(33)SXH	The PIM Dense Mode Fallback Prevention in a Network Following RP Information Loss feature enables you to prevent PIM-DM fallback when all RPs fail. Preventing the use of dense mode is very important to multicast networks whose reliability is critical. This feature provides a mechanism to keep the multicast groups in sparse mode. This feature also allows you to block multicast traffic for groups not specifically configured. The following sections provide information about this feature: • Configuring Sparse Mode with Auto-RP, page 2 • Configuring Sparse Mode with Anycast RP, page 8 • Configuring Sparse Mode with a Bootstrap Router, page 12 • Configuring Sparse Mode with a Single Static RP, page 16 The following command was introduced by this feature: **ip pim dm-fallback**.
PIM Dense Mode Fallback Prevention in a Network Following RP Information Loss	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Using MSDP to Interconnect Multiple PIM-SM Domains

First Published: August 21, 2007
Last Updated: August 21, 2007

This module describes the tasks associated with using Multicast Source Discovery Protocol (MSDP) to interconnect multiple PIM-SM domains. The tasks explain how to configure MSDP peers, mesh groups, and default peers, how to use filters to control and scope MSDP activity, and how to monitor and maintain MSDP. Using MSDP with PIM-SM greatly reduces the complexity of connecting multiple PIM-SM domains.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Using MSDP to Interconnect Multiple PIM-SM Domains" section on page 49.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Using MSDP to Interconnect Multiple PIM-SM Domains

Before configuring MSDP, the addresses of all MSDP peers must be known in Border Gateway Protocol (BGP).

Information About Using MSDP to Interconnect Multiple PIM-SM Domains

To use MSDP to interconnect multiple PIM-SM domains, you should understand the following concepts:

Benefits of Using MSDP to Interconnect Multiple PIM-SM Domains

- Allows a rendezvous points (RP) to dynamically discover active sources outside of its domain.
- Introduces a more manageable approach for building multicast distribution trees between multiple domains.

Use of MSDP to Interconnect Multiple PIM-SM Domains

MSDP is a mechanism to connect multiple PIM-SM domains. The purpose of MSDP is to discover multicast sources in other PIM domains. The main advantage of MSDP is that it reduces the complexity of interconnecting multiple PIM-SM domains by allowing PIM-SM domains to use an interdomain source tree (rather than an common shared tree). When MSDP is configured in a network, RPs exchange source information with RPs in other domains. An RP can join the interdomain source tree for sources that are sending to groups for which it has receivers. The RP can do that because it is the root of the shared tree within its domain, which has branches to all points in the domain where there are active receivers. When a last-hop router learns of a new source outside the PIM-SM domain (through the arrival of a multicast packet from the source down the shared tree), it then can send a join toward the source and join the interdomain source tree.

Note If the RP either has no shared tree for a particular group or a shared tree whose outgoing interface list is null, it does not send a join to the source in another domain.

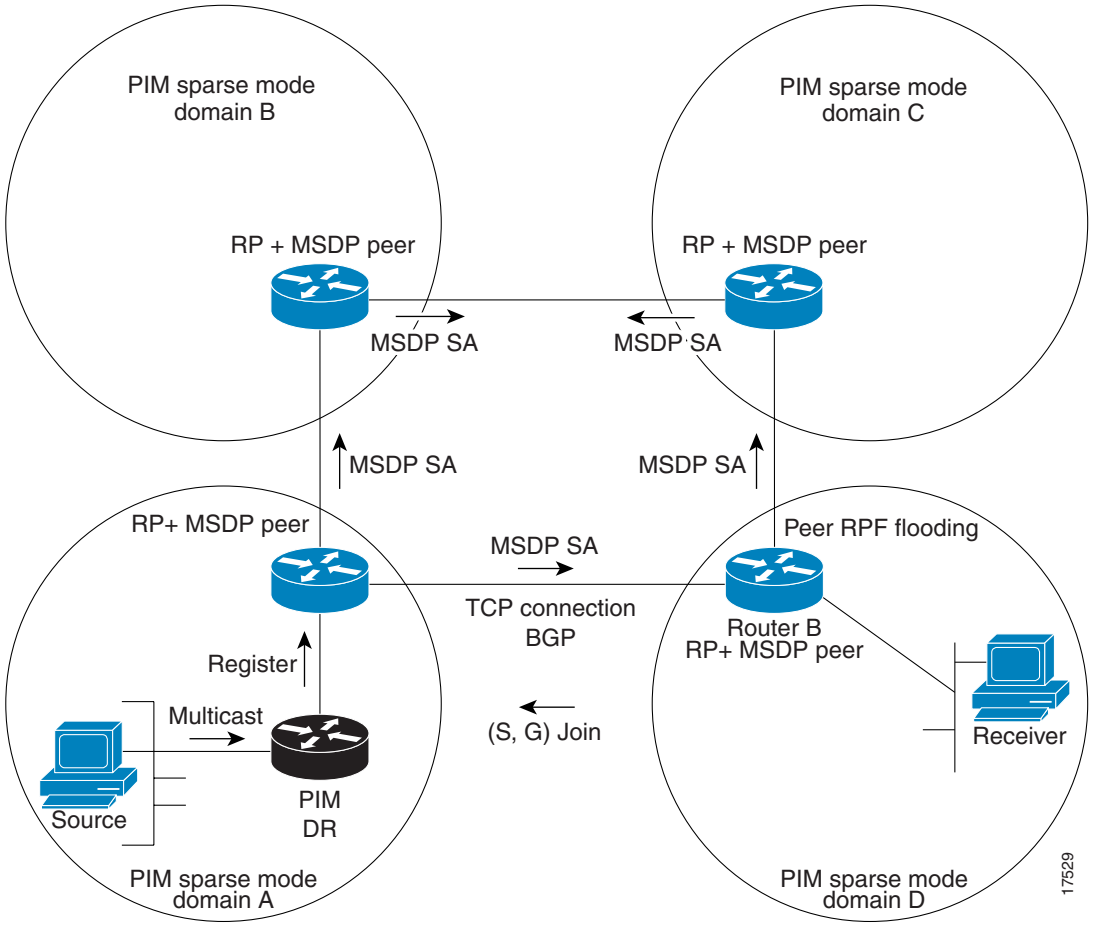
When MSDP is enabled, an RP in a PIM-SM domain maintains MSDP peering relationships with MSDP-enabled routers in other domains. This peering relationship occurs over a TCP connection, where primarily a list of sources sending to multicast groups is exchanged. MSDP uses TCP (port 639) for its peering connections. As with BGP, using point-to-point TCP peering means that each peer must be explicitly configured. The TCP connections between RPs, moreover, are achieved by the underlying routing system. The receiving RP uses the source lists to establish a source path. If the multicast sources

are of interest to a domain that has receivers, multicast data is delivered over the normal, source-tree building mechanism provided by PIM-SM. MSDP is also used to announce sources sending to a group. These announcements must originate at the RP of the domain.

Note MSDP depends on BGP or multiprotocol BGP (MBGP) for interdomain operation. We recommended that you run MSDP on RPs sending to global multicast groups.

Figure 1 illustrates MSDP operating between two MSDP peers. PIM uses MSDP as the standard mechanism to register a source with the RP of a domain.

Figure 1 MSDP Running Between RP Peers



When MSDP is implemented, the following sequence of events occurs:

1. When a PIM designated router (DR) registers a source with its RP as illustrated in Figure 1, the RP sends a Source-Active (SA) message to all of its MSDP peers.

Note The DR sends the encapsulated data to the RP only once per source (when the source goes active). If the source times out, this process happens again when it goes active again. This situation is different from the periodic SA message that contains all sources that are registered to the originating RP. Those SA messages are MSDP control packets, and, thus, do not contain encapsulated data from active sources.

2. The SA message identifies the source address, the group that the source is sending to, and the address or the originator ID of the RP, if configured.
3. Each MSDP peer that receives the SA message floods the SA message to all of its peers downstream from the originator. In some cases (such as the case with the RPs in PIM-SM domains B and C in Figure 1), an RP may receive a copy of an SA message from more than one MSDP peer. To prevent looping, the RP consults the BGP next-hop database to determine the next hop toward the originator of the SA message. If both MBGP and unicast BGP are configured, MBGP is checked first, and then unicast BGP. That next-hop neighbor is the RPF-peer for the originator. SA messages that are received from the originator on any interface other than the interface to the RPF peer are dropped. The SA message flooding process, therefore, is referred to as *peer-RPF flooding*. Because of the peer-RPF flooding mechanism, BGP or MBGP must be running in conjunction with MSDP.

Note (M)BGP is not required in MSDP mesh group scenarios. For more information about MSDP mesh groups, see the "Configuring an MSDP Mesh Group" section.

Note (M)BGP is not required in default MSDP peer scenarios or in scenarios where only one MSDP peer is configured. For more information, see the "Configuring a Default MSDP Peer" section.

4. When an RP receives an SA message, it checks to see whether there are any members of the advertised groups in its domain by checking to see whether there are interfaces on the group's (*, G) outgoing interface list. If there are no group members, the RP does nothing. If there are group members, the RP sends an (S, G) join toward the source. As a result, a branch of the interdomain source tree is constructed across autonomous system boundaries to the RP. As multicast packets arrive at the RP, they are then forwarded down its own shared tree to the group members in the RP's domain. The members' DRs then have the option of joining the rendezvous point tree (RPT) to the source using standard PIM-SM procedures.
5. The originating RP continues to send periodic SA messages for the (S, G) state every 60 seconds for as long as the source is sending packets to the group. When an RP receives an SA message, it caches the SA message. Suppose, for example, that an RP receives an SA message for (172.16.5.4, 228.1.2.3) from originating RP 10.5.4.3. The RP consults its mroute table and finds that there are no active members for group 228.1.2.3, so it passes the SA message to its peers downstream of 10.5.4.3. If a host in the domain then sends a join to the RP for group 228.1.2.3, the RP adds the interface toward the host to the outgoing interface list of its (*, 224.1.2.3) entry. Because the RP caches SA messages, the router will have an entry for (172.16.5.4, 228.1.2.3) and can join the source tree as soon as a host requests a join.

Note In all current and supported Cisco IOS software releases, caching of MSDP SA messages is mandatory and cannot be manually enabled or disabled. By default, when an MSDP peer is configured, the **ip multicast cache-sa-state** command will automatically be added to the running configuration. Prior to Cisco IOS Releases 12.1(7) and 12.0(14)S1, caching of SAs was disabled by default and could be enabled with the **ip msdp cache-sa-state** command.

MSDP Message Types

There are four basic MSDP message types, each encoded in their own Tag, Length, and Value (TLV) data format.

- SA messages
- SA request messages
- SA response messages
- Keepalive messages

SA Messages

SA messages are used to advertise active sources in a domain. In addition, these SA messages may contain the initial multicast data packet that was sent by the source.

SA messages contain the IP address of the originating RP as well as one or more (S, G) pairs being advertised. In addition, the SA message may contain an encapsulated data packet.

Note For more information about SA messages, see the "SA Message Origination, Receipt, and Processing" section.

SA Request Messages

SA request messages are used to request a list of active sources for a specific group. These messages are sent to an MSDP SA cache that is maintains a list of active (S, G) pairs in its SA cache. Join latency can be reduced by using SA request messages to request the list of active sources for a group instead of having to wait up to 60 seconds for all active sources in the group to be readvertised by originating RPs.

Note For more information about SA request messages, see the "Requesting Source Information from MSDP Peers" section.

SA Response Messages

SA response messages are sent by the MSDP peer in response to an SA request message. SA response messages contain the IP address of the originating RP as well as one or more (S, G) pairs of the active sources in the originating RP's domain that are stored in the cache.

Note For more information about SA response messages, see the "Controlling the Response to Outgoing SA Request Messages from MSDP Peers Using SA Request Filters" section.

Keepalive Messages

Keepalive messages are sent every 60 seconds in order to keep the MSDP session active. If no keepalive messages or SA messages are received for 75 seconds, the MSDP session is reset.

Note For more information about keepalive messages, see the "Adjusting the MSDP Keepalive and Hold-Time Intervals" section.

SA Message Origination, Receipt, and Processing

The section describes SA message origination, receipt, and processing in detail.

SA Message Origination

SA messages are triggered by an RP (assuming MSDP is configured) when any new source goes active within a local PIM-SM domain. A local source is a source that is directly connected to the RP or is the first-hop DR that has registered with it. An RP originates SA messages only for local sources in its PIM-SM domain; that is, for local sources that register with it.

Note A local source is denoted by the A flag being set in the (S, G) mroute entry on the RP (which can be viewed in the output of the **show ip mroute** command). This flag indicates that the source is a candidate for advertisement by the RP to other MSDP peers.

When a source is in the local PIM-SM domain, it causes the creation of (S, G) state in the RP. New sources are detected by the RP either by the receipt of a register message or the arrival of the first (S, G) packet from a directly connected source. The initial multicast packet sent by the source (either encapsulated in the register message or received from a directly connected source) is encapsulated in the initial SA message.

SA Message Receipt

SA messages are only accepted from the MSDP RPF peer that is in the best path back towards the originator. The same SA message arriving from other MSDP peers must be ignored or SA loops can occur. Deterministically selecting the MSDP RPF peer for an arriving SA message requires knowledge of the MSDP topology. However, MSDP does not distribute topology information in the form of routing updates. MSDP infers this information by using (M)BGP routing data as the best approximation of the MSDP topology for the SA RPF check mechanism. An MSDP topology, therefore, must follow the same general topology as the BGP peer topology. Besides a few exceptions (such as default MSDP peers and MSDP peers in MSDP mesh groups), MSDP peers, in general should also be (M)BGP peers.

How RPF Check Rules Are Applied to SA Messages

The rules that apply to RPF checks for SA messages are dependent on the BGP peerings between the MSDP peers:

- Rule 1: Applied when the sending MSDP peer is also an interior (M)BGP peer.
- Rule 2: Applied when the sending MSDP peer is also an exterior (M)BGP peer.
- Rule 3: Applied when the sending MSDP peer is not an (M)BGP peer.

RPF checks are not performed in the following cases:

- If the sending MSDP peer is the only MSDP peer, which would be the case if only a single MSDP peer or a default MSDP peer is configured.
- If the sending MSDP peer is a member of a mesh group.
- If the sending MSDP peer address is the RP address contained in the SA message.

How the Cisco IOS Software Determines the Rule to Apply to RPF Checks

The Cisco IOS software uses the following logic to determine which RPF rule to apply to RPF checks:

- Find the (M)BGP neighbor that has the same IP address as the sending MSDP peer.
  - If the matching (M)BGP neighbor is an internal BGP (iBGP) peer, apply Rule 1.
  - If the matching (M)BGP neighbor is an external BGP (eBGP) peer, apply Rule 2.
  - If no match is found, apply Rule 3.

Note The implication of the RPF check rule selection is as follows: The IP address used to configure an MSDP peer on a router must match the IP address used to configure the (M)BGP peer on the same router.

Rule 1 of RPF Checking of SA Messages in MSDP

Rule 1 of RPF checking in MSDP is applied when the sending MSDP peer is also an i(M)BGP peer. When Rule 1 is applied, the RPF check proceeds as follows:

1. The peer searches the BGP Multicast Routing Information Base (MRIB) for the best path to the RP that originated the SA message. If a path is not found in the MRIB, the peer then searches the Unicast Routing Information Base (URIB). If a path is still not found, the RPF check fails.
2. If the previous search succeeds (that is, the best path is found), the peer then determines the address of the BGP neighbor for this best path, which will be the address of the BGP neighbor that sent the peer the path in BGP update messages.

Note The BGP neighbor address is not the same as the next-hop address in the path. Because i(M)BGP peers do not update the next-hop attribute of a path, it is usually the case that the next-hop address is not the same as the address of the BGP peer that sent us the path.

Note The BGP neighbor address is also not necessarily the same as the BGP router ID of the peer that sent the peer the path.

3. If the IP address of the sending MSDP peer is the same as the BGP neighbor address (that is, the address of the BGP peer that sent the peer the path), then the RPF check succeeds; otherwise it fails.

Implications of Rule 1 of RPF Checking on MSDP

The MSDP topology must mirror the (M)BGP topology. In general, wherever there is an i(M)BGP peer connection between two routers, an MSDP peer connection should be configured. More specifically, the IP address of the far end MSDP peer connection must be the same as the far end i(M)BGP peer connection. The addresses must be the same because the BGP topology between i(M)BGP peers inside an autonomous system is not described by the AS path. If it were always the case that i(M)BGP peers

updated the next-hop address in the path when sending an update to another i(M)BGP peer, then the peer could rely on the next-hop address to describe the i(M)BGP topology (and hence the MSDP topology). However, because the default behavior for i(M)BGP peers is to not update the next-hop address, the peer cannot rely on the next-hop address to describe the (M)BGP topology (MSDP topology). Instead, the i(M)BGP peer uses the address of the i(M)BGP peer that sent the path to describe the i(M)BGP topology (MSDP topology) inside the autonomous system.

Tip Care should be taken when configuring the MSDP peer addresses to make sure that the same address is used for both i(M)BGP and MSDP peer addresses.

Rule 2 of RPF Checking of SA Messages in MSDP

Rule 2 of RPF checking in MSDP is applied when the sending MSDP peer is also an e(M)BGP peer. When Rule 2 is applied, the RPF check proceeds as follows:

1. The peer searches the BGP MRIB for the best path to the RP that originated the SA message. If a path is not found in the MRIB, the peer then searches the URIB. If a path is still not found, the RPF check fails.
2. If the previous search succeeds (that is, the best path is found), the peer then examines the path. If the first autonomous system in the best path to the RP is the same as the autonomous system of the e(M)BGP peer (which is also the sending MSDP peer), then the RPF check succeeds; otherwise it fails.

Implications of Rule 2 of RPF Checking on MSDP

The MSDP topology must mirror the (M)BGP topology. In general, wherever there is an e(M)BGP peer connection between two routers, an MSDP peer connection should be configured. As opposed to Rule 1, the IP address of the far end MSDP peer connection does *not* have to be the same as the far end e(M)BGP peer connection.The reason that the addresses do not have to be identical is that BGP topology between two e(M)BGP peers is not described by the AS path.

Rule 3 of RPF Checking of SA Messages in MSDP

Rule 3 of RPF checking is applied when the sending MSDP peer is not an (M)BGP peer at all. When Rule 3 is applied, the RPF check proceeds as follows:

1. The peer searches the BGP MRIB for the best path to the RP that originated the SA message. If a path is not found in the MRIB, the peer then searches the URIB. If a path is still not found, the RPF check fails.
2. If the previous search succeeds (that is, the best path to the RP that originated the SA message is found), the peer then searches the BGP MRIB for the best path to the MSDP peer that sent the SA message. If a path is not found in the MRIB, the peer then searches the URIB. If a path is still not found, the RPF check fails.

Note The autonomous system of the MSDP peer that sent the SA is the origin autonomous system, which is the last autonomous system in the AS path to the MSDP peer.

3. If the first autonomous system in the best path to the RP is the same as the autonomous system of the sending MSDP peer, then the RPF check succeeds; otherwise it fails.

SA Message Processing

The following steps are taken by an MSDP peer whenever it processes an SA message:

1. Using the group address G of the (S, G) pair in the SA message, the peer locates the associated (*, G) entry in the mroute table. If the (*, G) entry is found and its outgoing interface list is not null, then there are active receivers in the PIM-SM domain for the source advertised in the SA message.
2. The MSDP peer then creates an (S, G) entry for the advertised source.
3. If the (S, G) entry did not already exist, the MSDP peer immediately triggers an (S, G) join toward the source in order to join the source tree.
4. The peer then floods the SA message to all other MSDP peers with the exception of:
   - The MSDP peer from which the SA message was received.
   - Any MSDP peers that are in the same MSDP mesh group as this router (if the peer is a member of a mesh group).

Note SA messages are stored locally in the router's SA cache.

How to Use MSDP to Interconnect Multiple PIM-SM Domains

Perform the following tasks to use MSDP to interconnect multiple PIM-SM domains. The first task is required; all other tasks are optional.

- Configuring an MSDP Peer, page 10 (required)
- Shutting Down an MSDP Peer, page 11 (optional)
- Configuring MSDP MD5 Password Authentication Between MSDP Peers, page 12 (optional)
- Preventing DoS Attacks by Limiting the Number of SA Messages Allowed in the SA Cache from Specified MSDP Peers, page 15 (optional)
- Adjusting the MSDP Keepalive and Hold-Time Intervals, page 16 (optional)
- Adjusting the MSDP Connection-Retry Interval, page 18 (optional)
- Configuring MSDP Compliance with IETF RFC 3618, page 19 (optional)
- Configuring a Default MSDP Peer, page 21 (optional)
- Configuring an MSDP Mesh Group, page 23 (optional)
- Controlling SA Messages Originated by an RP for Local Sources, page 25 (optional)
- Controlling the Forwarding of SA Messages to MSDP Peers Using Outgoing Filter Lists, page 27 (optional)
- Controlling the Receipt of SA Messages from MSDP Peers Using Incoming Filter Lists, page 30 (optional)
- Using TTL Thresholds to Limit the Multicast Data Sent in SA Messages, page 32 (optional)
- Requesting Source Information from MSDP Peers, page 33 (optional)
- Controlling the Response to Outgoing SA Request Messages from MSDP Peers Using SA Request Filters, page 35 (optional)
- Including a Bordering PIM Dense Mode Region in MSDP, page 36 (optional)

- Configuring an Originating Address Other Than the RP Address, page 37 (optional)
- Monitoring MSDP, page 38 (optional)
- Clearing MSDP Connections, Statistics, and SA Cache Entries, page 41 (optional)
- Enabling SNMP Monitoring of MSDP, page 42 (optional)

Configuring an MSDP Peer

Perform this required task to configure an MSDP peer.

Note By enabling an MSDP peer, you implicitly enable MSDP.

MSDP Peers

Like BGP, MSDP establishes neighbor relationships with other MSDP peers. MSDP peers connect using TCP port 639. The lower IP address peer takes the active role of opening the TCP connection. The higher IP address peer waits in LISTEN state for the other to make the connection. MSDP peers send keepalive messages every 60 seconds. The arrival of data performs the same function as the keepalive message and keeps the session from timing out. If no keepalive messages or data is received for 75 seconds, the TCP connection is reset.

Prerequisites

- This task assumes that you have enabled IP multicast routing and have configured PIM-SM. For more information about PIM-SM, see the "IP Multicast Technology Overview" and the "Configuring Basic IP Multicast" module in the *Cisco IOS IP Multicast Configuration Guide*, Release 12.4.
- With the exception of a single MSDP peer, default MSDP peer, and MSDP mesh group scenarios, all MSDP peers must be configured to run BGP prior to being configured for MSDP. For more information about configuring BGP, see the "Configuring a Basic BGP Network" module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp peer** {*peer-name* | *peer-address*} [**connect-source** *type number*] [**remote-as** *as-number*]
4. **ip msdp description** {*peer-name* | *peer-address*} *text*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp peer** {*peer-name* | *peer-address*} [**connect-source** *type number*] [**remote-as** *as-number*] **Example:** `Router(config)# ip msdp peer 192.168.1.2 connect-source loopback0`	Enables MSDP and configures an MSDP peer as specified by the DNS name or IP address. **Note** The router that is selected to be configured as an MSDP peer is also usually a Border Gateway Protocol (BGP) neighbor. If it is not, see the "Configuring a Default MSDP Peer" section or the "Configuring an MSDP Mesh Group" section. • If you specify the **connect-source** keyword, the primary address of the specified local interface *type* and *number* values are used as the source IP address for the TCP connection. The **connect-source** keyword is recommended, especially for MSDP peers on a border that peer with a router inside of a remote domain.
Step 4	**ip msdp description** {*peer-name* | *peer-address*} *text* **Example:** `Router(config)# ip msdp description 192.168.1.2 router at customer a`	(Optional) Configures a description for a specified peer to make it easier to identify in a configuration or in **show** command output.
Step 5	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Shutting Down an MSDP Peer

Perform this optional task to shut down an MSDP peer.

If you are configuring several MSDP peers and you do not want any of the peers to go active until you have finished configuring all of them, you can shut down each peer, configure each peer, and later bring each peer up. You might also want to shut down an MSDP session without losing the configuration for that MSDP peer.

Note When an MSDP peer is shut down, the TCP connection is terminated and not restarted until the peer is brought back up using the **no** form of the **ip msdp shutdown** command (for the specified peer).

Prerequisites

This task assumes that you are running MSDP and have configured MSDP peers. For more information about configuring MSDP peers, see the "Configuring an MSDP Peer" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp shutdown** {*peer-name* | *peer address*}
4. Repeat Step 3 to shutdown additional MSDP peers.
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp shutdown** {*peer-name* | *peer-address*} **Example:** `Router(config)# ip msdp shutdown 192.168.1.3`	Administratively shuts down the specified MSDP peer.
Step 4	Repeat Step 3 to shutdown additional MSDP peers.	—
Step 5	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring MSDP MD5 Password Authentication Between MSDP Peers

Perform this optional task to configure MSDP MD5 password authentication between MSDP peers.

MSDP MD5 Password Authentication

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

How MSDP MD5 Password Authentication Works

Developed in accordance with RFC 2385, the MSDP MD5 password authentication feature is used to verify each segment sent on the TCP connection between MSDP peers. The **ip msdp password peer** 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. MD5 authentication must be configured with the same password on both MSDP peers; otherwise, the connection between them will not be made. Configuring MD5 authentication causes the Cisco IOS software to generate and verify the MD5 digest of every segment sent on the TCP connection.

Benefits of MSDP MD5 Password Authentication

- Protects MSDP against the threat of spoofed TCP segments being introduced into the TCP connection stream.
- Uses the industry-standard MD5 algorithm for improved reliability and security.

Prerequisites

This task assumes that you are running MSDP and have configured MSDP peers. For more information about configuring MSDP peers, see the "Configuring an MSDP Peer" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp** [**vrf** *name*] **password peer** {*peer-name* | *peer-address*} [*encryption-type*] *string*
4. **end**
5. **show ip msdp peer** [*peer-address* | *peer-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp** [**vrf** *name*] **password peer** {*peer-name* | *peer-address*} [*encryption-type*] *string* **Example:** `Router(config)# ip msdp password peer 10.32.43.144 0 test`	Enables MD5 password encryption for a TCP connection between two MSDP peers. **Note** MD5 authentication must be configured with the same password on both MSDP peers; otherwise, the connection between them will not be made. • If you configure or change the password or key used for MD5 authentication between two MSDP peers, the local router will not tear down the existing session after you configure the password. The local router will attempt to maintain the peering session using the new password until the keepalive period expires. If the password is not entered or changed on the remote router before the keepalive period expires, the session will time out and the MSDP session will reset.
Step 4	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**show ip msdp peer** [peer-address | peer-name] **Example:** `Router# show ip msdp peer`	(Optional) Displays detailed information about MSDP peers. **Note** Use this command to verify whether MD5 password authentication is enabled on an MSDP peer.

Troubleshooting Tips

If a router has a password configured for an MSDP peer, but the MSDP peer does not, a message such as the following will appear on the console while the routers attempt to establish an MSDP session between them:

```
%TCP-6-BADAUTH: No MD5 digest from [peer's IP address]:11003 to [local router's
IP address]:179
```

Similarly, if the two routers have different passwords configured, a message such as the following will appear on the console:

```
%TCP-6-BADAUTH: Invalid MD5 digest from [peer's IP address]:11004 to [local router's
IP address]:179
```

The **debug ip tcp transactions** command is used to display information on significant TCP transactions such as state changes, retransmissions, and duplicate packets. In the context of monitoring or troubleshooting MSDP MD5 password authentication, use the **debug ip tcp transactions** command to verify that the MD5 password is enabled and that the keepalive message is received by the MSDP peer.

Preventing DoS Attacks by Limiting the Number of SA Messages Allowed in the SA Cache from Specified MSDP Peers

Perform this optional (but highly recommended) task to limit the overall number of SA messages that the router can accept from specified MSDP peers. Performing this task protects an MSDP-enabled router from distributed denial-of-service (DoS) attacks.

Note We recommend that you perform this task for all MSDP peerings on the router.

SA Message Limits

The **ip msdp sa-limit** command is used to limit the overall number of SA messages that a router can accept from specified MSDP peers. When the **ip msdp sa-limit** command is configured, the router maintains a per-peer count of SA messages stored in the SA cache and will ignore new messages from a peer if the configured SA message limit for that peer has been reached.

The **ip msdp sa-limit** command was introduced as a means to protect an MSDP-enabled router from denial of service (DoS) attacks. We recommended that you configure SA message limits for all MSDP peerings on the router. An appropriately low SA limit should be configured on peerings with a stub MSDP region (for example, a peer that may have some further downstream peers but that will not act as a transit for SA messages across the rest of the Internet). A high SA limit should be configured for all MSDP peerings that act as transits for SA messages across the Internet.

Prerequisites

This task assumes that you are running MSDP and have configured MSDP peers. For more information about configuring MSDP peers, see the "Configuring an MSDP Peer" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp sa-limit** {*peer-address* | *peer-name*} *sa-limit*
4. Repeat Step 3 to configure SA limits for additional MSDP peers.
5. **end**
6. **show ip msdp count** [*as-number*]
7. **show ip msdp peer** [*peer-address* | *peer-name*]
8. **show ip msdp summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp sa-limit** {*peer-address* | *peer-name*} *sa-limit* **Example:** `Router(config)# ip msdp sa-limit 192.168.10.1 100`	Limits the number of SA messages allowed in the SA cache from the specified MSDP.
Step 4	Repeat Step 3 to configure SA limits for additional MSDP peers.	—
Step 5	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 6	**show ip msdp count** [*as-number*] **Example:** `Router# show ip msdp count`	(Optional) Displays the number of sources and groups originated in MSDP SA messages and the number of SA messages from an MSDP peer in the SA cache.
Step 7	**show ip msdp peer** [*peer-address* | *peer-name*] **Example:** `Router# show ip msdp peer`	(Optional) Displays detailed information about MSDP peers. **Note** The output of this command displays the number of SA messages received from MSDP peers that are stored in the cache.
Step 8	**show ip msdp summary** **Example:** `Router# show ip msdp summary`	(Optional) Displays MSDP peer status. **Note** The output of this command displays a per-peer "SA Count" field that displays the number of SAs stored in the cache.

Adjusting the MSDP Keepalive and Hold-Time Intervals

Perform this optional task to adjust the interval at which an MSDP peer will send keepalive messages and the interval at which the MSDP peer will wait for keepalive messages from other peers before declaring them down. By default, it may take as long as 75 seconds for an MSDP peer to detect that a peering session with another MSDP peer has gone down. In network environments with redundant MSDP peers, decreasing the hold-time interval (by lowering the value for *hold-time-interval* argument from the default of 75 seconds) can expedite the reconvergence time of MSDP peers in the event that an MSDP peer fails.

Note We recommend that you do not change the command defaults for the **ip msdp keepalive** command, as the command defaults are in accordance with RFC 3618, *Multicast Source Discovery Protocol*. If your network environment requires that you modify the defaults, you must configure the same time values for the *keepalive-interval* and *hold-time-interval* arguments on both ends of the MSDP peering session.

MSDP Keepalive and Hold-Time Intervals

The **ip msdp keepalive** command is used to adjust the interval at which an MSDP peer will send keepalive messages and the interval at which the MSDP peer will wait for keepalive messages from other peers before declaring them down.

Once an MSDP peering session is established, each side of the connection sends a keepalive message and sets a keepalive timer. If the keepalive timer expires, the local MSDP peer sends a keepalive message and restarts its keepalive timer; this interval is referred to as the keepalive interval. The *keepalive-interval* argument is used to adjust the interval for which keepalive messages will be sent. The keepalive timer is set to the value specified for the *keepalive-interval* argument when the peer comes up. The keepalive timer is reset to the value of the *keepalive-interval* argument whenever an MSDP keepalive message is sent to the peer and reset when the timer expires. The keepalive timer is deleted when an MSDP peering session is closed. By default, the keepalive timer is set to 60 seconds.

Note The value specified for the *keepalive-interval* argument must be less than the value specified for the *holdtime-interval* argument and must be at least one second.

The hold-time timer is initialized to the value of the *hold-time-interval* argument whenever an MSDP peering connection is established, and is reset to the value of the *hold-time-interval* argument whenever an MSDP keepalive message is received. The hold-time timer is deleted whenever an MSDP peering connection is closed. By default, the hold-time interval is set to 75 seconds.

Use the *hold-time-interval* argument to adjust the interval at which the MSDP peer will wait for keepalive messages from other peers before declaring them down.

Prerequisites

This task assumes that you are running MSDP and have configured MSDP peers. For more information about configuring MSDP peers, see the "Configuring an MSDP Peer" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp** [**vrf** *vrf-name*] **keepalive** {*peer-address* | *peer-name*} *keepalive-interval hold-time-interval*
4. Repeat Step 3 to adjust the keepalive message interval for additional MSDP peers.
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp** [**vrf** *vrf-name*] **keepalive** {*peer-address* | *peer-name*} *keepalive-interval hold-time-interval* **Example:** `Router(config)# ip msdp keepalive 10.1.1.3 40 55`	Configures the interval at which an MSDP peer will send keepalive messages and the interval at which the MSDP peer will wait for keepalive messages from other peers before declaring them down. • By default, an MSDP peer sends keepalive messages at an interval of once every 60 seconds, and the hold-time interval for an MSDP peer is set to 75 seconds. • Use the *keepalive-interval* argument to specify the interval, in seconds, at which the MSDP peer will send keepalive messages. The range is from 1 to 60. • Use the *hold-time-interval* argument to specify the interval, in seconds, at which the MSDP peer will wait for keepalive messages from other peers before declaring them down. The range is from 1 to 75.
Step 4	Repeat Step 3 to adjust the keepalive message interval for additional MSDP peers.	—
Step 5	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Adjusting the MSDP Connection-Retry Interval

Perform this optional task to adjust the interval at which MSDP peers will wait after peering sessions are reset before attempting to reestablish the peering sessions. In network environments where fast recovery of SA messages is required (such as in trading floor network environments), you may want to decrease the connection-retry interval to a time value less than the default value of 30 seconds.

MSDP Connection-Retry Interval

The **ip msdp timer** command is used to adjust the interval at which all MSDP peers will wait after peering sessions are reset before attempting to reestablish the peering sessions. This interval is referred to as the connection-retry interval. By default, MSDP peers will wait 30 seconds after the session is reset before attempting to reestablish sessions with other peers. When the **ip msdp timer** command is configured, the configured connection-retry interval applies to all MSDP peering sessions on the router.

Prerequisites

This task assumes that you are running MSDP and have configured MSDP peers. For more information about configuring MSDP peers, see the "Configuring an MSDP Peer" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp** [**vrf** *vrf-name*] **timer** *connection-retry-interval*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp** [**vrf** *vrf-name*] **timer** *connection-retry-interval* **Example:** `Router# ip msdp timer 45`	Configures the interval at which MSDP peers will wait after peering sessions are reset before attempting to reestablish the peering sessions. • By default, an MSDP peer will wait 30 seconds after a peering session is reset before attempting to reestablish the peering session with any peer. • Use the *connection-retry-interval* argument to specify the interval, in seconds, at which MSDP peers will wait after peering sessions are reset before attempting to reestablish the peering sessions. The range is from 1 to 60.
Step 4	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring MSDP Compliance with IETF RFC 3618

Perform this optional task to configure MSDP peers to be compliant with Internet Engineering Task Force (IETF) RFC 3618 specifications for MSDP.

MSDP Compliance with IETF RFC 3618

When the MSDP Compliance with IETF RFC 3618 feature is configured, the peer-RPF forwarding rules defined in IETF RFC 3618 are applied to MSDP peers. IETF RFC 3618 provides peer-RPF forwarding rules that are used for forwarding SA messages throughout an MSDP-enabled internet. Unlike the RPF check used when forwarding data packets, which compares a packet's source address against the interface upon which the packet was received, the peer-RPF check compares the RP address carried in the SA message against the MSDP peer from which the message was received. Except when MSDP mesh groups are being used, SA messages from an RP address are accepted from only one MSDP peer to avoid looping SA messages.

Note For more information about the MSDP peer-forwarding rules defined in RFC 3618, see RFC 3618, *Multicast Source Discovery Protocol (MSDP)*.

Benefits of MSDP Compliance with RFC 3618

- You can use BGP route reflectors (RRs) without running MSDP on them. This capability is useful to service providers that need to reduce the load on RRs.
- You can use an Interior Gateway Protocol (IGP) for the Reverse Path Forwarding (RPF) checks and thereby run peerings without (M)BGP. This capability is useful to enterprise customers that do not run (M)BGP and require larger topologies than mesh groups can provide.

Note IGP peerings must always be between directly connected MSDP peers or else the RPF checks will fail.

- You can have peerings between routers in nondirectly connected autonomous systems (that is, with one or more autonomous systems between them). This capability helps in confederation configurations and for redundancy.

Prerequisites

This task assumes that you are running MSDP and have configured MSDP peers. For more information about configuring MSDP peers, see the "Configuring an MSDP Peer" task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp** [**vrf** *vrf-name*] **rpf rfc3618**
4. **end**
5. **show ip msdp** [**vrf** *vrf-name*] **rpf-peer** *rp-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp** [**vrf** *vrf-name*] **rpf rfc3618** **Example:** `Router(config)# ip msdp vrf vrf1 rpf rfc3618`	Enables compliance with the peer-RPF forwarding rules specified in IETF RFC 3618.
Step 4	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**show ip msdp** [**vrf** *vrf-name*] **rpf-peer** *rp-address* **Example:** `Router# show ip msdp rpf-peer 192.168.1.5`	(Optional) Displays the unique MSDP peer information from which a router will accept SA messages originating from the specified RP.

Configuring a Default MSDP Peer

Perform this optional task to specify a default MSDP peer.

Default MSDP Peers

In most scenarios, an MSDP peer is also a BGP peer. If an autonomous system is a stub or nontransit autonomous system, and particularly if the autonomous system is not multihomed, there is little or no reason to run BGP to its transit autonomous system. A static default route at the stub autonomous system, and a static route pointing to the stub prefixes at the transit autonomous system, is generally sufficient. But if the stub autonomous system is also a multicast domain and its RP must peer with an RP in the neighboring domain, MSDP depends on the BGP next-hop database for its peer-RPF checks. You can disable this dependency on BGP by defining a default peer from which to accept all SA messages without performing the peer-RPF check, using the **ip msdp default-peer** command. A default MSDP peer must be a previously configured MSDP peer.

A stub autonomous system also might want to have MSDP peerings with more than one RP for the sake of redundancy. For example, SA messages cannot just be accepted from multiple default peers, because there is no RPF check mechanism. Instead, SA messages are accepted from only one peer. If that peer fails, SA messages are then accepted from the other peer. The underlying assumption here, of course, is that both default peers are sending the same SA messages.

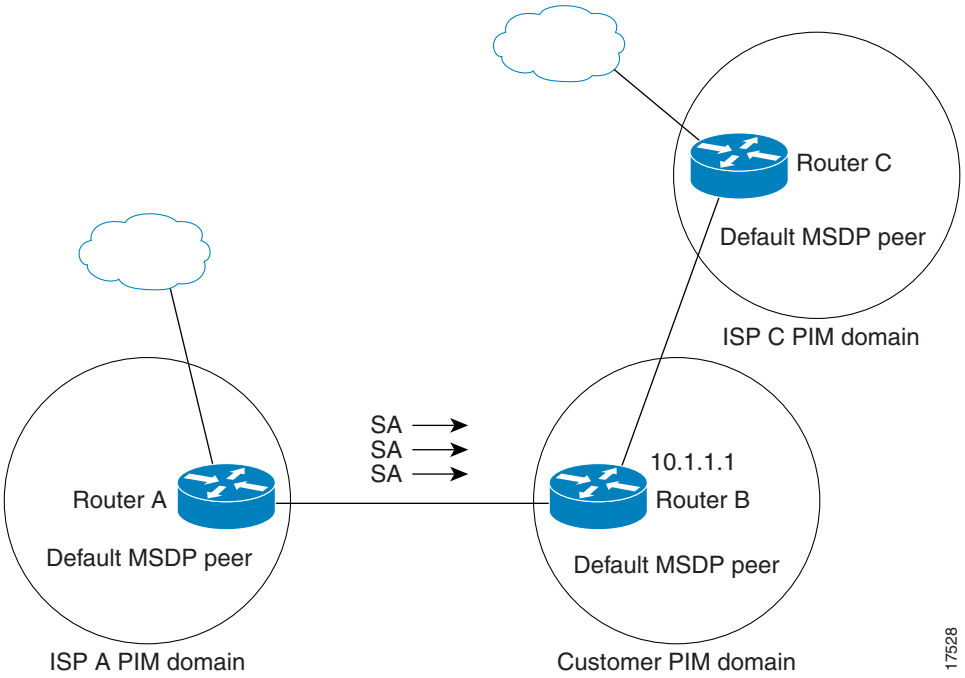
Figure 2 illustrates a scenario where default MSDP peers might be used. In the figure, a customer that owns Router B is connected to the Internet through two Internet service providers (ISPs), one that owns Router A and the other that owns Router C. They are not running BGP or MBGP between them. In order

for the customer to learn about sources in the ISP domain or in other domains, Router B identifies Router A as its default MSDP peer. Router B advertises SA messages to both Router A and Router C, but accepts SA messages either from Router A only or Router C only. If Router A is the first default peer in the configuration, it will be used if it is up and running. Only if Router A is not running will Router B accept SA messages from Router C.

The ISP will also likely use a prefix list to define which prefixes it will accept from the customer router. The customer will define multiple default peers, each having one or more prefixes associated with it.

The customer has two ISPs to use. The customer defines both ISPs as default peers. As long as the first default peer identified in the configuration is up and running, it will be the default peer and the customer will accept all SA messages it receives from that peer.

Figure 2 Default MSDP Peer Scenario



Router B advertises SAs to Router A and Router C, but uses only Router A or Router C to accept SA messages. If Router A is first in the configuration, it will be used if it is up and running. Only when Router A is not running will Router B accept SAs from Router C. This is the behavior without a prefix list.

If you specify a prefix list, the peer will be a default peer only for the prefixes in the list. You can have multiple active default peers when you have a prefix list associated with each. When you do not have any prefix lists, you can configure multiple default peers, but only the first one is the active default peer as long as the router has connectivity to this peer and the peer is alive. If the first configured peer goes down or the connectivity to this peer goes down, the second configured peer becomes the active default, and so on.

Prerequisites

An MSDP default peer must be a previously configured MSDP peer. Before configuring a default MSDP peer, you must first configure an MSDP peer. For more information about configuring MSDP peers, see the "Configuring an MSDP Peer" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp default-peer** {*peer-address* | *peer-name*} [**prefix-list** *list*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp default-peer** {*peer-address* | *peer-name*} [**prefix-list** *list*] **Example:** `Router(config)# ip msdp default-peer 192.168.1.3`	Defines a default peer from which to accept all MSDP SA messages
Step 4	**end** **Example:** `Router (config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring an MSDP Mesh Group

Perform this optional task to configure an MSDP mesh group.

MSDP Mesh Groups

An MSDP mesh group is a group of MSDP speakers that have fully meshed MSDP connectivity between one another. In other words, each of the MSDP peers in the group must have an MSDP peering relationship (MSDP connection) to every other MSDP peer in the group. When an MSDP mesh group is configured between a group of MSDP peers, SA message flooding is reduced. Because when an MSDP peer in the group receives an SA message from another MSDP peer in the group, it assumes that this SA message was sent to all the other MSDP peers in the group. As a result, it is not necessary for the receiving MSDP peer to flood the SA message to the other MSDP peers in the group.

Benefits of MSDP Mesh Groups

- Optimizes SA flooding—MSDP mesh groups are particularly useful for optimizing SA flooding when two or more peers are in a group.

- Reduces the amount of SA traffic across the Internet—When MSDP mesh groups are used, SA messages are not flooded to other mesh group peers.
- Eliminates RPF checks on arriving SA messages—When an MSDP mesh group is configured, SA messages are always accepted from mesh group peers.

Note You can configure multiple mesh groups per router.

Prerequisites

This task assumes that you are running MSDP and have configured MSDP peers. For more information about configuring MSDP peers, see the "Configuring an MSDP Peer" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp mesh-group** *mesh-name* {*peer-address* | *peer-name*}
4. Repeat Step 3 to add additional MSDP peers as members of the mesh group.
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp mesh-group** *mesh-name* {*peer-address* | *peer-name*} **Example:** `Router(config)# ip msdp mesh-group peermesh`	Configures an MSDP mesh group and indicates that an MSDP peer belongs to that mesh group. **Note** All MSDP peers on a router that participate in a mesh group must be fully meshed with all other MSDP peers in the group. Each MSDP peer on each router, therefore, must be configured as a peer with **ip msdp peer** and as a member of the mesh group using the **ip msdp mesh-group** command.
Step 4	Repeat Step 3 to add additional MSDP peers as members of the mesh group.	—
Step 5	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Controlling SA Messages Originated by an RP for Local Sources

Perform this task to control SA messages originated by an RP by enabling a filter to restrict which registered sources are advertised in SA messages.

Note For best practice information related to configuring MSDP SA message filters, see the *Multicast Source Discovery Protocol SA Filter Recommendations* tech note.

SA Origination Filters

By default, an RP that is configured to run MSDP will originate SA messages for all local sources for which it is the RP. Local sources that register with an RP, therefore, will be advertised in SA messages, which in some cases is not desirable. For example, if sources inside a PIM-SM domain are using private addresses (for example, network 10.0.0.0/8), you should configure an SA origination filter to restrict those addresses from being advertised to other MSDP peers across the Internet.

To control what sources are advertised in SA messages, you can configure SA origination filters on an RP using the **ip msdp redistribute** command. By creating SA origination filters, you can control the sources advertised in SA messages as follows:

- You can prevent an RP from originating SA messages for local sources by configuring the **ip msdp redistribute** command without any keywords or arguments. Issuing this form of the command effectively prevents the router from advertising local sources in SA messages.

Note When the **ip msdp redistribute** command is entered without any keywords or arguments, the router will still forward SA messages from other MSDP peers in the normal fashion; it will just not originate any SA messages for local sources.

- You can configure the router to originate SA messages only for (S, G) pairs defined in an extended access list by configuring the **ip msdp redistribute** command with the optional **list** keyword and *access-list* argument. Issuing the form of the command effectively configures the router to only originate SA messages for local sources sending to specific groups that match (S, G) pairs defined in the extended access list. All other local sources will not be advertised in SA messages.
- You can configure the router to originate SA messages only for AS paths defined in an AS-path access list by configuring the **ip msdp redistribute** command with the optional **asn** keyword and *as-access-list* argument. Issuing this form of the command effectively configures the router to only originate SA messages for local sources sending to specific groups that the match AS paths defined in an AS-path access list. All other local sources will not be advertised in SA messages.

Note AS-path access lists are configured using the **ip as-path access-list** command. For more information about the **ip as-path access-list** command, see the *Cisco IOS Routing Protocols Command Reference*.

- You can configure the router to originate SA messages only for local sources that match the criteria defined in a route map by configuring the **ip msdp redistribute** command with the optional **route-map** keyword and *map-name* argument. Issuing this form of the command effectively configures the router to only originate SA messages for local sources that match the criteria defined in the route map. All other local sources will not be advertised in SA messages.

Note You can configure an SA origination filter that includes an extended access list, an AS-path access list, and route map (or a combination thereof). In that case, all conditions must be true before any local sources are advertised in SA messages.

Prerequisites

- This task assumes that you are running MSDP and have configured MSDP peers. For more information about configuring MSDP peers, see the "Configuring an MSDP Peer" section.
- If you plan on configuring an SA origination filter that references an extended access list, AS-path access list, or route map, you must configure the extended access list, AS-path access list, or route map prior to performing this task.

  For more information about configuring extended access lists, see the "Creating an IP access list and Applying It to an Interface" module.

  For more information about configuring AS-path access lists, see the "Connecting to a Service Provider Using External BGP" module.

  For more information about configuring route maps, see the "Configuring IP Routing Protocol-Independent Features" module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp redistribute** [**list** *access-list*] [**asn** *as-access-list*] [**route-map** *map-name*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip msdp redistribute** [**list** *access-list*] [**asn** *as-access-list*] [**route-map** *map-name*] **Example:** `Router(config)# ip msdp redistribute route-map customer-sources`	Enables a filter for MSDP SA messages originated by the local router. **Note** The **ip msdp redistribute** command could also be used to advertise sources that are known to the RP but not registered. However, it is strongly recommended that you *not* originate advertisements for sources that have not registered with the RP.
Step 4	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Controlling the Forwarding of SA Messages to MSDP Peers Using Outgoing Filter Lists

Perform this optional task to control the forwarding of SA messages to MSDP peers by configuring outgoing filter lists.

Note For best practice information related to configuring MSDP SA message filters, see the *Multicast Source Discovery Protocol SA Filter Recommendations* tech note.

Use of Outgoing Filter Lists in MSDP

By default, an MSDP-enabled router forwards all SA messages it receives to all of its MSDP peers. However, you can prevent SA messages from being forwarded to MSDP peers by creating outgoing filter lists using the **ip msdp sa-filter out** command.

Note Outgoing filter lists (configured using the **ip msdp sa-filter out** command) apply to all SA messages, whether locally originated or received from another MSDP peer, whereas SA origination filters (configured using the **ip msdp redistribute** command) apply only to locally originated SA messages. For more information about using the **ip msdp redistribute** command to enable a filter for MSDP SA messages originated by the local router, see the "Controlling SA Messages Originated by an RP for Local Sources" section.

By creating an outgoing filter list, you can control the SA messages that a router forwards to a peer as follows:

- You can filter all outgoing SA messages forwarded to a specified MSDP peer by configuring the **ip msdp filter-sa-request out** command without any keywords or arguments. Issuing this form of the command effectively configures the router to stop forwarding its SA messages to the MSDP peer.
- You can filter a subset of outgoing SA messages forwarded to a specified MSDP peer based on (S, G) pairs defined in an extended access list by configuring the **ip msdp sa-filter out** command with the optional **list** keyword and *access-list* argument. Issuing the form of the command effectively configures the router to only forward SA messages to the MSDP peer that match the (S, G) pairs permitted in an extended access list. The forwarding of all other SA messages to the MSDP peer will be stopped.
- You can filter a subset of outgoing SA messages forwarded to a specified MSDP peer based on match criteria defined in a route map by configuring the **ip msdp sa-filter out** command with the optional **route-map** keyword and *map-name* argument. Issuing this form of the command effectively configures the router to only forward SA messages that match the criteria defined in the route map. The forwarding of all other SA messages to the MSDP peer will be stopped.
- You can filter a subset of outgoing SA messages from a specified peer based on the announcing RP address contained in the SA message by configuring the **ip msdp sa-filter out** command with the optional **rp-list** keyword and *list* argument or with the **rp-route-map** keyword *map-name* argument. This type of outgoing filter list enables the router to filter outgoing SA messages based on their origin, even after an SA message has been transmitted across one or more MSDP peers. The forwarding of all other SA messages to the MSDP peer will be stopped.

Note You can configure an outgoing filter list that includes an extended access list, a route map, and either an RP access list or an RP route map. In that case, all conditions must be true for the MSDP peer to forward the outgoing SA message.

Caution Arbitrary filtering of SA messages can result in downstream MSDP peers being starved of SA messages for legitimate active sources. Care, therefore, should be taken when using these sorts of filters. Normally, outgoing filter lists are used only to reject undesirable sources, such as sources using private addresses.

Prerequisites

- This task assumes that you are running MSDP and have configured MSDP peers. For more information about configuring MSDP peers, see the "Configuring an MSDP Peer" section.
- If you plan on configuring an outgoing filter list that references an extended access list or route map, you must configure the extended access list or route map prior to performing this task.

For more information about configuring extended access lists, see the "Creating an IP Access List and Applying It to an Interface" module.

For more information about configuring route maps, see the "Configuring IP Routing Protocol-Independent Features" module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp sa-filter out** {*peer-address* | *peer-name*} [**list** *access-list*] [**route-map** *map-name*] [**rp-list** *access-list* | **rp-route-map** *map-name*]
4. Repeat Step 3 to configure outgoing filter lists for additional MSDP peers.
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp sa-filter out** {*peer-address* | *peer-name*} [**list** *access-list*] [**route-map** *map-name*] [**rp-list** *access-list* | **rp-route-map** *map-name*] **Example:** `Router(config)# ip msdp sa-filter out 192.168.1.5 peerone`	Enables a filter for outgoing MSDP messages.
Step 4	Repeat Step 3 to configure outgoing filter lists for additional MSDP peers.	—
Step 5	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Controlling the Receipt of SA Messages from MSDP Peers Using Incoming Filter Lists

Perform this optional task to control the receipt of incoming SA messages from MSDP peers.

Note For best practice information related to configuring MSDP SA message filters, see the *Multicast Source Discovery Protocol SA Filter Recommendations* tech note.

Use of Incoming Filter Lists in MSDP

By default, an MSDP-enabled router receives all SA messages sent to it from its MSDP peers. However, you can control the source information that a router receives from its MSDP peers by creating incoming filter lists using the **ip msdp sa-filter in** command.

By creating incoming filter lists, you can control the incoming SA messages that a router receives from its peers as follows:

- You can filter all incoming SA messages from a specified MSDP peer by configuring the **ip msdp filter-sa-request in** command without any keywords or arguments. Issuing this form of the command effectively configures the router to ignore all SA messages sent to it from the specified MSDP peer.
- You can filter a subset of incoming SA messages from a specified peer based on (S, G) pairs defined in an extended access list by configuring the **ip msdp sa-filter in** command with the optional **list** keyword and *access-list* argument. Issuing the form of the command effectively configures the router to only receive SA messages from the MSDP peer that match the (S, G) pairs defined in the extended access list. All other incoming SA messages from the MSDP peer will be ignored.
- You can filter a subset of incoming SA request messages from a specified peer based on match criteria defined in a route map by configuring the **ip msdp sa-filter in** command with the optional **route-map** keyword and *map* argument. Issuing this form of the command effectively configures the router to only receive SA messages that match the criteria defined in the route map. All other incoming SA messages from the MSDP peer will be ignored.
- You can filter a subset of incoming SA messages from a specified peer based on both (S, G) pairs defined in an extended access list and on match criteria defined in a route map by configuring the **ip msdp sa-filter in** command with the optional **list** keyword and *access-list* argument and with the optional **route-map** keyword and *map-name* argument. Issuing this form of the command configures the router to only receive incoming SA messages that both match the (S, G) pairs defined in the extended access list and match the criteria defined in the route map. All other incoming SA messages from the MSDP peer will be ignored.
- You can filter a subset of incoming SA messages from a specified peer based on the announcing RP address contained in the SA message by configuring the **ip msdp sa-filter in** command with the optional **rp-list** keyword and *list* argument or with the **rp-route-map** *map-name*. This type of incoming filter list enables the router to filter incoming SA messages based on their origin, even after the SA message may have already been transmitted across one or more MSDP peers.

Note You can configure an incoming filter list that includes an extended access list, a route map, and either an RP access list or an RP route map. In that case, all conditions must be true for the MSDP peer to receive the incoming SA message.

Caution Arbitrary filtering of SA messages can result in downstream MSDP peers being starved of SA messages for legitimate active sources. Care, therefore, should be taken when using these sorts of filters. Normally, incoming filter lists are used only to reject undesirable sources, such as sources using private addresses.

Prerequisites

- This task assumes that you are running MSDP and have configured MSDP peers. For more information about configuring MSDP peers, see the "Configuring an MSDP Peer" section.
- If you plan on configuring an incoming filter list that references an extended access list or route map, you must configure the extended access list or route map prior to performing this task.

  For more information about configuring extended access lists, see the "Creating an IP Access List and Applying It to an Interface" module.

  For more information about configuring route maps, see the "Configuring IP Routing Protocol-Independent Features" module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp sa-filter in** {*peer-address* | *peer-name*} [**list** *access-list*] [**route-map** *map-name*] [**rp-list** *access-list* | **rp-route-map** *map-name*]
4. Repeat Step 3 to configure incoming filter lists for additional MSDP peers.
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp sa-filter in** {*peer-address* | *peer-name*} [**list** *access-list*] [**route-map** *map-name*] [**rp-list** *access-list* | **rp-route-map** *map-name*] **Example:** `Router(config)# ip msdp sa-filter in 192.168.1.3`	Enables a filter for incoming MSDP SA messages.

	Command or Action	Purpose
Step 4	Repeat Step 3 to configure incoming filter lists for additional MSDP peers.	—
Step 5	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Using TTL Thresholds to Limit the Multicast Data Sent in SA Messages

Perform this optional task to establish a TTL threshold.

TTL Thresholds in MSDP

The time-to-live (TTL) value provides a means to limit the number of hops a packet can take before being dropped. The **ip multicast ttl-threshold** command is used to specify a TTL for data-encapsulated SA messages sent to specified MSDP peers. By default, multicast data packets in SA messages are sent to an MSDP peer, provided the TTL value of the packet is greater than 0, which is standard TTL behavior.

In general, a TTL-threshold problem can be introduced by the encapsulation of a source's initial multicast packet in an SA message. Because the multicast packet is encapsulated inside of the unicast SA message (whose TTL is 255), its TTL is not decremented as the SA message travels to the MSDP peer. Furthermore, the total number of hops that the SA message traverses can be drastically different than a normal multicast packet because multicast and unicast traffic may follow completely different paths to the MSDP peer and hence the remote PIM-SM domain. As a result, encapsulated packets can end up violating TTL thresholds. The solution to this problem is to configure a TTL threshold that is associated with any multicast packet that is encapsulated in an SA message sent to a particular MSDP peer using the **ip multicast ttl-threshold** command. The **ip msdp ttl-threshold** command prevents any multicast packet whose TTL in the IP header is less than the TTL value specified for the *ttl-value* argument from being encapsulated in SA messages sent to that peer.

Prerequisites

This task assumes that you are running MSDP and have configured MSDP peers. For more information about configuring MSDP peers, see the "Configuring an MSDP Peer" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp ttl-threshold** {*peer-address* | *peer-name*} *ttl-value*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp ttl-threshold** {*peer-address* | *peer-name*} *ttl-value* `Router(config)# ip msdp ttl-threshold 192.168.1.5 8`	Sets a TTL value for MSDP messages originated by the local router. • By default, multicast data packets in SA messages are sent to an MSDP peer, provided the TTL value of the packet is greater than 0, which is standard TTL behavior.
Step 4	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Requesting Source Information from MSDP Peers

Perform this optional task to enable a router to request source information from MSDP peers.

Note Because SA caching is enabled by default and cannot be explicitly enabled or disabled in Cisco IOS Release 12.1(7) and 12.0(14)S1 and later Cisco IOS software releases, performing this task is seldom needed.

SA Request Messages

The **ip msdp sa-request** command is used to enable a noncaching router to send SA request messages to a specified MSDP peer. You can enter this command multiple times to specify that the router send SA request messages to additional MSDP peers.

If an noncaching RP has an MSDP peer that is caching SAs, you can reduce the join latency for a noncaching peer by enabling the noncaching peer to send SA request messages. When a host requests a join to a particular group, the noncaching RP sends an SA request message to its caching peers. If a peer has cached source information for the group in question, it sends the information to the requesting RP with an SA response message. The requesting RP uses the information in the SA response but does not forward the message to any other peers. If a noncaching RP receives an SA request, it sends an error message back to the requestor.

Note In all current and supported Cisco IOS software releases, caching of MSDP SA messages is mandatory and cannot be manually enabled or disabled. By default, when an MSDP peer is configured, the **ip multicast cache-sa-state** command will automatically be added to the running configuration. Prior to Cisco IOS Releases 12.1(7) and 12.0(14)S1, caching of SAs was disabled by default and could be enabled with the **ip msdp cache-sa-state** command.

Prerequisites

This task assumes that you are running MSDP and have configured MSDP peers. For more information about configuring MSDP peers, see the "Configuring an MSDP Peer" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp sa-request** {*peer-address* | *peer-name*}
4. Repeat Step 3 to specify that the router send SA request messages to additional MSDP caching peers.
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp sa-request** {*peer-address* | *peer-name*} **Example:** `Router(config)# ip msdp sa-request 192.168.10.1`	Specifies that the router send SA request messages to the specified MSDP peer.
Step 4	Repeat Step 3 to specify that the router send SA request messages to additional MSDP caching peers.	—
Step 5	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Controlling the Response to Outgoing SA Request Messages from MSDP Peers Using SA Request Filters

Perform this optional task to control the outgoing SA request messages that the router will honor from MSDP peers.

SA Request Filters

By default, a router honors all outgoing SA request messages from its MSDP peers; that is, it sends cached source information to requesting MSDP peers in SA response messages. You can control the outgoing SA request messages that a router will honor from specified peers by enabling an SA request filter using the **ip msdp filter-sa-request** command. By creating an SA request filter, you can control the outgoing SA requests that the router will honor from MSDP peers as follows:

- You can filter all SA request messages from a specified peer by configuring the **ip msdp filter-sa-request** command without the optional **list** keyword and *access-list* argument. Issuing this form of the **ip msdp filter-sa request** command effectively configures the router to ignore all SA requests from the specified MSDP peer.
- You can filter a subset of SA request messages from a specified peer based on groups defined in a standard access list by configuring the **ip msdp filter-sa-request** command with the optional **list** keyword and *access-list* argument. Issuing the form of the command effectively configures the router to honor only SA request messages from the MSDP peer that match the groups defined in a standard access list. SA request messages from the specified peer for other groups will be ignored.

Prerequisites

- This task assumes that you are running MSDP and have configured MSDP peers. For more information about configuring MSDP peers, see the "Configuring an MSDP Peer" section.
- If you plan on configuring an SA request filter that references a standard access list, configure the standard access list prior to performing this task. For more information about configuring standard access lists, see the "Creating an IP Access List and Applying It to an Interface" module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp filter-sa-request** {*peer-address* | *peer-name*} [**list** *access-list*]
4. Repeat Step 3 to configure SA request filters for additional MSDP peers.
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp filter-sa-request** {*peer-address* | *peer-name*} [**list** *access-list*] **Example:** `Router(config)# ip msdp filter sa-request 172.31.2.2 list 1`	Enables a filter for outgoing SA request messages. **Note** Only one SA request filter can be configured per MSDP peer.
Step 4	Repeat Step 3 to configure SA request filters for additional MSDP peers.	—
Step 5	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Including a Bordering PIM Dense Mode Region in MSDP

Perform this optional task to configure a border router to send SA messages for sources active in the dense mode region.

You might have a router that borders a PIM-SM region and a PIM-DM region. By default, sources in the PIM-DM domain are not included in MSDP. You could configure this border router to send SA messages for sources active in the PIM-DM domain. If you do so, it is very important to also configure the **ip msdp redistribute** command to control what local sources from the PIM-DM domain are advertised. Not configuring this command can result in the (S, G) state remaining long after a source in the PIM-DM domain has stopped sending.

Note For more information about using the **ip msdp redistribute** command to control the sources advertised in SA messages, see the "Controlling SA Messages Originated by an RP for Local Sources" section.

Prerequisites

This task assumes that you are running MSDP and have configured MSDP peers. For more information about configuring MSDP peers, see the "Configuring an MSDP Peer" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp border sa-address** *type number*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp border sa-address** *type number* **Example:** `Router(config)# ip msdp border sa-address ethernet0`	Configures the router on the border between a PIM-SM and PIM-DM domain to originate SA messages for active sources in the PIM-DM domain. • The IP address of the interface is used as the originator ID, which is the RP field in the SA message.
Step 4	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring an Originating Address Other Than the RP Address

Perform this optional task to allow an MSDP speaker that originates an SA message to use the IP address of its interface as the RP address in the SA message.

The **ip msdp originator-id** command is used to change the default RP address used in MSDP messages. If you need to change the originator ID for any reason, use the **ip msdp originator-id** command. For example, you might change the originator ID in one of these cases:

- If you configure multiple routers in an MSDP mesh group for Anycast RP.
- If you have a router that borders a PIM-SM domain and a PIM-DM domain. If a router borders a PIM-SM domain and a PIM-DM domain and you want to advertise active sources within the PIM-DM domain, use the **ip msdp originator-id** command to configure the RP address in SA messages to be the address of the originating router's interface.

Prerequisites

This task assumes that you are running MSDP and have configured MSDP peers. For more information about configuring MSDP peers, see the "Configuring an MSDP Peer" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp originator-id** *type number*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp originator-id** *type number* **Example:** `Router(config)# ip msdp originator-id ethernet 1`	Configures the RP address in SA messages to be the address of the originating router's interface.
Step 4	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Monitoring MSDP

Perform this optional task to monitor MSDP SA messages, peers, state, and peer status.

SUMMARY STEPS

1. **enable**
2. **debug ip msdp** [*peer-address* | *peer-name*] [**detail**] [**routes**]
3. **debug ip msdp resets**
4. **show ip msdp count** [*as-number*]
5. **show ip msdp peer** [*peer-address* | *peer-name*]
6. **show ip msdp sa-cache** [*group-address* | *source-address* | *group-name* | *source-name*] [*as-number*]
7. **show ip msdp summary**

DETAILED STEPS

Step 1 **enable**

Use this command to enable privileged EXEC mode. Enter your password if prompted.

```
Router# enable
```

Step 2 **debug ip msdp** [*peer-address* | *peer-name*] [**detail**] [**routes**]

Use this command to debug MSDP activity.

Use the optional *peer-address* or *peer-name* argument to specify for which peer debug events are logged.

The following is sample output from the **debug ip msdp** command:

```
Router# debug ip msdp

MSDP debugging is on
Router#
MSDP: 224.150.44.254: Received 1388-byte message from peer
MSDP: 224.150.44.254: SA TLV, len: 1388, ec: 115, RP: 172.31.3.92
MSDP: 224.150.44.254: Peer RPF check passed for 172.31.3.92, used EMBGP peer
MSDP: 224.150.44.250: Forward 1388-byte SA to peer
MSDP: 224.150.44.254: Received 1028-byte message from peer
MSDP: 224.150.44.254: SA TLV, len: 1028, ec: 85, RP: 172.31.3.92
MSDP: 224.150.44.254: Peer RPF check passed for 172.31.3.92, used EMBGP peer
MSDP: 224.150.44.250: Forward 1028-byte SA to peer
MSDP: 224.150.44.254: Received 1388-byte message from peer
MSDP: 224.150.44.254: SA TLV, len: 1388, ec: 115, RP: 172.31.3.111
MSDP: 224.150.44.254: Peer RPF check passed for 172.31.3.111, used EMBGP peer
MSDP: 224.150.44.250: Forward 1388-byte SA to peer
MSDP: 224.150.44.250: Received 56-byte message from peer
MSDP: 224.150.44.250: SA TLV, len: 56, ec: 4, RP: 192.168.76.241
MSDP: 224.150.44.250: Peer RPF check passed for 192.168.76.241, used EMBGP peer
MSDP: 224.150.44.254: Forward 56-byte SA to peer
MSDP: 224.150.44.254: Received 116-byte message from peer
MSDP: 224.150.44.254: SA TLV, len: 116, ec: 9, RP: 172.31.3.111
MSDP: 224.150.44.254: Peer RPF check passed for 172.31.3.111, used EMBGP peer
MSDP: 224.150.44.250: Forward 116-byte SA to peer
MSDP: 224.150.44.254: Received 32-byte message from peer
MSDP: 224.150.44.254: SA TLV, len: 32, ec: 2, RP: 172.31.3.78
MSDP: 224.150.44.254: Peer RPF check passed for 172.31.3.78, used EMBGP peer
MSDP: 224.150.44.250: Forward 32-byte SA to peer
```

Step 3 **debug ip msdp resets**

Use this command to debug MSDP peer reset reasons.

```
Router# debug ip msdp resets
```

Step 4 **show ip msdp count** [*as-number*]

Use this command to display the number of sources and groups originated in MSDP SA messages and the number of SA messages from an MSDP peer in the SA cache. The **ip msdp cache-sa-state** command must be configured for this command to produce any output.

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

```
Router# show ip msdp count

SA State per Peer Counters, <Peer>: <# SA learned>
    192.168.4.4: 8

SA State per ASN Counters, <asn>: <# sources>/<# groups>
    Total entries: 8
    ?: 8/8
```

Step 5 **show ip msdp peer** [*peer-address* | *peer-name*]

Use this command to display detailed information about MSDP peers.

Use the optional *peer-address* or *peer-name* arguments to display information about a particular peer.

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

```
Router# show ip msdp peer 192.168.4.4

MSDP Peer 192.168.4.4 (?), AS 64512 (configured AS)
  Connection status:
    State: Up, Resets: 0, Connection source: Loopback0 (2.2.2.2)
    Uptime(Downtime): 00:07:55, Messages sent/received: 8/18
    Output messages discarded: 0
    Connection and counters cleared 00:08:55 ago
  SA Filtering:
    Input (S,G) filter: none, route-map: none
    Input RP filter: none, route-map: none
    Output (S,G) filter: none, route-map: none
    Output RP filter: none, route-map: none
  SA-Requests:
    Input filter: none
  Peer ttl threshold: 0
  SAs learned from this peer: 8
  Input queue size: 0, Output queue size: 0
  MD5 signature protection on MSDP TCP connection: not enabled
```

Step 6 **show ip msdp sa-cache** [*group-address* | *source-address* | *group-name* | *source-name*] [*as-number*]

Use this command to display the (S, G) state learned from MSDP peers.

The following is sample output from the **show ip msdp sa-cache** command:

```
Router# show ip msdp sa-cache

MSDP Source-Active Cache - 8 entries
(10.44.44.5, 239.232.1.0), RP 192.168.4.4, BGP/AS 64512, 00:01:20/00:05:32, Peer
192.168.4.4
(10.44.44.5, 239.232.1.1), RP 192.168.4.4, BGP/AS 64512, 00:01:20/00:05:32, Peer
192.168.4.4
(10.44.44.5, 239.232.1.2), RP 192.168.4.4, BGP/AS 64512, 00:01:19/00:05:32, Peer
192.168.4.4
(10.44.44.5, 239.232.1.3), RP 192.168.4.4, BGP/AS 64512, 00:01:19/00:05:32, Peer
192.168.4.4
(10.44.44.5, 239.232.1.4), RP 192.168.4.4, BGP/AS 64512, 00:01:19/00:05:32, Peer
192.168.4.4
(10.44.44.5, 239.232.1.5), RP 192.168.4.4, BGP/AS 64512, 00:01:19/00:05:32, Peer
192.168.4.4
(10.44.44.5, 239.232.1.6), RP 192.168.4.4, BGP/AS 64512, 00:01:19/00:05:32, Peer
192.168.4.4
(10.44.44.5, 239.232.1.7), RP 192.168.4.4, BGP/AS 64512, 00:01:19/00:05:32, Peer
192.168.4.4
```

Step 7 **show ip msdp summary**

Use this command to display MSDP peer status.

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

```
Router# show ip msdp summary

MSDP Peer Status Summary
Peer Address     AS    State    Uptime/  Reset SA    Peer Name
                                Downtime Count Count
192.168.4.4      4     Up       00:08:05 0     8     ?
```

Clearing MSDP Connections, Statistics, and SA Cache Entries

Perform this optional task to clear MSDP connections, statistics, or SA cache entries.

SUMMARY STEPS

1. **enable**
2. **clear ip msdp peer** [*peer-address* | *peer-name*]
3. **clear ip msdp statistics** [*peer-address* | *peer-name*]
4. **clear ip msdp sa-cache** [*group-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ip msdp peer** [*peer-address* | *peer-name*] **Example:** `Router# clear ip msdp peer`	Clears the TCP connection to the specified MSDP peer. • This command resets all MSDP message counters.
Step 3	**clear ip msdp statistics** [*peer-address* | *peer-name*] **Example:** `Router# clear ip msdp statistics`	Clears the TCP connection to the specified MSDP peer. • This command resets all MSDP message counters.
Step 4	**clear ip msdp sa-cache** [*group-address*] **Example:** `Router# clear ip msdp sa-cache`	Clears SA cache entries. • If the **clear ip msdp sa-cache** is specified the optional *group-address* argument or *source-address* argument, all SA cache entries are cleared. • Use the optional *group-address* argument to clear all SA cache entries associated with a specific group.

Enabling SNMP Monitoring of MSDP

Perform this optional task to enable Simple Network Management Protocol (SNMP) monitoring of MSDP.

MSDP MIB

The MSDP MIB describes managed objects that can be used to remotely monitor MSDP speakers using SNMP. The MSDP MIB module contains four scalar objects and three tables. The tables are the Requests table, the Peer table, and the Source-Active (SA) Cache table. The Cisco implementation supports the Peer table and SA Cache table only. The Requests table contains information used to determine which peer to send SA requests to. However, the MSDP implementation used in Cisco IOS software does not associate sending SA requests to peers with group addresses (or group address masks).

Note The MSDP-MIB.my file can be downloaded from the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml.

Note For more information about network monitoring using SNMP, refer to the "Configuring SNMP Support" module in the *Cisco IOS Network Management Configuration Guide*, Release 12.4T.

Prerequisites

- This task assumes that you have configured SNMP and MSDP on your devices.
- In each PIM-SM domain there should be a device that is configured as the MSDP speaker. This device must have SNMP and the MSDP MIB enabled.

Restrictions

- All MSDP-MIB objects are implemented as read-only.
- The Requests table is not supported in Cisco's implementation of the MSDP MIB.
- The msdpEstablished notification is not supported in Cisco's implementation of the MSDP MIB.

SUMMARY STEPS

1. **enable**
2. **snmp-server enable traps msdp**
3. **snmp-server host** *host* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **priv** | **noauth**]}] *community-string* [**udp-port** *port-number*] **msdp**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**snmp-server enable traps msdp** **Example:** `Router# snmp-server enable traps msdp`	Enables the sending of MSDP notifications for use with SNMP. **Note** The **snmp-server enable traps** command enables both traps and informs.
Step 3	**snmp-server host** *host* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **priv** | **noauth**]}] *community-string* [**udp-port** *port-number*] **msdp** **Example:** `Router# snmp-server host examplehost msdp`	Specifies the recipient (host) for MSDP traps or informs.
Step 4	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

You can compare the results of MSDP MIB notifications to the output from the Cisco IOS software by using the **show ip msdp summary** and **show ip msdp peer** commands on the appropriate router. You can also compare the results of these commands to the results from SNMP Get operations. You can verify SA cache table entries using the **show ip msdp sa-cache** command. Additional troubleshooting information, such as the local address of the connection, the local port, and the remote port, can be obtained using the output from the **debug ip msdp** command.

Configuration Examples for Using MSDP to Interconnect Multiple PIM-SM Domains

This section provides the following MSDP configuration examples:

Configuring an MSDP Peer: Example

The following example shows how to establish MSDP peering connections between three MSDP peers:

Router A

```
!
interface Loopback 0
 ip address 10.220.8.1 255.255.255.255
!
ip msdp peer 10.220.16.1 connect-source Loopback0
ip msdp peer 10.220.32.1 connect-source Loopback0
!
```

Router B

```
!
interface Loopback 0
 ip address 10.220.16.1 255.255.255.255
!
ip msdp peer 10.220.8.1 connect connect-source Loopback0
ip msdp peer 10.220.32.1 connect connect-source Loopback0
!
```

Router C

```
!
interface Loopback 0
 ip address 10.220.32.1 255.255.255.255
!
ip msdp peer 10.220.8.1 connect 10.220.8.1 connect-source Loopback0
ip msdp peer 10.220.16.1 connect 10.220.16.1 connect-source Loopback0
!
```

Configuring MSDP MD5 Password Authentication: Example

The following example shows how to enable MD5 password authentication for TCP connections between two MSDP peers:

Router A

```
!
ip msdp peer 10.3.32.154
ip msdp password peer 10.3.32.154 0 test
!
```

Router B

```
!
ip msdp peer 10.3.32.153
ip msdp password peer 10.3.32.153 0 test
!
```

Configuring MSDP Compliance with IETF RFC 3618: Example

The following example shows how to configure the MSDP peers at 10.10.2.4 and 10.20.1.2 to be compliant with peer-RPF forwarding rules specified in IETF RFC 3618:

```
ip msdp peer 10.10.2.4
ip msdp peer 10.20.1.2
ip msdp rpf rfc3618
```

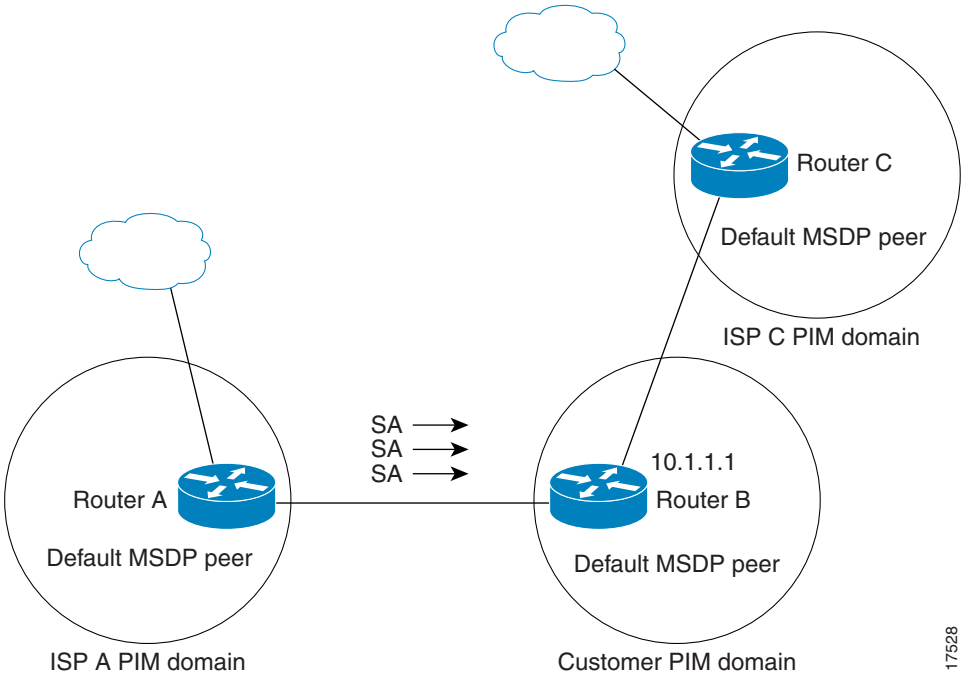
Configuring a Default MSDP Peer: Example

Figure 3 illustrates a scenario where default MSDP peers might be used. In the figure, a customer that owns Router B is connected to the internet through two Internet service providers (ISPs), one that owns Router A and the other that owns Router C. They are not running BGP or MBGP between them. In order for the customer to learn about sources in the ISP domain or in other domains, Router B identifies Router A as its default MSDP peer. Router B advertises SA messages to both Router A and Router C, but accepts SA messages either from Router A only or Router C only. If Router A is the first default peer in the configuration, it will be used if it is up and running. Only if Router A is not running will Router B accept SA messages from Router C.

The ISP will also likely use a prefix list to define which prefixes it will accept from the customer router. The customer will define multiple default peers, each having one or more prefixes associated with it.

The customer has two ISPs to use. The customer defines both ISPs as default peers. As long as the first default peer identified in the configuration is up and running, it will be the default peer and the customer will accept all SA messages it receives from that peer.

Figure 3 Default MSDP Peer Scenario



Router B advertises SAs to Router A and Router C, but uses only Router A or Router C to accept SA messages. If Router A is first in the configuration file, it will be used if it is up and running. Only when Router A is not running will Router B accept SAs from Router C. This is the behavior without a prefix list.

If you specify a prefix list, the peer will be a default peer only for the prefixes in the list. You can have multiple active default peers when you have a prefix list associated with each. When you do not have any prefix lists, you can configure multiple default peers, but only the first one is the active default peer as long as the router has connectivity to this peer and the peer is alive. If the first configured peer goes down or the connectivity to this peer goes down, the second configured peer becomes the active default, and so on.

The following example shows a partial configuration of Router A and Router C in Figure 3. Each of these ISPs may have more than one customer like the customer in Figure 3 that use default peering. In that case, they may have similar configurations. That is, they will only accept SAs from a default peer if the SA is permitted by the corresponding prefix list.

Router A Configuration

```
ip msdp default-peer 10.1.1.1
ip msdp default-peer 10.1.1.1 prefix-list site-b ge 32
ip prefix-list site-b permit 10.0.0.0/8
```

Router C Configuration

```
ip msdp default-peer 10.1.1.1 prefix-list site-b ge 32
ip prefix-list site-b permit 10.0.0.0/8
```

Configuring MSDP Mesh Groups: Example

The following example shows how to configure three routers to be fully meshed members of an MSDP mesh group:

Router A Configuration

```
ip msdp peer 10.2.2.2
ip msdp peer 10.3.3.3
ip msdp mesh-group test-mesh-group 10.2.2.2
ip msdp mesh-group test-mesh-group 10.3.3.3
```

Router B Configuration

```
ip msdp peer 10.1.1.1
ip msdp peer 10.3.3.3
ip msdp mesh-group test-mesh-group 10.1.1.1
ip msdp mesh-group test-mesh-group 10.3.3.3
```

Router C Configuration

```
ip msdp peer 10.1.1.1
ip msdp peer 10.2.2.2
ip msdp mesh-group test-mesh-group 10.1.1.1
ip msdp mesh-group test-mesh-group 10.2.2.2
```

Additional References

The following sections provide references related to using MSDP to interconnect PIM-SM domains.

Related Documents

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

Standards

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

MIBs

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

RFCs

RFCs	Title
RFC 2385	*Protection of BGP Sessions via the TCP MD5 Signature Option*
RFC 2858	*Multiprotocol Extensions for BGP-4*
RFC 3618	*Multicast Source Discovery Protocol (MSDP)*

Technical Assistance

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

Feature Information for Using MSDP to Interconnect Multiple PIM-SM Domains

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Releases 12.2(1) or 12.0(3)S or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the *IP Multicast Features Roadmap*.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Using MSDP to Interconnect Multiple PIM-SM Domains

Feature Name	Releases	Feature Information
MSDP Compliance with IETF RFC 3618	12.3(4)T 12.0(27)S 12.2(25)S 12.2(27)SBC 12.2(33)SRA 12.2(33)SXH	The MSDP Compliance with IETF RFC 3618 feature enables you to configure MSDP to comply with the peer-RPF forwarding rules defined in the IETF RFC 3618 specifications. Enabling the MSDP Compliance with IETF RFC 3618 feature prevents SA message loops. Additionally, enabling the MSDP Compliance with IETF RFC 3618 feature eliminates the requirement that BGP RRs run MSDP, enables the use of an IGP for the RPF check, and allows MSDP peerings between routers in nondirectly connected autonomous systems. The following sections provide information about this feature: • Configuring MSDP Compliance with IETF RFC 3618, page 19 • Configuring MSDP Compliance with IETF RFC 3618: Example, page 45 The following commands were introduced or modified by this feature: **ip msdp rpf rfc3618**, **show ip msdp rpf-peer**.

Table 1 Feature Information for Using MSDP to Interconnect Multiple PIM-SM Domains (continued)

Feature Name	Releases	Feature Information
MSDP MD5 Password Authentication	12.4(2)T 12.2(33)SXH	The MSDP MD5 password authentication feature is an enhancement to support MD5 signature protection on a TCP connection between two MSDP peers. This feature provides added security by protecting MSDP against the threat of spoofed TCP segments being introduced into the TCP connection stream. The following sections provide information about this feature: • Configuring MSDP MD5 Password Authentication Between MSDP Peers, page 12 • Configuring MSDP MD5 Password Authentication: Example, page 44 The following commands were introduced or modified by this feature: **ip msdp password peer**, **show ip msdp peer**.
MSDP compliance with IETF RFC 3618	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Configuring Source Specific Multicast

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

This chapter describes how to configure Source Specific Multicast (SSM). The Source Specific Multicast feature is an extension of IP multicast where datagram traffic is forwarded to receivers from only those multicast sources to which the receivers have explicitly joined. For multicast groups configured for SSM, only source-specific multicast distribution trees (no shared trees) are created.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release.

SSM Components Overview

SSM is a datagram delivery model that best supports one-to-many applications, also known as broadcast applications. SSM is a core networking technology for the Cisco implementation of IP multicast solutions targeted for audio and video broadcast application environments. This chapter discusses the following Cisco IOS components that support the implementation of SSM:

- Protocol Independent Multicast source specific mode (PIM-SSM)
- Internet Group Management Protocol Version 3 (IGMPv3)
- Internet Group Management Protocol Version 3 lite (IGMP v3lite)
- URL Rendezvous Directory (URD)

PIM-SSM is the routing protocol that supports the implementation of SSM and is derived from PIM sparse mode (PIM-SM). IGMP is the Internet Engineering Task Force (IETF) standards track protocol used for hosts to signal multicast group membership to routers. Version 3 of this protocol supports source filtering, which is required for SSM. To run SSM with IGMPv3, SSM must be supported in the Cisco IOS router, the host where the application is running, and the application itself. IGMP v3lite and

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

URD are two Cisco-developed transition solutions that enable the immediate development and deployment of SSM services, without the need to wait for the availability of full IGMPv3 support in host operating systems and SSM receiver applications. IGMP v3lite is a solution for application developers that allows immediate development of SSM receiver applications switching to IGMPv3 as soon as it becomes available. URD is a solution for content providers and content aggregators that enables them to deploy receiver applications that are not yet SSM enabled (through support for IGMPv3). IGMPv3, IGMP v3lite, and URD interoperate with each other, so that both IGMP v3lite and URD can easily be used as transitional solutions toward full IGMPv3 support in hosts.

How SSM Differs from Internet Standard Multicast

The current IP multicast infrastructure in the Internet and many enterprise intranets is based on the PIM-SM protocol and Multicast Source Discovery Protocol (MSDP). These protocols have proved to be reliable, extensive, and efficient. However, they are bound to the complexity and functionality limitations of the Internet Standard Multicast (ISM) service model. For example, with ISM, the network must maintain knowledge about which hosts in the network are actively sending multicast traffic. With SSM, this information is provided by receivers through the source addresses relayed to the last hop routers by IGMPv3, IGMP v3lite, or URD. SSM is an incremental response to the issues associated with ISM and is intended to coexist in the network with the protocols developed for ISM. In general, SSM provides a more advantageous IP multicast service for applications that utilize SSM.

ISM service is described in RFC 1112. This service consists of the delivery of IP datagrams from any source to a group of receivers called the multicast host group. The datagram traffic for the multicast host group consists of datagrams with an arbitrary IP unicast source address S and the multicast group address G as the IP destination address. Systems will receive this traffic by becoming members of the host group. Membership to a host group simply requires signalling the host group through IGMP Version 1, 2, or 3.

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 will receive this traffic by becoming members of the (S, G) channel. In both SSM and ISM, no signalling is required to become a source. However, in SSM, receivers must subscribe or unsubscribe to (S, G) channels to receive or not receive traffic from specific sources. In other words, receivers can receive traffic only from (S, G) channels to which they are subscribed, whereas in ISM, receivers need not know the IP addresses of sources from which they receive their traffic. The proposed standard approach for channel subscription signalling utilizes IGMP INCLUDE mode membership reports, which are supported only in IGMP Version 3.

SSM IP Address Range

SSM can coexist with the ISM service by applying the SSM delivery model to a configured subset of the IP multicast group address range. The Internet Assigned Numbers Authority (IANA) has reserved the address range 232.0.0.0 through 232.255.255.255 for SSM applications and protocols. Cisco IOS 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 will 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 or is SSM enabled through URD).

SSM Operations

An established network, in which IP multicast service is based on PIM-SM, can support SSM services. SSM can also be deployed alone in a network without the full range of protocols that are required for interdomain PIM-SM (for example, MSDP, Auto-RP, or bootstrap router [BSR]) if only SSM service is needed.

If SSM is deployed in a network already configured for PIM-SM (Cisco IOS Release 12.0 or later releases is recommended), then only the last hop routers must be upgraded to a Cisco IOS software image that supports SSM. Routers that are not directly connected to receivers do not have to upgrade to a Cisco IOS software image that supports SSM. In general, these nonlast hop routers must only run PIM-SM in the SSM range, and may need additional access control configuration to suppress MSDP signalling, registering, or PIM-SM shared tree operations from occurring within the SSM range.

The SSM mode of operation is enabled by configuring the SSM range through the **ip pim ssm** global configuration command. This configuration has the following effects:

- For groups within the SSM range, (S, G) channel subscriptions are accepted through IGMPv3 INCLUDE mode membership reports, IGMP v3lite, or URD (each of these methods must be configured on a per-interface basis). IGMP v3lite and URD (S, G) channel subscriptions are ignored for groups outside the SSM range.

  Both IGMP v3lite and URD are based on utilizing existing application IGMP group membership and extending it with their respective (S, G) channel subscription mechanism, which is ignored by Cisco IOS software outside the SSM range of addresses. Within the SSM range, IGMP Version 1 (IGMPv1) or Version 2 (IGMPv2) group membership reports or IGMPv3 EXCLUDE mode membership reports are acted upon only in conjunction with an (S, G) specific membership report from URD or IGMP v3lite.
- PIM operations within the SSM range of addresses change to PIM-SSM, a mode derived from PIM-SM. In this mode, only PIM (S, G) join and prune messages are generated by the router, and no (S, G) rendezvous point tree (RPT) or (*, G) RPT messages are generated. Incoming messages related to RPT operations are ignored or rejected and incoming PIM register messages are immediately answered with register-stop messages. PIM-SSM is backward compatible with PIM-SM, unless a router is a last hop router. Therefore, routers that are not last hop routers can run PIM-SM for SSM groups (for example, if they do not yet support SSM).
- No MSDP Source-Active (SA) messages within the SSM range will be accepted, generated, or forwarded.

IGMPv3 Host Signalling

IGMPv3 is the third version of the IETF standards track protocol in which hosts signal membership to last hop routers of multicast groups. IGMPv3 introduces the ability for hosts to signal group membership with filtering capabilities with respect to sources. A host can either signal that it wants to receive traffic from all sources sending to a group except for some specific sources (called EXCLUDE mode), or that it wants to receive traffic only from some specific sources sending to the group (called INCLUDE mode).

IGMPv3 can operate with both ISM and SSM. In ISM, both EXCLUDE and INCLUDE mode reports are applicable. In SSM, only INCLUDE mode reports are accepted by the last hop router. EXCLUDE mode reports are ignored.

For more information on IGMPv3, see the “Configuring IP Multicast Routing” chapter in this document.

IGMP v3lite Host Signalling

IGMP v3lite is a Cisco-developed transitional solution for application developers to immediately start programming SSM applications. It allows you to write and run SSM applications on hosts that do not yet support IGMPv3 in their operating system kernel.

Applications must be compiled with the Host Side IGMP Library (HSIL) for IGMP v3lite. This software provides applications with a subset of the IGMPv3 applications programming interface (API) that is required to write SSM applications. HSIL was developed for Cisco by Talarian and is available from the following web page:

http://www.talarianmulticast.com/cgi-bin/igmpdownld

One part of the HSIL is a client library linked to the SSM application. It provides the SSM subset of the IGMPv3 API to the SSM application. If possible, the library checks whether the operating system kernel supports IGMPv3. If it does, then the API calls simply are passed through to the kernel. If the kernel does not support IGMPv3, then the library uses the IGMP v3lite mechanism.

When using the IGMP v3lite mechanism, the library tells the operating system kernel to join to the whole multicast group, because joining to the whole group is the only method for the application to receive traffic for that multicast group (if the operating system kernel only supports IGMPv1 or IGMPv2). In addition, the library signals the (S, G) channel subscriptions to an IGMP v3lite server process, which is also part of the HSIL. A server process is needed because multiple SSM applications may be on the same host. This server process will then send IGMP v3lite-specific (S, G) channel subscriptions to the last hop Cisco IOS router, which needs to be enabled for IGMP v3lite. This Cisco IOS router will then "see" both the IGMPv1 or IGMPv2 group membership report from the operating system kernel and the (S, G) channel subscription from the HSIL daemon. If the router sees both of these messages, it will interpret them as an SSM (S, G) channel subscription and join to the channel through PIM-SSM. We recommend referring to the documentation accompanying the HSIL software for further information on how to utilize IGMP v3lite with your application.

IGMP v3lite is supported by Cisco only through the API provided by the HSIL, not as a function of the router independent of the HSIL. By default, IGMP v3lite is disabled. When IGMP v3lite is configured through the **ip igmp v3lite** interface configuration command on an interface, it will be active only for IP multicast addresses in the SSM range.

URD Host Signalling

URD is a Cisco-developed transitional solution that allows existing IP multicast receiver applications to be used with SSM without the need to modify the application and change or add any software on the receiver host running the application. URD is a content provider solution in which the receiver applications can be started or controlled through a web browser.

URD operates by passing a special URL from the web browser to the last hop router. This URL is called a URD intercept URL. A URD intercept URL is encoded with the (S, G) channel subscription and has a format that allows the last hop router to easily intercept it.

As soon as the last hop router intercepts both an (S, G) channel subscription encoded in a URD intercept URL and sees an IGMP group membership report for the same multicast group from the receiver application, the last hop router will use PIM-SSM to join toward the (S, G) channel as long as the application maintains the membership for the multicast group G. The URD intercept URL is thus only needed initially to provide the last hop router with the address of the sources to join to.

A URD intercept URL has the following syntax:

`http://`*`webserver`*`:465/`*`path`*`?group=`*`group`*`&source=`*`source1`*`&...source=`*`sourceN`*`&`

The *webserver* string is the name or IP address to which the URL is targeted. This target need not be the IP address of an existing web server, except for situations where the web server wants to recognize that the last hop router failed to support the URD mechanism. The number 465 indicates the URD port. Port 465 is reserved for Cisco by the IANA for the URD mechanism so that no other applications can use this port.

When the browser of a host encounters a URD intercept URL, it will try to open a TCP connection to the web server on port 465. If the last hop router is enabled for URD on the interface where the router receives the TCP packets from the host, it will intercept all packets for TCP connections destined to port 465 independent of the actual destination address of the TCP connection (independent of the address of the web server). Once intercepted, the last hop router will "speak" a very simple subset of HTTP on this TCP connection, emulating a web server. The only HTTP request that the last hop router will understand and reply to is the following GET request:

```
GET argument HTTP/1.0
argument = /path?group=group&source=source1&...source=sourceN&
```

When it receives a GET command, the router tries to parse the argument according to this syntax to derive one or more (S, G) channel memberships. The *path* string of the argument is anything up to, but not including, the first question mark, and is ignored. The *group* and *source1* through *sourceN* strings are the IP addresses or fully qualified domain names of the channels for which this argument is a subscription request. If the argument matches the syntax shown, the router interprets the argument to be subscriptions for the channels (*source1*, *group*) through (*sourceN*, *group*).

The router will accept the channel subscriptions if the following conditions are met:

- The IP address of the multicast group is within the SSM range.
- The IP address of the host that originated the TCP connection is directly connected to the router.

If the channel subscription is accepted, the router will respond to the TCP connection with the following HTML page format:

```
HTTP/1.1 200 OK
Server:cisco IOS
Content-Type:text/html
<html>
<body>
Retrieved URL string successfully
</body>
</html>
```

If an error condition occurs, the <body> part of the returned HTML page will carry an appropriate error message. The HTML page is a by-product of the URD mechanism. This returned text may, depending on how the web pages carrying a URD intercept URL are designed, be displayed to the user or be sized so that the actual returned HTML page is invisible.

The primary effect of the URD mechanism is that the router will remember received channel subscriptions and will match them against IGMP group membership reports received by the host. The router will "remember" a URD (S, G) channel subscription for up to 3 minutes without a matching IGMP group membership report. As soon as the router sees that it has received both an IGMP group membership report for a multicast group G and a URD (S, G) channel subscription for the same group G, it will join the (S, G) channel through PIM-SSM. The router will then continue to join to the (S, G) channel based only on the presence of a continuing IGMP membership from the host. Thus, one initial URD channel subscription is all that is needed to be added through a web page to enable SSM with URD.

If the last hop router from the receiver host is not enabled for URD, then it will not intercept the HTTP connection toward the web server on port 465. This situation will result in a TCP connection to port 465 on the web server. If no further provisions on the web server are taken, then the user may see a notice

(for example, “Connection refused”) in the area of the web page reserved for displaying the URD intercept URL (if the web page was designed to show this output). It is also possible to let the web server “listen” to requests on port 465 and install a Common Gateway Interface (CGI) script that would allow the web server to know if a channel subscription failed (for example, to subsequently return more complex error descriptions to the user).

Because the router returns a Content-Type of text and HTML, the best way to include the URD intercept URL into a web page is to use a frame. By defining the size of the frame, you can also hide the URD intercept URL on the displayed page.

By default, URD is disabled on all interfaces. When URD is configured through the **ip urd** interface configuration command on an interface, it will be active only for IP multicast addresses in the SSM range.

Benefits

IP Multicast Address Management Not Required

In the ISM service, applications must acquire a unique IP multicast group address because traffic distribution is based only on the IP multicast group address used. If two applications with different sources and receivers use the same IP multicast group address, then receivers of both applications will receive traffic from the senders of both applications. Even though the receivers, if programmed appropriately, can filter out the unwanted traffic, this situation would cause generally unacceptable levels of unwanted traffic.

Allocating a unique IP multicast group address for an application is still a problem. Most short-lived applications use mechanisms like Session Description Protocol (SDP) and Session Announcement Protocol (SAP) to get a random address, a solution that does not work well with a rising number of applications in the Internet. The best current solution for long-lived applications is described in RFC 2770, but this solution suffers from the restriction that each autonomous system is limited to only 255 usable IP multicast addresses.

In SSM, traffic from each source is forwarded between routers in the network independent of traffic from other sources. Thus different sources can reuse multicast group addresses in the SSM range.

Denial of Service Attacks from Unwanted Sources Inhibited

In SSM, multicast traffic from each individual source will be transported across the network only if it was requested (through IGMPv3, IGMP v3lite, or URD memberships) from a receiver. In contrast, ISM forwards traffic from any active source sending to a multicast group to all receivers requesting that multicast group. In Internet broadcast applications, this ISM behavior is highly undesirable because it allows unwanted sources to easily disturb the actual Internet broadcast source by simply sending traffic to the same multicast group. This situation depletes bandwidth at the receiver side with unwanted traffic and thus disrupts the undisturbed reception of the Internet broadcast. In SSM, this type of denial of service (DoS) attack cannot be made by simply sending traffic to a multicast group.

Easy to Install and Manage

SSM is easy to install and provision in a network because it does not require the network to maintain which active sources are sending to multicast groups. This requirement exists in ISM (with IGMPv1, IGMPv2, or IGMPv3).

The current standard solutions for ISM service are PIM-SM and MSDP. Rendezvous point (RP) management in PIM-SM (including the necessity for Auto-RP or BSR) and MSDP is required only for the network to learn about active sources. This management is not necessary in SSM, which makes SSM easier than ISM to install and manage, and therefore easier than ISM to operationally scale in deployment. Another factor that contributes to the ease of installation of SSM is the fact that it can leverage preexisting PIM-SM networks and requires only the upgrade of last hop routers to support IGMPv3, IGMP v3lite, or URD.

Ideal for Internet Broadcast Applications

The three benefits previously described make SSM ideal for Internet broadcast-style applications for the following reasons:

- The ability to provide Internet broadcast services through SSM without the need for unique IP multicast addresses allows content providers to easily offer their service (IP multicast address allocation has been a serious problem for content providers in the past).
- The prevention against DoS attacks is an important factor for Internet broadcast services because, with their exposure to a large number of receivers, they are the most common targets for such attacks.
- The ease of installation and operation of SSM makes it ideal for network operators, especially in those cases where content needs to be forwarded between multiple independent PIM domains (because there is no need to manage MSDP for SSM between PIM domains).

Restrictions

Legacy Applications Within the SSM Range Restrictions

Existing applications in a network predating SSM will not work within the SSM range unless they are modified to support (S, G) channel subscriptions or are enabled through URD. Therefore, enabling SSM in a network may cause problems for existing applications if they use addresses within the designated SSM range.

IGMP v3lite and URD Require a Cisco IOS Last Hop Router

SSM and IGMPv3 are solutions that are being standardized in the IETF. However, IGMP v3lite and URD are Cisco-developed solutions. For IGMP v3lite and URD to operate properly for a host, the last hop router toward that host must be a Cisco IOS router with IGMP v3lite or URD enabled.

Note This limitation does not apply to an application using the HSIL if the host has kernel support for IGMPv3, because then the HSIL will use the kernel IGMPv3 instead of IGMP v3lite.

Address Management Restrictions

Address management is still necessary to some degree when SSM is used with Layer 2 switching mechanisms. Cisco Group Management Protocol (CGMP), IGMP snooping, or Router-Port Group Management Protocol (RGMP) currently support only group-specific filtering, not (S, G) channel-specific filtering. If different receivers in a switched network request different (S, G) channels sharing the same group, then they will not benefit from these existing mechanisms. Instead, both receivers will receive all (S, G) channel traffic (and filter out the unwanted traffic on input). Because of the ability of SSM to reuse the group addresses in the SSM range for many independent applications, this situation can lead to less than expected traffic filtering in a switched network. For this reason it is important to follow the recommendations set forth in the IETF drafts for SSM to use random IP addresses out of the SSM range for an application to minimize the chance for reuse of a single address within the SSM range between different applications. For example, an application service providing a set of television channels should, even with SSM, use a different group for each television (S, G) channel. This setup will guarantee that multiple receivers to different channels within the same application service will never experience traffic aliasing in networks that include Layer 2 switches.

IGMP Snooping and CGMP Limitations

IGMPv3 uses new membership report messages that may not be recognized correctly by older IGMP Snooping switches, in which case hosts will not properly receive traffic. This situation is not an issue if URD or IGMP v3lite is used with hosts where the operating system is not upgraded for IGMPv3, because IGMP v3lite and URD rely only on IGMPv1 or IGMPv2 membership reports. For more information about switching issues related to IGMP (especially with CGMP), refer to the "Configuring IGMP Version 3" section of the "Configuring IP Multicast Routing" chapter in this document.

URD Intercept URL Limitations

A URD intercept URL string must be fewer than 256 bytes in length, starting from the */path* argument. In the HTTP/TCP connection, this string must also be contained within a single TCP/IP packet. For example, for a 256-byte string, a link maximum transmission unit (MTU) of 128 bytes between the host and intercepting router would cause incorrect operation of URD.

State Maintenance Limitations

In PIM-SSM, the last hop router will continue to periodically send (S, G) join messages if appropriate (S, G) subscriptions are on the interfaces. Therefore, as long as receivers send (S, G) subscriptions, the shortest path tree (SPT) state from the receivers to the source will be maintained, even if the source is not sending traffic for longer periods of time (or even never).

This case is opposite to PIM-SM, where (S, G) state is maintained only if the source is sending traffic and receivers are joining the group. If a source stops sending traffic for more than 3 minutes in PIM-SM, the (S, G) state will be deleted and only reestablished after packets from the source arrive again through the RPT. Because no mechanism in PIM-SSM notifies a receiver that a source is active, the network must maintain the (S, G) state in PIM-SSM as long as receivers are requesting receipt of that channel.

HSIL Limitations

As explained in the "IGMP v3lite Host Signalling" section, the HSIL tries to determine if the host operating system supports IGMPv3. This check is made so that a single application can be used both on hosts where the operating system has been upgraded to IGMPv3 and on hosts where the operating system only supports IGMPv1 or IGMPv2. Checking for the availability of IGMPv3 in the host operating system can only be made by the HSIL if IGMPv3 kernel support exists for at least one version of this operating system at the time when the HSIL was provided. If such an IGMPv3 kernel implementation has become available only recently, then users may need to also upgrade the HSIL on their hosts so that applications compiled with the HSIL will then dynamically bind to the newest version of the HSIL, which should support the check for IGMPv3 in the operating system kernel. Upgrading the HSIL can be done independently of upgrading the application itself.

SSM Configuration Task List

To configure SSM, perform the tasks described in the following sections. The tasks in the first section are required; the tasks in the remaining section are optional.

- Configuring SSM (Required)
- Monitoring SSM (Optional)

Configuring SSM

To configure SSM, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip pim ssm [default | range access-list]`	Defines the SSM range of IP multicast addresses.
Step 2	`Router(config)# interface type number`	Selects an interface that is connected to hosts on which IGMPv3, IGMP v3lite, and URD can be enabled.
Step 3	`Router(config-if)# ip pim {sparse-mode | sparse-dense-mode}`	Enables PIM on an interface. You must use either sparse mode or sparse-dense mode.
Step 4	`Router(config-if)# ip igmp version 3`	Enables IGMPv3 on this interface. The default version of IGMP is set to Version 2.
	or	or
	`Router(config-if)# ip igmp v3lite`	Enables the acceptance and processing of IGMP v3lite membership reports on an interface.
	or	or
	`Router(config-if)# ip urd`	Enables interception of TCP packets sent to the reserved URD port 465 on an interface and processing of URD channel subscription reports.

Monitoring SSM

To monitor SSM, use the following commands in privileged EXEC mode, as needed:

Command	Purpose
`Router# show ip igmp groups detail`	Displays the (S, G) channel subscription through IGMPv3, IGMP v3lite, or URD.
`Router# show ip mroute`	Displays whether a multicast group supports SSM service or whether a source-specific host report was received.

SSM Configuration Examples

This section provides the following SSM configuration examples:

- SSM with IGMPv3 Example
- SSM with IGMP v3lite and URD Example
- SSM Filtering Example

SSM with IGMPv3 Example

The following example shows how to configure a router (running IGMPv3) for SSM:

```
ip multicast-routing
!
interface Ethernet3/1
 ip address 172.21.200.203 255.255.255.0
 description backbone interface
 ip pim sparse-dense-mode
!
interface Ethernet3/2
 ip address 131.108.1.2 255.255.255.0
 ip pim sparse-dense-mode
 description ethernet connected to hosts
 ip igmp version 3
!
ip pim ssm default
```

SSM with IGMP v3lite and URD Example

The following example shows how to configure IGMP v3lite and URD on interfaces connected to hosts for SSM. Configuring IGMP v3lite and URD is not required or recommended on backbone interfaces.

```
interface ethernet 3/1
 ip address 172.21.200.203 255.255.255.0
 ip pim sparse-dense-mode
 description ethernet connected to hosts
!
interface ethernet 1
description ethernet connected to hosts
 ip address 131.108.1.2 255.255.255.0
 ip pim sparse-dense-mode
 ip urd
 ip igmp v3lite
```

SSM Filtering Example

The following example shows how to configure filtering on a legacy RP router running Cisco IOS releases earlier than Release 12.1(3)T for SSM routing. This filtering will suppress all unwanted PIM-SM and MSDP traffic in the SSM range. Without this filtering, SSM will still operate, but there may be additional RPT traffic if legacy first hop and last hop routers exist in the network.

```
ip access-list extended  no-ssm-range
  deny   ip any 232.0.0.0 0.255.255.255 ! SSM range
  permit ip any any
! Deny registering in SSM range
ip pim accept-register list no-ssm-range

ip access-list extended msdp-nono-list
  deny   ip any 232.0.0.0 0.255.255.255 ! SSM Range
  ! .
  ! .
  ! .
  ! See ftp://ftpeng.cisco.com/ipmulticast/config-notes/msdp-sa-filter.txt for other SA
  ! messages that typically need to be filtered.
  permit ip any any

! Filter generated SA messages in SSM range. This configuration is only needed if there
! are directly connected sources to this router. The "ip pim accept-register" command
! filters remote sources.
ip msdp redistribute list msdp-nono-list

! Filter received SA messages in SSM range. "Filtered on receipt" means messages are
! neither processed or forwarded. Needs to be configured for each MSDP peer.
ip msdp sa-filter in msdp-peer1 list msdp-nono-list
! .
! .
! .
ip msdp sa-filter in msdp-peerN list msdp-nono-list
```

Source Specific Multicast (SSM) Mapping

The Source Specific Multicast (SSM) Mapping feature extends the Cisco IOS suite of SSM transition tools, which also includes URL Rendezvous Directory (URD) and Internet Group Management Protocol Version 3 Lite (IGMP v3lite). SSM mapping supports SSM transition in cases where neither URD nor IGMP v3lite is available, or when supporting SSM on the end system is impossible or unwanted due to administrative or technical reasons. SSM mapping enables you to leverage SSM for video delivery to legacy set-top boxes (STBs) that do not support IGMPv3 or for applications that do not take advantage of the IGMPv3 host stack.

History for the SSM Mapping Feature

Feature History	
Release	**Modification**
12.3(2)T	This feature was introduced.
12.2(18)S	This feature was integrated into Cisco IOS Release 12.2(18)S.
12.2(18)SXD3	This feature was integrated into Cisco IOS Release 12.2(18)SXD3.
12.2(27)SBC	This feature was integrated into Cisco IOS Release 12.2(27)SBC.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for SSM Mapping

- One option available for using SSM mapping is to install it together with a Domain Name System (DNS) server to simplify administration of the SSM Mapping feature in larger deployments.

  Before you can configure and use SSM mapping with DNS lookups, you need to be able to add records to a running DNS server. If you do not already have a DNS server running, you need to install one. The Cisco IOS software does not provide for DNS server functionality. You may want to use a product such as Cisco Network Registrar (CNR).

Restrictions for SSM Mapping

- The SSM Mapping feature does not share the benefit of full SSM (unlike URD or IGMP v3lite). Because SSM mapping takes a group G join from a host and identifies this group with an application associated with one or more sources, it can only support one such application per group G.

  Nevertheless, full SSM applications may still share the same group also used in SSM mapping. That is, SSM mapping is compatible with simultaneous URD, IGMP v3lite or IGMPv3 membership reports.

- Enable IGMPv3 with care on the last hop router when you rely solely on SSM mapping as a transition solution for full SSM.

  When both SSM mapping and IGMPv3 are enabled, the router will send out IGMPv3 membership query messages instead of IGMPv3 membership messages. If the receiver hosts that are to be supported with SSM mapping can only support IGMPv1 or IGMPv2, then enabling SSM mapping on an interface with IGMPv3 is fine. IGMPv3 membership query messages will be interpreted as IGMPv1 or IGMPv2 queries and the host will continue to report with IGMPv1 or IGMPv2 reports.

  However, when both SSM mapping and IGMPv3 are enabled and the hosts already support IGMPv3 (but not SSM), then they will start to send IGMPv3 group reports. These IGMPv3 group reports are not supported with SSM mapping and the router will not correctly associate sources with these reports.

Information About SSM Mapping

To configure the SSM Mapping feature, you should understand the following concepts:

SSM Components

SSM is a datagram delivery model that best supports one-to-many applications, also known as broadcast applications. SSM is a core networking technology for the Cisco implementation of IP multicast solutions targeted for audio and video broadcast application environments and is described in RFC 3569. The following two Cisco IOS components together support the implementation of SSM:

- Protocol Independent Multicast source-specific mode (PIM-SSM)
- Internet Group Management Protocol Version 3 (IGMPv3)

PIM-SSM is the routing protocol that supports the implementation of SSM and is derived from PIM sparse mode (PIM-SM). IGMP is the Internet Engineering Task Force (IETF) standards track protocol used for hosts to signal multicast group membership to routers. IGMPv3 supports source filtering, which is required for SSM. For SSM to run with IGMPv3, SSM must be supported in the Cisco IOS router, the host where the application is running, and the application itself.

SSM Benefits

IP Multicast Address Management

In the Internet Standard Multicast (ISM) service, applications must acquire a unique IP multicast group address because traffic distribution is based only on the IP multicast group address used. If two applications with different sources and receivers use the same IP multicast group address, receivers of both applications will receive traffic from the senders of both applications. Even though the receivers, if programmed appropriately, can filter out the unwanted traffic, this situation would cause generally unacceptable levels of unwanted traffic.

Allocating a unique IP multicast group address for an application is problematic. Most short-lived applications use mechanisms like Session Description Protocol (SDP) and Session Announcement Protocol (SAP) to get a random address, a solution that does not work well with a rising number of applications in the Internet. The best current solution for long-lived applications is described in RFC 2770, but this solution suffers from the restriction that each autonomous system is limited to only 255 usable IP multicast addresses.

In SSM, traffic from each source is forwarded among routers in the network independently of traffic from other sources. Thus different sources can reuse multicast group addresses in the SSM range.

Inhibition of Denial of Service Attacks

In SSM, multicast traffic from each individual source will be transported across the network only if it was requested (through IGMPv3 or URD memberships) from a receiver. In contrast, ISM forwards traffic from any active source sending to a multicast group to all receivers requesting that multicast group. In Internet broadcast applications, this ISM behavior is highly undesirable because it allows unwanted sources to easily disturb the actual Internet broadcast source by simply sending traffic to the same multicast group. This situation depletes bandwidth at the receiver side with unwanted traffic and thus disrupts the reception of the Internet broadcast. In SSM, this type of denial-of-service (DoS) attack cannot be made by simply sending traffic to a multicast group.

Installation and Management

SSM is easy to install and provision in a network because it does not require the network to maintain information about which active sources are sending to multicast groups. This requirement exists in ISM (with IGMPv1, IGMPv2, or IGMPv3).

The current standard solutions for ISM service are PIM-SM and Multicast Source Discovery Protocol (MSDP). Rendezvous point (RP) management in PIM-SM (including the necessity for Auto-RP or bootstrap router [BSR]) and MSDP is required only for the network to learn about active sources. This management is not necessary in SSM. SSM is therefore easier than ISM to install and manage and easier to operationally scale in deployment. Another factor that contributes to the ease of installation of SSM is the fact that it can leverage preexisting PIM-SM networks.

Internet Broadcast Applications

The three benefits listed above make SSM ideal for Internet broadcast-style applications for the following reasons:

- The ability to provide Internet broadcast services through SSM without the need for unique IP multicast addresses allows content providers to easily offer their service. IP multicast address allocation has been a serious problem for content providers in the past.
- The prevention of DoS attacks is an important factor for Internet broadcast services because, with their exposure to a large number of receivers, they are the most common targets for such attacks.
- The ease of installation and operation of SSM makes it ideal for network operators, especially in those cases where content needs to be forwarded between multiple independent PIM domains (because there is no need to manage MSDP for SSM between PIM domains).

SSM Transition Solutions

The Cisco IOS suite of SSM transition solutions consists of the following transition solutions that enable the immediate development and deployment of SSM services, without the need to wait for the availability of full IGMPv3 support in host operating systems and SSM receiver applications:

- Internet Group Management Protocol Version 3 lite (IGMP v3lite)
- URL Rendezvous Directory (URD)
- SSM mapping

IGMP v3lite is a solution for application developers that allows immediate development of SSM receiver applications switching to IGMPv3 as soon as it becomes available.

For more information about IGMP v3lite, see the "Configuring Source Specific Multicast" module.

URD is an SSM transition solution for content providers and content aggregators that allows them to deploy receiver applications that are not yet SSM enabled (through support for IGMPv3) by enabling the receiving applications to be started and controlled through a web browser.

For more information about URD, see the see the "Configuring Source Specific Multicast" module.

SSM mapping supports SSM transition in cases where neither URD nor IGMP v3lite are available, or when supporting SSM on the end system is impossible or unwanted due to administrative or technical reasons.

SSM Mapping Overview

SSM mapping supports SSM transition in cases where neither URD nor IGMP v3lite is available, or when supporting SSM on the end system is impossible or unwanted due to administrative or technical reasons. Using SSM to deliver live streaming video to legacy STBs that do not support IGMPv3 is a typical application of SSM mapping.

Prior to the introduction of SSM mapping, the following conditions would have prevented SSM transition in the case of legacy STB deployments with STB receivers that only support IGMPv1 or IGMPv2:

- The operating system on the receivers do not support IGMPv3; thus, IGMPv3 cannot be used to support SSM.
- Moreover, the application running on the receivers cannot be upgraded to support SSM; thus, IGMPv3 lite cannot be used to support SSM transition.
- To further exacerbate the issue, the application itself cannot be started through a web browser; thus, URD cannot be used to support SSM transition.

SSM mapping provides an SSM transition solution for hosts and applications that meet those conditions.

In a typical STB deployment, each TV channel uses one separate IP multicast group and has one active server host sending the TV channel. A single server may of course send multiple TV channels, but each to a different group. In this network environment, if a router receives an IGMPv1 or IGMPv2 membership report for a particular group G, the report implicitly addresses the well-known TV server for the TV channel associated with the multicast group.

SSM mapping introduces a means for the last hop router to discover sources sending to groups. When SSM mapping is configured, if a router receives an IGMPv1 or IGMPv2 membership report for a particular group G, the router translates this report into one or more (S, G) channel memberships for the well-known sources associated with this group.

Note As is the case for the other SSM transition solutions (URD and IGMP v3lite), SSM mapping only needs to be configured on the last hop router connected to receivers. No support is needed on any other routers in the network. SSM mapping, in addition, is fully compatible with IGMPv3, IGMP v3lite, and URD.

When the router receives an IGMPv1 or IGMPv2 membership report for group G, the router uses SSM mapping to determine one or more source IP addresses for group G. SSM mapping then translates the membership report as an IGMPv3 report INCLUDE (G, [S1, G], [S2, G]...[S*n*, G] and continues as if it had received an IGMPv3 report. The router then sends out PIM joins toward (S1, G) to (S*n*, G) and continues to be joined to these groups as long as it continues to receive the IGMPv1 or IGMPv2 membership reports and as long as the SSM mapping for the group remains the same. SSM mapping, thus, enables you to leverage SSM for video delivery to legacy STBs that do not support IGMPv3 or for applications that do not take advantage of the IGMPv3 host stack.

SSM mapping enables the last hop router to determine the source addresses either by a statically configured table on the router or by consulting a DNS server. When the statically configured table is changed, or when the DNS mapping changes, the router will leave the current sources associated with the joined groups.

Static SSM Mapping

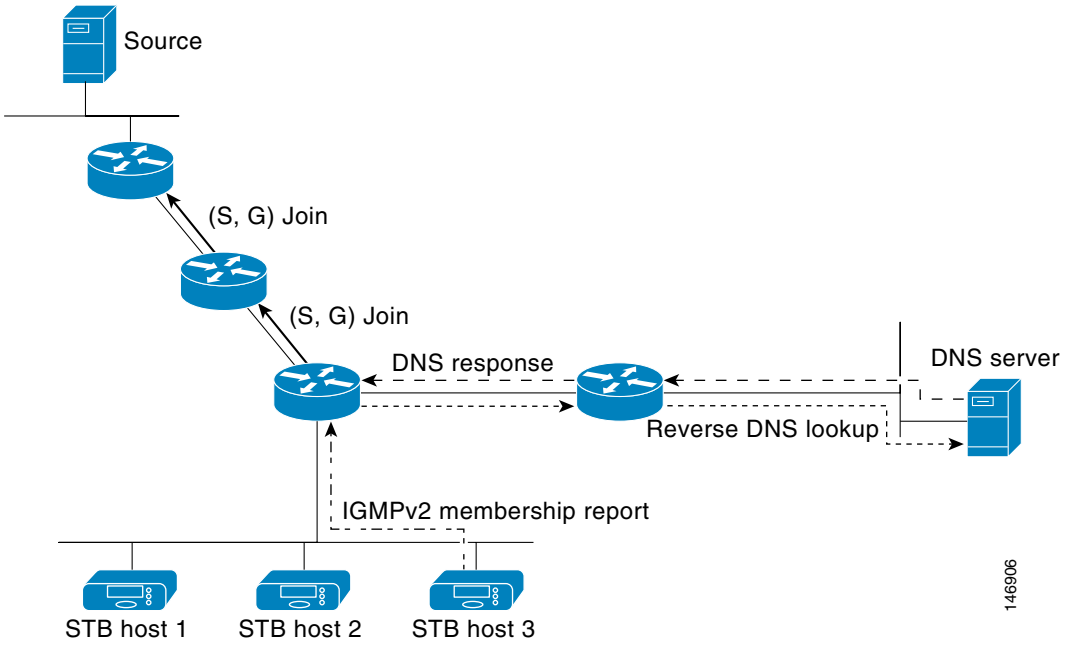
SSM static mapping enables you to configure the last hop router to use a static map to determine the sources sending to groups. Static SSM mapping requires that you configure access lists (ACLs) to define group ranges. The groups permitted by those ACLs then can be mapped to sources using the **ip igmp static ssm-map** command.

You can configure static SSM mapping in smaller networks when a DNS is not needed or to locally override DNS mappings that may be temporarily incorrect. When configured, static SSM mappings take precedence over DNS mappings.

DNS-Based SSM Mapping

DNS-based SSM mapping enables you to configure the last hop router to perform a reverse DNS lookup to determine sources sending to groups (see Figure 1). When DNS-based SSM mapping is configured, the router constructs a domain name that includes the group address G and performs a reverse lookup into the DNS. The router looks up IP address resource records (IP A RRs) to be returned for this constructed domain name and uses the returned IP addresses as the source addresses associated with this group. SSM mapping supports up to 20 sources for each group. The router joins all sources configured for a group.

Figure 1 DNS-Based SSM-Mapping



The SSM mapping mechanism that enables the last hop router to join multiple sources for a group can be used to provide source redundancy for a TV broadcast. In this context, the redundancy is provided by the last hop router using SSM mapping to join two video sources simultaneously for the same TV channel. However, to prevent the last hop router from duplicating the video traffic, it is necessary that the video sources utilize a server-side switchover mechanism where one video source is active while the other backup video source is passive. The passive source waits until an active source failure is detected before sending the video traffic for the TV channel. The server-side swichover mechanism, thus, ensures that only one of the servers is actively sending the video traffic for the TV channel.

To look up one or more source addresses for a group G that includes G1, G2, G3, and G4, the following DNS resource records (RRs) must be configured on the DNS server:

G4.G3.G2.G1 [*multicast-domain*] [*timeout*]	IN A *source-address-1*
	IN A *source-address-2*
	IN A *source-address-n*

The *multicast-domain* argument is a configurable DNS prefix. The default DNS prefix is in-addr.arpa. You should only use the default prefix when your installation is either separate from the internet or if the group names that you map are global scope group addresses (RFC 2770 type addresses that you configure for SSM) that you own.

The *timeout* argument configures the length of time for which the router performing SSM mapping will cache the DNS lookup. This argument is optional and defaults to the timeout of the zone in which this entry is configured. The timeout indicates how long the router will keep the current mapping before querying the DNS server for this group. The timeout is derived from the cache time of the DNS RR entry and can be configured for each group/source entry on the DNS server. You can configure this time for larger values if you want to minimize the number of DNS queries generated by the router. Configure this time for a low value if you want to be able to quickly update all routers with new source addresses.

Note Refer to your DNS server documentation for more information about configuring DNS RRs.

To configure DNS-based SSM mapping in Cisco IOS software, you must configure a few global commands but no per-channel specific configuration is needed. There is no change to the Cisco IOS configuration for SSM mapping if additional channels are added. When DNS-based SSM mapping is configured, the mappings are handled entirely by one or more DNS servers. All DNS techniques for configuration and redundancy management can be applied to the entries needed for DNS-based SSM mapping.

SSM Mapping Benefits

- The SSM Mapping feature provides almost the same ease of network installation and management as a pure SSM solution based on IGMPv3. Some additional configuration is necessary to enable SSM mapping.
- The SSM benefit of inhibition of DoS attacks applies when SSM mapping is configured. When SSM mapping is configured the only segment of the network that may still be vulnerable to DoS attacks are receivers on the LAN connectioned to the last hop router. Since those receivers may still be using IGMPv1 and IGMPv2, they are vulnerable to attacks from unwanted sources on the same LAN. SSM mapping, however, does protect those receivers (and the network path leading towards them) from multicast traffic from unwanted sources anywhere else in the network.

- Address assignment within a network using SSM mapping needs to be coordinated, but it does not need assignment from outside authorities, even if the content from the network is to be transited into other networks.

How to Configure SSM Mapping

This section contains the following tasks:

Configuring Static SSM Mapping

Perform this task to configure the last hop router in an SSM deployment to use static SSM mapping to determine the IP addresses of sources sending to groups.

Prerequisites

- Enable IP multicast routing, enable PIM sparse mode, and configure SSM before performing this task. For more information, see the "Configuring Basic Multicast" module.
- Before you configure static SSM mapping, you must configure ACLs that define the group ranges to be mapped to source addresses.

  For information about how to configure an ACL, see the "Creating an IP Access List and Applying It to an Interface" module.

Restrictions

- Enable IGMPv3 with care on the last hop router when you rely solely on SSM mapping as a transition solution for full SSM.

  When both SSM mapping and IGMPv3 are enabled, the router will send out IGMPv3 membership query messages instead of IGMPv3 membership messages. If the receiver hosts that are to be supported with SSM mapping can only support IGMPv1 or IGMPv2, then enabling SSM mapping on an interface with IGMPv3 is fine. IGMPv3 reports will be interpreted as IGMPv1 or IGMPv2 queries and the host will continue to report with IGMPv1 or IGMPv2 reports.

  However, when both SSM mapping and IGMPv3 are enabled, if the hosts already support IGMPv3 (but not SSM), then they will start to send IGMPv3 group reports. These IGMPv3 group reports are not supported with SSM mapping and the router will not correctly associate sources with these reports.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp ssm-map enable**

4. **no ip igmp ssm-map query dns**
5. **ip igmp ssm-map static** *access-list source-address*
6. Repeat Step 5 to configure additional static SSM mappings, if required.
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip igmp ssm-map enable** **Example:** `Router(config)# ip igmp ssm-map enable`	Enables SSM mapping for groups in the configured SSM range. **Note** By default, this command enables DNS-based SSM mapping.
Step 4	**no ip igmp ssm-map query dns** **Example:** `Router(config)# no ip igmp ssm-map query dns`	(Optional) Disables DNS-based SSM mapping. **Note** Disable DNS-based SSM mapping if you only want to rely on static SSM mapping. By default, the **ip igmp ssm-map** command enables DNS-based SSM mapping.
Step 5	**ip igmp ssm-map static** *access-list source-address* **Example:** `Router(config)# ip igmp ssm-map static 11 172.16.8.11`	Configures static SSM mapping. • The ACL supplied for the *access-list* argument defines the groups to be mapped to the source IP address entered for the *source-address* argument. **Note** You can configure additional static SSM mappings. If additional SSM mappings are configured and the router receives an IGMPv1 or IGMPv2 membership report for a group in the SSM range, the Cisco IOS software determines the source addresses associated with the group by walking each configured **ip igmp ssm-map static** command. The Cisco IOS software associates up to 20 sources per group.
Step 6	Repeat Step 5 to configure additional static SSM mappings, if required.	—
Step 7	**end** **Example:** `Router(config)# end`	Ends the current configuration session and returns to privileged EXEC mode.

What to Do Next

Proceed to the "Configuring DNS-Based SSM Mapping" section on page 10 or to the "Verifying SSM Mapping Configuration and Operation" section on page 13.

Configuring DNS-Based SSM Mapping

Perform this task to configure the last hop router to perform DNS lookups to learn the IP addresses of sources sending to a group.

Prerequisites

- Enable IP multicast routing, enable PIM sparse mode, and configure SSM before performing this task. For more information, see the "Configuring Basic Multicast" module.
- Before you can configure and use SSM mapping with DNS lookups, you need to be able to add records to a running DNS server. If you do not already have a DNS server running, you need to install one. The Cisco IOS software does not provide for DNS server functionality. You may want to use a product such as Cisco Network Registrar (CNR).

Restrictions

- Enable IGMPv3 with care on the last hop router when you rely solely on SSM mapping as a transition solution for full SSM.

  When both SSM mapping and IGMPv3 are enabled, the router will send out IGMPv3 membership query messages instead of IGMPv3 membership messages. If the receiver hosts that are to be supported with SSM mapping can only support IGMPv1 or IGMPv2, then enabling SSM mapping on an interface with IGMPv3 is fine. IGMPv3 reports will be interpreted as IGMP version 1 or IGMP version 2 queries and the host will continue to report with IGMPv1 or IGMPv2 reports.

  However, when both SSM mapping and IGMPv3 are enabled, if the hosts already support IGMPv3 (but not SSM), then they will start to send IGMPv3 group reports. These IGMPv3 group reports are not supported with SSM mapping and the router will not correctly associate sources with these reports.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp ssm-map enable**
4. **ip igmp ssm-map query dns**
5. **ip domain multicast** *domain-prefix*
6. **ip name-server** *server-address1* [*server-address2...server-address6*]
7. Repeat Step 6 to configure additional DNS servers for redundancy, if required.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip igmp ssm-map enable** **Example:** `Router(config)# ip igmp ssm-map enable`	Enables SSM mapping for groups in a configured SSM range.
Step 4	**ip igmp ssm-map query dns** **Example:** `Router(config)# ip igmp ssm-map query dns`	(Optional) Enables DNS-based SSM mapping. • By default, the **ip igmp ssm-map** command enables DNS-based SSM mapping. Only the **no** form of this command is saved to the running configuration. **Note** Use this command to reenable DNS-based SSM mapping if DNS-based SSM mapping is disabled.
Step 5	**ip domain multicast** *domain-prefix* **Example:** `Router(config)# ip domain multicast ssm-map.cisco.com`	(Optional) Changes the domain prefix used by the Cisco IOS software for DNS-based SSM mapping. • By default, the Cisco IOS software uses the ip-addr.arpa domain prefix.
Step 6	**ip name-server** *server-address1* [*server-address2...server-address6*] **Example:** `Router(config)# ip name-server 10.48.81.21`	Specifies the address of one or more name servers to use for name and address resolution.
Step 7	Repeat Step 6 to configure additional DNS servers for redundancy, if required.	—

What to Do Next

Proceed to the "Configuring Static Traffic Forwarding with SSM Mapping" section on page 12 or to the "Verifying SSM Mapping Configuration and Operation" section on page 13.

Configuring Static Traffic Forwarding with SSM Mapping

Perform this task to configure static traffic forwarding with SSM mapping on the last hop router. Static traffic forwarding can be used in conjunction with SSM mapping to statically forward SSM traffic for certain groups. When static traffic forwarding with SSM mapping is configured, the last hop router uses DNS-based SSM mapping to determine the sources associated with a group. The resulting (S, G) channels are then statically forwarded.

Prerequisites

This task does not include the steps for configuring DNS-based SSM mapping. See the "Configuring DNS-Based SSM Mapping" task for more information about configuring DNS-based SSM mapping.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip igmp static-group** *group* **source ssm-map**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 1/0`	Selects an interface on which to statically forward traffic for a multicast group using SSM mapping and enters interface configuration mode. **Note** Static forwarding of traffic with SSM mapping works with either DNS-based SSM mapping or statically-configured SSM mapping.
Step 4	**ip igmp static-group** *group-address* **source ssm-map** **Example:** `Router(config-if)# ip igmp static-group 232.1.2.1 source ssm-map`	Configures SSM mapping to be used to statically forward a (S, G) channel out of the interface. • Use this command if you want to statically forward SSM traffic for certain groups, but you want to use DNS-based SSM mapping to determine the source addresses of the channels.

What to Do Next

Proceed to the "Verifying SSM Mapping Configuration and Operation" section on page 13.

Verifying SSM Mapping Configuration and Operation

Perform this optional task to verify SSM mapping configuration and operation.

SUMMARY STEPS

1. **enable**
2. **show ip igmp ssm-mapping**
3. **show ip igmp ssm-mapping** *group-address*
4. **show ip igmp groups** [*group-name* | *group-address* | *interface-type interface-number*] [**detail**]
5. **show host**
6. **debug ip igmp** *group-address*

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

```
Router> enable
```

Step 2 **show ip igmp ssm-mapping**

(Optional) Displays information about SSM mapping.

The following example shows how to display information about SSM mapping configuration. In this example, SSM static mapping and DNS-based SSM mapping are enabled.

```
Router# show ip igmp ssm-mapping

SSM Mapping  : Enabled
DNS Lookup   : Enabled
Mcast domain : ssm-map.cisco.com
Name servers : 10.0.0.3
               10.0.0.4
```

Step 3 **show ip igmp ssm-mapping** *group-address*

(Optional) Displays the sources that SSM mapping uses for a particular group.

The following example shows how to display information about the configured DNS-based SSM mapping. In this example, the router has used DNS-based mapping to map group 232.1.1.4 to sources 172.16.8.5 and 172.16.8.6. The timeout for this entry is 860000 milliseconds (860 seconds).

```
Router# show ip igmp ssm-mapping 232.1.1.4

Group address: 232.1.1.4
Database     : DNS
DNS name     : 4.1.1.232.ssm-map.cisco.com
Expire time  : 860000
Source list  : 172.16.8.5
               : 172.16.8.6
```

Step 4 **show ip igmp groups** [*group-name* | *group-address* | *interface-type interface-number*] [**detail**]

(Optional) Displays the multicast groups with receivers that are directly connected to the router and that were learned through IGMP.

The following is sample output from the **show ip igmp groups** command with the *group-address* argument and **detail** keyword. In this example the "M" flag indicates that SSM mapping is configured.

```
Router# show ip igmp group 232.1.1.4 detail

            Interface:      Ethernet2
            Group:          232.1.1.4 SSM
            Uptime:         00:03:20
            Group mode:     INCLUDE
            Last reporter:  0.0.0.0
            CSR Grp Exp:    00:02:59
            Group source list: (C - Cisco Src Report, U - URD, R - Remote,
                                S - Static, M - SSM Mapping)
              Source Address   Uptime    v3 Exp   CSR Exp   Fwd  Flags
              172.16.8.3       00:03:20  stopped   00:02:59  Yes  CM
              172.16.8.4       00:03:20  stopped   00:02:59  Yes  CM
              172.16.8.5       00:03:20  stopped   00:02:59  Yes  CM
              172.16.8.6       00:03:20  stopped   00:02:59  Yes  CM
```

Step 5 **show host**

(Optional) Displays the default domain name, the style of name lookup service, a list of name server hosts, and the cached list of hostnames and addresses.

The following is sample output from the **show host** command. Use this command to display DNS entries as they are learned by the router.

```
Router# show host

Default domain is cisco.com
Name/address lookup uses domain service
Name servers are 10.48.81.21

Codes: UN - unknown, EX - expired, OK - OK, ?? - revalidate
       temp - temporary, perm - permanent
       NA - Not Applicable None - Not defined

Host                      Port   Flags      Age   Type    Address(es)
10.0.0.0.ssm-map.cisco.c  None  (temp, OK)   0     IP     172.16.8.5
                                                          172.16.8.6
                                                          172.16.8.3
```

172.16.8.4

Step 6 **debug ip igmp** *group-address*

(Optional) Displays the IGMP packets received and sent and IGMP host-related events.

The following is sample output from the **debug ip igmp** command when SSM static mapping is enabled. The following output indicates that the router is converting an IGMPv2 join for group G into an IGMPv3 join:

```
IGMP(0): Convert IGMPv2 report (*,232.1.2.3) to IGMPv3 with 2 source(s) using STATIC.
```

The following is sample output from the **debug ip igmp** command when DNS-based SSM mapping is enabled. The following output indicates that a DNS lookup has succeeded:

```
IGMP(0): Convert IGMPv2 report (*,232.1.2.3) to IGMPv3 with 2 source(s) using DNS.
```

The following is sample output from the **debug ip igmp** command when DNS-based SSM mapping is enabled and a DNS lookup has failed:

IGMP(0): DNS source lookup failed for (*, 232.1.2.3), IGMPv2 report failed

Configuration Examples for SSM Mapping

This section provides the following configuration examples:

SSM Mapping: Example

The following configuration example shows a router configuration for SSM mapping. This example also displays a range of other IGMP and SSM configuration options to show compatibility between features. Do not use this configuration example as a model unless you understand all of the features used in the example.

Note Address assignment in the global SSM range 232.0.0.0/8 should be random. If you copy parts or all of this sample configuration, make sure to select a random address range but not 232.1.1.x as shown in this example. Using a random address range minimizes the possibility of address collision and may prevent conflicts when other SSM content is imported while SSM mapping is used.

```
!
no ip domain lookup
ip domain multicast ssm.map.cisco.com
ip name-server 10.48.81.21
!
!
ip multicast-routing
ip igmp ssm-map enable
ip igmp ssm-map static 10 172.16.8.10
ip igmp ssm-map static 11 172.16.8.11
!
!
.
.
.
!
interface Ethernet0/0
 description Sample IGMP Interface Configuration for SSM-Mapping Example
 ip address 10.20.1.2 255.0.0.0
 ip pim sparse-mode
 ip igmp last-member-query-interval 100
 ip igmp static-group 232.1.2.1 source ssm-map
 ip igmp version 3
 ip igmp explicit-tracking
 ip igmp limit 2
 ip igmp v3lite
 ip urd
!
.
.
.
!
ip pim ssm default
!
```

```
access-list 10 permit 232.1.2.10
access-list 11 permit 232.1.2.0 0.0.0.255
!
```

Table 1 describes the significant commands shown in the SSM mapping configuration example.

Table 1 SSM Mapping Configuration Example Command Descriptions

Command	Description
no ip domain lookup	Disables IP DNS-based hostname-to-address translation. **Note** The **no ip domain-list** command is shown in the configuration only to demonstrate that disabling IP DNS-based hostname-to-address translation does not conflict with configuring SSM mapping. If this command is enabled, the Cisco IOS software will try to resolve unknown strings as hostnames.
ip domain multicast ssm-map.cisco.com	Specifies ssm-map.cisco.com as the domain prefix for SSM mapping.
ip name-server 10.48.81.21	Specifies 10.48.81.21 as the IP address of the DNS server to be used by SSM mapping and any other service in the Cisco IOS software that utilizes DNS.
ip multicast-routing	Enables IP multicast routing.
ip igmp ssm-map enable	Enables SSM mapping.
ip igmp ssm-map static 10 172.16.8.10	Configures the groups permitted by ACL 10 to use source address 172.16.8.10. • In this example, ACL 10 permits all groups in the 232.1.2.0/25 range except 232.1.2.10.
ip igmp ssm-map static 11 172.16.8.11	Configures the groups permitted by ACL 11 to use source address 172.16.8.11. • In this example, ACL 11 permits group 232.1.2.10.
ip pim sparse-mode	Enables PIM sparse mode.
ip igmp last-member-query-interval 100	Reduces the leave latency for IGMPv2 hosts. **Note** This command is not required for configuring SSM mapping; however, configuring this command can be beneficial for IGMPv2 hosts relying on SSM mapping.
ip igmp static-group 232.1.2.1 source ssm-map	Configures SSM mapping to be used to determine the sources associated with group 232.1.2.1. The resulting (S, G) channels are statically forwarded.
ip igmp version 3	Enables IGMPv3 on this interface. **Note** This command is shown in the configuration only to demonstrate that IGMPv3 can be configured simultaneously with SSM mapping; however, it is not required.

Table 1 SSM Mapping Configuration Example Command Descriptions (continued)

Command	Description
ip igmp explicit-tracking	Minimizes the leave latency for IGMPv3 host leaving a multicast channel. **Note** This command is not required for configuring SSM mapping.
ip igmp limit 2	Limits the number of IGMP states resulting from IGMP membership states on a per-interface basis. **Note** This command is not required for configuring SSM mapping.
ip igmp v3lite	Enables the acceptance and processing of IGMP v3lite membership reports on this interface. **Note** This command is shown in the configuration only to demonstrate that IGMP v3lite can be configured simultaneously with SSM mapping; however, it is not required.
ip urd	Enables interception of TCP packets sent to the reserved URD port 465 on an interface and processing of URD channel subscription reports. **Note** This command is shown in the configuration only to demonstrate that URD can be configured simultaneously with SSM mapping; however, it is not required.
ip pim ssm default	Configures SSM service. • The **default** keyword defines the SSM range access list as 232/8.
access-list 10 permit 232.1.2.10 **access-list 11 permit 232.1.2.0 0.0.0.255**	Configures the ACLs to be used for static SSM mapping. **Note** These are the ACLs that are referenced by the **ip igmp ssm-map static** commands in this configuration example.

DNS Server Configuration: Example

To configure DNS-based SSM mapping, you need to create a DNS server zone or add records to an existing zone. If the routers that are using DNS-based SSM mapping are also using DNS for other purposes besides SSM mapping, you should use a normally-configured DNS server. If DNS-based SSM mapping is the only DNS implementation being used on the router, you can configure a fake DNS setup with an empty root zone, or a root zone that points back to itself.

The following example shows how to create a zone and import the zone data using Network Registrar:

```
nrcmd> zone 1.1.232.ssm-map.cisco.com. create primary file=named.ssm-map
100 Ok
nrcmd> dns reload
100 Ok
```

The following example shows how to import the zone files from a named.conf file for BIND 8:

```
nrcmd> ::import named.conf /etc/named.conf
```

```
nrcmd> dns reload
100 Ok:
```

Note Network Registrar version 8.0 and later support import BIND 8 format definitions.

Additional References

The following sections provide additional references related to the SSM Mapping feature.

Related Documents

Related Topic	Document Title
SSM concepts and configuration	"Configuring Basic IP Multicast" module
Cisco IOS IP multicast commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	*Cisco IOS IP Multicast Command Reference*
IGMP v3lite and URD concepts and configuration	"Configuring Source Specific Multicast" chapter in the "IP Multicast" part of the *Cisco IOS IP Configuration Guide*, Release 12.2

Standards

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

MIBs

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

RFCs

RFCs	Title
RFC 2365	*Administratively Scoped IP Multicast*
RFC 2770	*GLOP Addressing in 233/8*
RFC 3569	*An Overview of Source-Specific Multicast*

Technical Assistance

Description	Link
The Cisco Technical Support & Documentation 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

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IP Multicast Command Reference* at http://www.cisco.com/en/US/docs/ios/ipmulti/command/reference/imc_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

- **debug ip igmp**
- **ip domain multicast**
- **ip igmp ssm-map enable**
- **ip igmp ssm-map query dns**
- **ip igmp ssm-map static**
- **ip igmp static-group**
- **show ip igmp groups**
- **show ip igmp ssm-mapping**

Glossary

DNS—Domain Name System. System used on the Internet for translating names of network nodes into addresses.

IGMP—Internet Group Management Protocol. Protocol used by IP hosts to report their multicast group memberships to an adjacent multicast router.

IGMPv3—IGMP is the protocol used by IPv4 systems to report their IP multicast group memberships to neighboring multicast routers. Version 3 of IGMP adds support for source filtering, that is, the ability for a system to report interest in receiving packets only from specific source addresses, or from all but specific source addresses, sent to a particular multicast address.

PIM—Protocol Independent Multicast. Multicast routing architecture that allows the addition of IP multicast routing on existing IP networks. PIM is unicast routing protocol independent and can be operated in two modes: dense and sparse.

SSM—Source Specific Multicast. A datagram delivery model that best supports one-to-many applications, also known as broadcast applications. SSM is the core networking technology for the Cisco implementation of the IP Multicast Lite suite of solutions targeted for audio and video broadcast application environments

URD—URL Rendezvous Directory. Multicast solution that directly provides the network with information about the specific source of a content stream. It enables the network to quickly establish the most direct distribution path from the source to the receiver, thus substantially reducing the time and effort required in receiving the streaming media. URD allows an application to identify the source of the content stream through a web page link or web directly. When that information is sent back to the application it is then conveyed back to the network using URD.

VRF—A VPN routing and forwarding instance. 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 PE router.

Note Refer to the *Internetworking Terms and Acronyms* for terms not included in this glossary.

SSM Channel Based Filtering for Multicast Boundaries

The SSM Channel Based Filtering for Multicast Boundaries feature enables the user to apply filtering policies based on Source Specific Multicast (SSM) channels for Source and Group (S,G) addresses, which is a combination of source and destination IP addresses.

Feature History for the SSM Channel Based Filtering for Multicast Boundaries Feature

Release	Modification
12.3(11)T	This feature was introduced.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for SSM Channel Based Filtering for Multicast Boundaries

- IP multicast needs to be configured on the router.

Restrictions for SSM Channel Based Filtering for Multicast Boundaries

- The **filter-autorp** keyword does not support extended access lists.

Information About the SSM Channel Based Filtering for Multicast Boundaries Feature

To configure the SSM Channel Based Filtering for Multicast Boundaries feature, you should understand the following concepts:

- Rules for Multicast Boundaries, page 2
- Benefits of SSM Channel Based Filtering for Multicast Boundaries, page 3

Rules for Multicast Boundaries

The SSM Channel Based Filtering for Multicast Boundaries feature expands the **ip multicast boundary** command for control plane filtering support. More than one **ip multicast boundary** command can be applied to an interface.

The following rules govern the **ip multicast boundary** command:

- One instance of the **in** and **out** keywords can be configured on an interface.
- The **in** and **out** keywords can be used for standard or extended access lists.
- Only standard access lists are permitted with the use of the **filter-autorp** keyword or no keyword.
- A maximum of three instances of a command will be allowed on an interface: one instance of **in**, one instance of **out**, and one instance of **filter-autorp** or no keyword.
- When multiple instances of the command are used, the filtering will be cumulative. If a boundary statement with no keyword exists with a boundary statement with the **in** keyword, both access lists will be applied on the in direction and a match on either one will be sufficient.
- All instances of the command apply to both control and data plane traffic.
- Protocol information on the extended access list is parsed to allow reuse and filtering for IOS consistency. An (S,G) operation will be filtered by an extended access list under all conditions stated above for keywords if the access list filters (S,G) traffic for all protocols.

Benefits of SSM Channel Based Filtering for Multicast Boundaries

- This feature allows input on the source interface.
- The access control capabilities are the same for SSM and Any Source Multicast (ASM).

How to Configure SSM Channel Based Filtering for Multicast Boundaries

This section contains the following procedures:

- Configuring the Multicast Boundaries, page 3

Configuring the Multicast Boundaries

Perform this task to configure the multicast boundary.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list {standard | extended}** *access-list-name*
4. **permit** *protocol* **host** *address* **host** *address*
5. **deny** *protocol* **host** *address* **host** *address*
6. Repeat Step 4 or Step 5 as needed.
7. **interface** *type interface-number port-number*
8. **ip multicast boundary** *access-list-name* [**in** |**out** | **filter-autorp**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list {standard | extended}** *access-list-name* **Example:** `Router(config)# ip access-list 101`	Configures the standard or extended access list.
Step 4	**permit** *protocol* **host** *address* **host** *address* **Example:** `Router(config-ext-nacl)# permit ip host 181.1.2.201 host 232.1.1.11`	Permits specified ip host traffic.
Step 5	**deny** *protocol* **host** *address* **host** *address* **Example:** `Router(config-acl-nacl)# deny ip host 181.1.2.203 host 232.1.1.1`	Denies specified multicast ip group and source traffic.
Step 6	`Repeat Step 4 or Step 5 as needed.`	Permits and denies specified host and source traffic.
Step 7	**interface** *type* **interface-number** *port-number* **Example:** `Router(config)# interface ethernet 2/3`	Enables interface configuration mode.
Step 8	**ip multicast boundary** *access-list-name* **[in | out | filter-autorp]** **Example:** `Router(config-if)# ip multicast boundary acc_grp1 out`	Configures the multicast boundary.

Configuration Examples for SSM Channel Based Filtering for Multicast Boundaries

This section provides the following configuration examples for the multicast boundaries:

Configuring the Multicast Boundaries: Example

The following example permits outgoing traffic for (181.1.2.201, 232.1.1.1) and (181.1.2.202, 232.1.1.1) and denies all other (S,G)s.

```
configure terminal
 ip access-list extended acc_grp1
 permit ip host 0.0.0.0 232.1.1.1 0.0.0.255
 permit ip host 181.1.2.201 host 232.1.1.1
 permit udp host 181.1.2.202 host 232.1.1.1
 permit ip host 181.1.2.202 host 232.1.1.1
 deny igmp host 181.2.3.303 host 232.1.1.1
 interface ethernet 2/3
 ip multicast boundary acc_grp1 out
```

The following example permits outgoing traffic for (181.1.2.201, 232.1.1.5) and 181.1.2.202, 232.1.1.5).

```
configure terminal
 ip access-list extended acc_grp6
 permit ip host 0.0.0.0 232.1.1.1 5.0.0.255
 deny udp host 181.1.2.201 host 232.1.1.5
 permit ip host 181.1.2.201 host 232.1.1.5
 deny pim host 181.1.2.201 host 232.1.1.5
 permit ip host 181.1.2.202 host 232.1.1.5
 deny igmp host 181.2.3.303 host 232.1.1.1
 interface ethernet 2/3
 ip multicast boundary acc_grp6 out
```

The following example denies a group-range that is being announced by the candidate RP. Since the group range is denied, there will be no pim auto-rp mappings created.

```
configure terminal
 ip access-list standard acc_grp10
 deny 225.0.0.0 0.255.255.255
 permit any
 access-list extended acc_grp12
 permit pim host 181.1.2.201 host 232.1.1.8
 deny udp host 181.1.2.201 host 232.1.1.8
 permit pim host 181.1.2.203 0.0.0.255 host 227.7.7.7
 permit ip host 0.0.0.0 host 227.7.7.7
 permit ip 181.1.2.203 0.0.0.255 host 227.7.7.7
 permit ip host 181.1.2.201 host 232.1.1.7
 ip access-list extended acc_grp13
 deny ip host 181.1.2.201 host 232.1.1.8
 permit ip any any
 interface ethernet 2/3
 ip multicast boundary acc_grp10 filter-autorp
 ip multicast boundary acc_grp12 out
 ip multicast boundary acc_grp13 in
```

Additional References

The following sections provide references related to the Multicast VPN MIB feature.

Related Documents

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

Standards

Standards	Title
No new or modified standards are supported by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 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/public/support/tac/home.shtml

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IP Multicast Command Reference* at http://www.cisco.com/en/US/docs/ios/ipmulti/command/reference/imc_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

ip multicast boundary

Verifying IP Multicast Operation

First Published: May 2, 2005
Last Updated: June 2, 2008

This module describes how to verify IP multicast operation in a network after Protocol Independent Multicast (PIM) sparse mode (PIM-SM) or Source Specific Multicast (PIM-SSM) has been implemented. The tasks in this module can be used to test IP multicast reachability and to confirm that receivers and sources are operating as expected in an IP multicast network.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Verifying IP Multicast Operation" section on page 21.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Verifying IP Multicast Operation

- Before performing the tasks in this module, you should be familiar with the concepts described in the "IP Multicast Technology Overview" module.
- The tasks in this module assume that IP multicast has been enabled and that PIM-SM or SSM has been configured using the relevant tasks described in the "Configuring Basic IP Multicast" module.

Restrictions for Verifying IP Multicast Operation

- For PIM-SM, this module assumes that the shortest path tree (SPT) threshold for PIM-enabled routers is set to the value of zero (the default) and not infinity. For more information about setting the SPT threshold, see the **ip pim spt-threshold** command page in the *Cisco IOS IP Multicast Command Reference*.
- Verifying IP multicast operation in a bidirectional PIM (bidir-PIM) network or a PIM-SM network with a finite or infinite SPT threshold is outside the scope of this module.

Information About Verifying IP Multicast Operation

Before you verify IP multicast operation, you should understand the following concept:

- Guidelines for Verifying IP Multicast Operation in a PIM-SM and PIM-SSM Network Environment, page 2

Guidelines for Verifying IP Multicast Operation in a PIM-SM and PIM-SSM Network Environment

When you verify the operation of IP multicast in a PIM-SM network environment or in an PIM-SSM network environment, a useful approach is to begin the verification process on the last hop router, and then continue the verification process on the routers along the SPT until the first hop router has been reached. The goal of the verification is to ensure that IP multicast traffic is being routed properly through an IP multicast network.

Common Commands Used to Verify IP Multicast Operation on the Last Hop Router (PIM-SM and PIM-SSM)

Table 1 describes the common commands used to verify IP multicast operation on the last hop router in PIM-SM and PIM-SSM network environments.

Table 1 Common IP Multicast Verification Commands (Last Hop Router)

Command	Description and Purpose
show ip igmp groups	Displays the multicast groups with receivers that are directly connected to the router and that were learned through the Internet Group Management Protocol (IGMP). • Use this command to confirm that the IGMP cache is being properly populated on the last hop router for the groups that receivers on the LAN have joined.
show ip pim rp mapping	Displays all group-to-RP mappings of which the router is aware (either configured or learned from Auto-RP or BSR). • Use this command to confirm that the group-to-RP mappings are being populated correctly on the last hop router. **Note** The **show ip pim rp mapping** command does not work with routers in a PIM-SSM network because PIM-SSM does not use rendezvous points (RPs).
show ip mroute	Displays the contents of the multicast routing (mroute) table. • Use this command to verify that the mroute table is being populated properly on the last hop router.
show ip interfaces	Displays information and statistics about configured interfaces. • Use this command to verify that IP multicast fast switching is enabled on the outgoing interface on the last hop router.
show ip mcache	Displays the contents of the IP multicast fast-switching cache. • Use this command to confirm that the IP multicast fast-switching cache is being populated properly on the last hop router. **Note** This command is used only if multicast fast switching is enabled (which is the default behavior).
show ip pim interface count	Displays statistics related to the number of multicast packets received by and sent out a PIM-enabled interface. • Use this command on the last hop router to confirm that multicast traffic is being forwarded on the last hop router.

Table 1 Common IP Multicast Verification Commands (Last Hop Router) (continued)

Command	Description and Purpose
show ip mroute active	Displays the rate that active sources are sending to multicast groups, in kilobits per second (kb/s). • Use this command to display information about the multicast packet rate for active sources sending to groups on the last hop router.
show ip mroute count	Displays statistics related to mroutes in the mroute table. • Use this command on the last hop router to confirm that multicast traffic is flowing on the last hop router.

Common Commands Used to Verify IP Multicast Operation on Routers Along the SPT (PIM-SM and PIM-SSM)

Table 2 describes the common commands used to verify IP multicast operation on routers along the SPT in PIM-SM and PIM-SSM network environments.

Table 2 Common IP Multicast Verification Commands (Routers Along SPT)

Command	Description and Purpose
show ip mroute	Displays the contents of the mroute table. • Use this command to confirm that the Reverse Path Forwarding (RPF) neighbor toward the source is the expected RPF neighbor for each router along the SPT.
show ip mroute active	Displays the rate that active sources are sending to multicast groups, in kb/s. • Use this command to display information about the multicast packet rate for active sources sending to groups on routers along the SPT.

Common Commands Used to Verify IP Multicast Operation on the First Hop Router (PIM-SM and PIM-SSM)

Table 3 describes the common commands used to verify IP multicast operation on the first hop router in PIM-SM and PIM-SSM network environments.

Table 3 Common IP Multicast Verification Commands (First Hop Router)

Command	Description and Purpose
show ip mroute	Displays the contents of the mroute table. • Use this command to confirm that the F flag is set for the mroutes on the first hop router.
show ip mroute active	Displays the rate that active sources are sending to multicast groups, in kb/s. • Use this command to display information about the multicast packet rate for active sources sending to groups on the first hop router.

How to Verify IP Multicast Operation

This section contains the following tasks:

- Using PIM-Enabled Routers to Test IP Multicast Reachability, page 5 (optional)
- Verifying IP Multicast Operation in a PIM-SM or a PIM-SSM Network, page 7 (optional)

Using PIM-Enabled Routers to Test IP Multicast Reachability

Perform the following tasks to use PIM-enabled routers to test IP multicast reachability.

If all the PIM-enabled routers and access servers that you administer are members of a multicast group, pinging that group causes all routers to respond, which can be a useful administrative and debugging tool.

To use PIM-enabled routers to test IP multicast reachability, perform the following tasks:

- Configuring Routers to Respond to Multicast Pings, page 5 (optional)
- Pinging Routers Configured to Respond to Multicast Pings, page 7 (optional)

Configuring Routers to Respond to Multicast Pings

Perform the following task to configure routers to respond to multicast pings. Performing this task configures interfaces on the router to join a specified group. This task should be performed on each interface on the router participating in the multicast network and on all routers participating in the multicast network.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip igmp join-group** *group-address*
5. Repeat Step 3 and Step 4 for each interface on the router participating in the multicast network.
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Enters interface configuration mode. • For the *type* and *number* arguments, specify an interface that is directly connected to hosts or is facing hosts.
Step 4	**ip igmp join-group** *group-address* **Example:** `Router(config-if)# ip igmp join-group 225.2.2.2`	(Optional) Configures an interface on the router to join the specified group. • For the purpose of this task, configure the same group address for the *group-address* argument on all interfaces on the router participating in the multicast network. **Note** With this method, the router accepts the multicast packets in addition to forwarding them. Accepting the multicast packets prevents the router from fast switching.
Step 5	Repeat Step 3 and Step 4 for each interface on the router participating in the multicast network.	—
Step 6	**end** **Example:** `Router(config-if)# end`	Ends the current configuration session and returns to privileged EXEC mode.

Pinging Routers Configured to Respond to Multicast Pings

Perform the following task on a router to initiate a ping test to the routers configured to respond to multicast pings. This task is used to test IP multicast reachability in a network.

SUMMARY STEPS

1. **enable**
2. **ping** *group-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**ping** *group-address* **Example:** `Router# ping 225.2.2.2`	Pings an IP multicast group address. • A successful response indicates that the group address is functioning.

Verifying IP Multicast Operation in a PIM-SM or a PIM-SSM Network

Perform the following optional tasks to verify IP multicast operation in a PIM-SM or a PIM-SSM network. You can perform the steps in these tasks to locate a faulty hop when sources and receivers are not operating as expected.

Note If packets are not reaching their expected destinations, you might want consider disabling IP multicast fast switching, which would place the router in process switching mode. If packets begin reaching their proper destinations after IP multicast fast switching has been disabled, then the issue most likely was related to IP multicast fast switching. See the "Monitoring and Maintaining IP Multicast" module for information on how to disable IP multicast fast switching.

To verify IP multicast operation in a PIM-SM or PIM-SSM multicast network, perform the following verification tasks:

- Verifying IP Multicast Operation on the Last Hop Router, page 7 (optional)
- Verifying IP Multicast on Routers Along the SPT, page 12 (optional)
- Verifying IP Multicast on the First Hop Router, page 13 (optional)

Verifying IP Multicast Operation on the Last Hop Router

Perform the following task to verify the operation of IP multicast on the last hop router.

Note If you are verifying a last hop router in a PIM-SSM network, ignore Step 3.

SUMMARY STEPS

1. **enable**
2. **show ip igmp groups**
3. **show ip pim rp mapping**
4. **show ip mroute**
5. **show ip interfaces** [*type number*]
6. **show ip mcache**
7. **show ip pim interface count**
8. **show ip mroute count**
9. **show ip mroute active** [*kb/s*]

DETAILED STEPS

Step 1 enable

Enables privileged EXEC mode. Enter your password if prompted.

```
Router> enable
```

Step 2 show ip igmp groups

Use this command to verify IGMP memberships on the last hop router. This information will confirm the multicast groups with receivers that are directly connected to the last hop router and that are learned through IGMP.

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

```
Router# show ip igmp groups

IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter
239.1.2.3        Ethernet1/0              00:05:14  00:02:14  10.1.0.6
224.0.1.39       Ethernet0/0              00:09:11  00:02:08  172.31.100.1
```

Step 3 **show ip pim rp mapping**

Use this command to confirm that the group-to-RP mappings are being populated correctly on the last hop router.

Note Ignore this step if you are verifying a last hop router in a PIM-SSM network. The **show ip pim rp mapping** command does work with routers in a PIM-SSM network because PIM-SSM does not use RPs. In addition, if configured correctly, PIM-SSM groups should not appear in the output of the **show ip pim rp mapping** command.

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

```
Router# show ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 224.0.0.0/4
  RP 172.16.0.1 (?), v2v1
    Info source: 172.16.0.1 (?), elected via Auto-RP
         Uptime: 00:09:11, expires: 00:02:47
```

Step 4 **show ip mroute**

Use this command to verify that the mroute table is being populated properly on the last hop router.

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

```
Router# show ip mroute

(*, 239.1.2.3), 00:05:14/00:03:04, RP 172.16.0.1, flags: SJC
  Incoming interface: Ethernet0/0, RPF nbr 172.31.100.1
  Outgoing interface list:
    Ethernet1/0, Forward/Sparse-Dense, 00:05:10/00:03:04

(10.0.0.1, 239.1.2.3), 00:02:49/00:03:29, flags: T
  Incoming interface: Ethernet0/0, RPF nbr 172.31.100.1
  Outgoing interface list:
    Ethernet1/0, Forward/Sparse-Dense, 00:02:49/00:03:04

(*, 224.0.1.39), 00:10:05/stopped, RP 0.0.0.0, flags: DC
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet1/0, Forward/Sparse-Dense, 00:05:15/00:00:00
    Ethernet0/0, Forward/Sparse-Dense, 00:10:05/00:00:00

(172.16.0.1, 224.0.1.39), 00:02:00/00:01:33, flags: PTX
  Incoming interface: Ethernet0/0, RPF nbr 172.31.100.1
```

Step 5 **show ip interfaces** [*type number*]

Use this command to verify that multicast fast switching is enabled for optimal performance on the outgoing interface on the last hop router.

Note Using the **no ip mroute-cache** interface command disables multicast fast-switching. When IP multicast fast switching is disabled, packets are forwarded through the process-switched path.

The following is sample output from the **show ip interfaces** command for a particular interface:

```
Router# show ip interfaces Ethernet 0/0

Ethernet0/0 is up, line protocol is up
  Internet address is 172.31.100.2/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined: 224.0.0.1 224.0.0.22 224.0.0.13
      224.0.0.5 224.0.0.6
  Outgoing access list is not set
  Inbound  access list is not set
  Proxy ARP is enabled
  Local Proxy ARP is disabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  IP fast switching is enabled
  IP fast switching on the same interface is disabled
  IP Flow switching is disabled
  IP CEF switching is disabled
  IP Fast switching turbo vector
  IP multicast fast switching is enabled
  IP multicast distributed fast switching is disabled
  IP route-cache flags are Fast
  Router Discovery is disabled
  IP output packet accounting is disabled
  IP access violation accounting is disabled
  TCP/IP header compression is disabled
  RTP/IP header compression is disabled
  Policy routing is disabled
  Network address translation is disabled
  WCCP Redirect outbound is disabled
  WCCP Redirect inbound is disabled
  WCCP Redirect exclude is disabled
  BGP Policy Mapping is disabled
```

Step 6 **show ip mcache**

Use this command to confirm that the IP multicast fast-switching cache is being populated properly on the last hop router.

Note Ignore this step if IP multicast has been disabled.

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

```
Router# show ip mcache

IP Multicast Fast-Switching Cache
(10.0.0.1/32, 239.1.2.3), Ethernet0/0, Last used: 00:00:00, MinMTU: 1500
  Ethernet1/0              MAC Header: 01005E010203AABBCC002B010800
(172.16.0.1/32, 224.0.1.39), Ethernet0/0, Last used: 00:01:40, MinMTU: 1500L
```

Step 7 **show ip pim interface count**

Use this command to confirm that multicast traffic is being forwarded on the last hop router.

The following is sample output from the **show ip pim interface** command with the **count** keyword:

```
Router# show ip pim interface count

State: * - Fast Switched, D - Distributed Fast Switched
       H - Hardware Switching Enabled
Address          Interface                FS Mpackets In/Out
172.31.100.2     Ethernet0/0              *   4122/0
10.1.0.1         Ethernet1/0              *   0/3193
```

Step 8 **show ip mroute count**

Use this command to confirm that multicast traffic is being forwarded on the last hop router.

The following is sample output from the **show ip mroute** command with the **count** keyword:

```
Router# show ip mroute count

IP Multicast Statistics
6 routes using 4008 bytes of memory
3 groups, 1.00 average sources per group
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per second
Other counts: Total/RPF failed/Other drops(OIF-null, rate-limit etc)

Group: 239.1.2.3, Source count: 1, Packets forwarded: 3165, Packets received: 3165
  RP-tree: Forwarding: 0/0/0/0, Other: 0/0/0
  Source: 10.0.0.1/32, Forwarding: 3165/20/28/4, Other: 0/0/0

Group: 224.0.1.39, Source count: 1, Packets forwarded: 21, Packets received: 120
  Source: 172.16.0.1/32, Forwarding: 21/1/48/0, Other: 120/0/99

Group: 224.0.1.40, Source count: 1, Packets forwarded: 10, Packets received: 10
  Source: 172.16.0.1/32, Forwarding: 10/1/48/0, Other: 10/0/0
```

Step 9 **show ip mroute active** [*kb/s*]

Use this command on the last hop router to display information about active multicast sources sending traffic to groups on the last hop router. The output of this command provides information about the multicast packet rate for active sources.

Note By default, the output of the **show ip mroute** command with the **active** keyword displays information about active sources sending traffic to groups at a rate greater than or equal to 4 kb/s. To display information about active sources sending low-rate traffic to groups (that is, traffic less than 4 kb/s), specify a value of a 1 for the *kb/s* argument. Specifying a value of 1 for this argument displays information about active sources sending traffic to groups at a rate equal to or greater than 1 kb/s, which effectively displays information about all possible active source traffic.

The following is sample output from the **show ip mroute** command with the **active** keyword:

```
Router# show ip mroute active

Active IP Multicast Sources - sending >= 4 kbps

Group: 239.1.2.3, (?)
   Source: 10.0.0.1 (?)
     Rate: 20 pps/4 kbps(1sec), 4 kbps(last 50 secs), 4 kbps(life avg)
```

Verifying IP Multicast on Routers Along the SPT

Perform the following task to verify the operation of IP multicast on routers along the SPT in a PIM-SM or PIM-SSM network.

SUMMARY STEPS

1. **enable**
2. **show ip mroute** [*group-address*]
3. **show ip mroute active** [*kb/s*]

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

```
Router> enable
```

Step 2 **show ip mroute** [*group-address*]

Use this command on routers along the SPT to confirm the RPF neighbor toward the source for a particular group or groups.

The following is sample output from the **show ip mroute** command for a particular group:

```
Router# show ip mroute 239.1.2.3

(*, 239.1.2.3), 00:17:56/00:03:02, RP 172.16.0.1, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0/0, Forward/Sparse-Dense, 00:17:56/00:03:02

(10.0.0.1, 239.1.2.3), 00:15:34/00:03:28, flags: T
  Incoming interface: Serial1/0, RPF nbr 172.31.200.1
  Outgoing interface list:
    Ethernet0/0, Forward/Sparse-Dense, 00:15:34/00:03:02
```

Step 3 **show ip mroute active**

Use this command on routers along the SPT to display information about active multicast sources sending to groups. The output of this command provides information about the multicast packet rate for active sources.

Note By default, the output of the **show ip mroute** command with the **active** keyword displays information about active sources sending traffic to groups at a rate greater than or equal to 4 kb/s. To display information about active sources sending low-rate traffic to groups (that is, traffic less than 4 kb/s), specify a value of a 1 for the *kb/s* argument. Specifying a value of 1 for this argument displays information about active sources sending traffic to groups at a rate equal to or greater than 1 kb/s, which effectively displays information about all possible active source traffic.

The following is sample output from the **show ip mroute** command with the **active** keyword:

```
Router# show ip mroute active

Active IP Multicast Sources - sending >= 4 kbps

Group: 239.1.2.3, (?)
```

```
   Source: 10.0.0.1 (?)
     Rate: 20 pps/4 kbps(1sec), 4 kbps(last 30 secs), 4 kbps(life avg)
```

Verifying IP Multicast on the First Hop Router

Perform the following task to verify the operation of IP multicast on the first hop router.

SUMMARY STEPS

1. **enable**
2. **show ip mroute** [*group-address*]
3. **show ip mroute active** [*kb/s*]

Step 1 **enable**

Enables privileged EXEC mode.

- Enter your password if prompted.

Example:

```
Router> enable
```

Step 2 **show ip mroute** [*group-address*]

Use this command on the first hop router to confirm the F flag has been set for mroutes on the first hop router.

The following is sample output from the **show ip mroute** for a particular group:

```
Router# show ip mroute 239.1.2.3

(*, 239.1.2.3), 00:18:10/stopped, RP 172.16.0.1, flags: SPF
  Incoming interface: Serial1/0, RPF nbr 172.31.200.2
  Outgoing interface list: Null

(10.0.0.1, 239.1.2.3), 00:18:10/00:03:22, flags: FT
  Incoming interface: Ethernet0/0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial1/0, Forward/Sparse-Dense, 00:18:10/00:03:19
```

Step 3 **show ip mroute active** [*kb/s*]

Use this command on the first hop router to display information about active multicast sources sending to groups. The output of this command provides information about the multicast packet rate for active sources.

Note By default, the output of the **show ip mroute** command with the **active** keyword displays information about active sources sending traffic to groups at a rate greater than or equal to 4 kb/s. To display information about active sources sending low-rate traffic to groups (that is, traffic less than 4 kb/s), specify a value of a 1 for the *kb/s* argument. Specifying a value of 1 for this argument displays information about active sources sending traffic to groups at a rate equal to or greater than 1 kb/s, which effectively displays information about all possible active source traffic.

The following is sample output from the **show ip mroute** command with the **active** keyword:

```
Router# show ip mroute active

Active IP Multicast Sources - sending >= 4 kbps

Group: 239.1.2.3, (?)
   Source: 10.0.0.1 (?)
     Rate: 20 pps/4 kbps(1sec), 4 kbps(last 30 secs), 4 kbps(life avg)
```

Configuration Examples for Verifying IP Multicast Operation

This section provides the following configuration example:

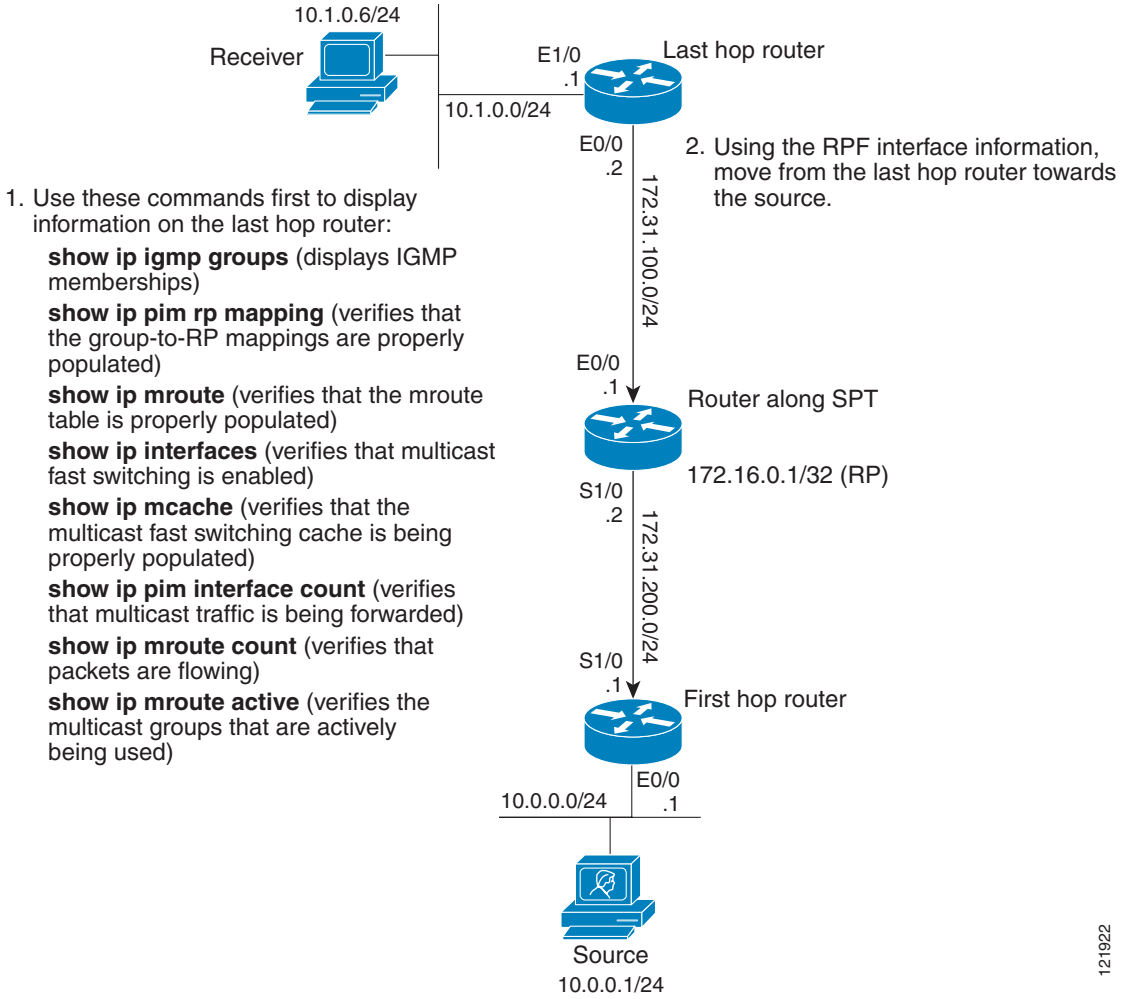
- Verifying IP Multicast Operation in a PIM-SM or PIM-SSM Network: Example, page 14

Verifying IP Multicast Operation in a PIM-SM or PIM-SSM Network: Example

The following example shows how to verify IP multicast operation after PIM-SM has been deployed in a network. The example is based on the PIM-SM topology illustrated in Figure 1.

From the last hop router to the first hop router shown in Figure 1, this example shows how to verify IP multicast operation for this particular PIM-SM network topology.

Figure 1 Locating a Faulty Hop in a Multicast Network



Verifying IP Multicast on the Last Hop Router: Example

The following is sample output from the **show ip igmp groups** command. The sample output displays the IGMP memberships on the last hop router shown in Figure 1. This command is used in this example to confirm that the IGMP cache is being properly populated for the groups that receivers on the LAN have joined.

```
Router# show ip igmp groups

IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter
239.1.2.3        Ethernet1/0              00:05:14  00:02:14  10.1.0.6
224.0.1.39       Ethernet0/0              00:09:11  00:02:08  172.31.100.1
```

The following is sample output from the **show ip pim rp mapping** command. In the sample output, notice the RP address displayed for the RP field. Use the RP address and group information to verify that the group-to-RP mappings have been properly populated on the last hop router shown in Figure 1.

Note In the output, the "(?)" indicates that the router is unable to resolve an IP address to a hostname.

```
Router# show ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 224.0.0.0/4
  RP 172.16.0.1 (?), v2v1
    Info source: 172.16.0.1 (?), elected via Auto-RP
         Uptime: 00:09:11, expires: 00:02:47
```

The following is sample output from the **show ip mroute** command. This command is used to verify that the mroute table is being properly populated on the last hop router shown in Figure 1. In the sample output, notice the T flag for the (10.0.0.1, 239.1.2.3) mroute. The T flag indicates that the SPT-bit has been set, which means a multicast packet was received on the SPT tree for this particular mroute. In addition, the RPF nbr field should point toward the RPF neighbor with the highest IP address determined by unicast routing toward the multicast source.

```
Router# show ip mroute

(*, 239.1.2.3), 00:05:14/00:03:04, RP 172.16.0.1, flags: SJC
  Incoming interface: Ethernet0/0, RPF nbr 172.31.100.1
  Outgoing interface list:
    Ethernet1/0, Forward/Sparse-Dense, 00:05:10/00:03:04

(10.0.0.1, 239.1.2.3), 00:02:49/00:03:29, flags: T
  Incoming interface: Ethernet0/0, RPF nbr 172.31.100.1
  Outgoing interface list:
    Ethernet1/0, Forward/Sparse-Dense, 00:02:49/00:03:04

(*, 224.0.1.39), 00:10:05/stopped, RP 0.0.0.0, flags: DC
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet1/0, Forward/Sparse-Dense, 00:05:15/00:00:00
    Ethernet0/0, Forward/Sparse-Dense, 00:10:05/00:00:00
```

The following is sample output from the **show ip interfaces** command for the incoming interface. This command is used in this example to confirm that IP multicast fast switching is enabled on the last hop router shown in Figure 1. When IP multicast fast switching is enabled, the line “IP multicast fast switching is enabled” displays in the output.

```
Router# show ip interfaces Ethernet 0/0

Ethernet0/0 is up, line protocol is up
  Internet address is 172.31.100.2/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined: 224.0.0.1 224.0.0.22 224.0.0.13
      224.0.0.5 224.0.0.6
  Outgoing access list is not set
  Inbound  access list is not set
  Proxy ARP is enabled
  Local Proxy ARP is disabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  IP fast switching is enabled
  IP fast switching on the same interface is disabled
  IP Flow switching is disabled
  IP CEF switching is disabled
```

```
  IP Fast switching turbo vector
  IP multicast fast switching is enabled
  IP multicast distributed fast switching is disabled
  IP route-cache flags are Fast
  Router Discovery is disabled
  IP output packet accounting is disabled
  IP access violation accounting is disabled
  TCP/IP header compression is disabled
  RTP/IP header compression is disabled
  Policy routing is disabled
  Network address translation is disabled
  WCCP Redirect outbound is disabled
  WCCP Redirect inbound is disabled
  WCCP Redirect exclude is disabled
  BGP Policy Mapping is disabled
```

The following is sample output from the **show ip mcache** command. This command is used in this example to confirm that the IP multicast fast-switching cache is being properly populated on the last hop router shown in Figure 1. When examining the output of this command, you should verify that (S, G) mroutes have been populated and the incoming and outgoing interfaces are listed correctly. The incoming interface appears on the same line as the (S, G). The outgoing interfaces will appear below the (S, G) output. For example, the incoming interface for the output (10.0.0.1/32, 239.1.2.3) in the following example is Ethernet0/0. The outgoing interface is Ethernet1/0.

Note The MAC headers displayed in the MAC Header field should always begin with 01005E.

```
Router# show ip mcache

IP Multicast Fast-Switching Cache
(10.0.0.1/32, 239.1.2.3), Ethernet0/0, Last used: 00:00:00, MinMTU: 1500
  Ethernet1/0              MAC Header: 01005E010203AABBCC002B010800
(172.16.0.1/32, 224.0.1.39), Ethernet0/0, Last used: 00:01:40, MinMTU: 1500
```

The following is sample output from the **show ip pim interface count** command. This command is used in this example to confirm that multicast traffic is being forwarded to the last hop router shown in Figure 1. In the sample output, notice the Mpackets In/Out field. This field displays the number of multicast packets received by and sent on each interface listed in the output.

```
Router# show ip pim interface count

State: * - Fast Switched, D - Distributed Fast Switched
       H - Hardware Switching Enabled
Address          Interface                FS Mpackets In/Out
172.31.100.2     Ethernet0/0              *   4122/0
10.1.0.1         Ethernet1/0              *   0/3193
```

The following is sample output from the **show ip mroute** command with the **count** keyword. This command is used on the last hop router shown in Figure 1 to verify the packets being sent to groups from active sources. In the sample output, notice the packet count displayed for the Forwarding field. This field displays the packet forwarding count for sources sending to groups.

```
Router# show ip mroute count

IP Multicast Statistics
6 routes using 4008 bytes of memory
3 groups, 1.00 average sources per group
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per second
Other counts: Total/RPF failed/Other drops(OIF-null, rate-limit etc)

Group: 239.1.2.3, Source count: 1, Packets forwarded: 3165, Packets received: 3165
```

```
  RP-tree: Forwarding: 0/0/0/0, Other: 0/0/0
  Source: 10.0.0.1/32, Forwarding: 3165/20/28/4, Other: 0/0/0

Group: 224.0.1.39, Source count: 1, Packets forwarded: 21, Packets received: 120
  Source: 172.16.0.1/32, Forwarding: 21/1/48/0, Other: 120/0/99

Group: 224.0.1.40, Source count: 1, Packets forwarded: 10, Packets received: 10
  Source: 172.16.0.1/32, Forwarding: 10/1/48/0, Other: 10/0/0
```

The following is sample output from the **show ip mroute** command with the **active** keyword. This command is used on the last hop router shown in Figure 1 to confirm the multicast groups with active sources on the last hop router.

Note In the output, the "(?)" indicates that the router is unable to resolve an IP address to a hostname.

```
Router# show ip mroute active

Active IP Multicast Sources - sending >= 4 kbps

Group: 239.1.2.3, (?)
   Source: 10.0.0.1 (?)
     Rate: 20 pps/4 kbps(1sec), 4 kbps(last 50 secs), 4 kbps(life avg)
```

Verifying IP Multicast on Routers Along the SPT: Example

The following is sample output from the **show ip mroute** for a particular group. This command is used in this example to verify that the RPF neighbor toward the source is the expected RPF neighbor for the router along the SPT shown in Figure 1.

```
Router# show ip mroute 239.1.2.3

(*, 239.1.2.3), 00:17:56/00:03:02, RP 172.16.0.1, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0/0, Forward/Sparse-Dense, 00:17:56/00:03:02

(10.0.0.1, 239.1.2.3), 00:15:34/00:03:28, flags: T
  Incoming interface: Serial1/0, RPF nbr 172.31.200.1
  Outgoing interface list:
    Ethernet0/0, Forward/Sparse-Dense, 00:15:34/00:03:02
```

The following is sample output from the **show ip mroute** command with the **active** keyword from the router along the SPT shown in Figure 1. This command is used to confirm the multicast groups with active sources on this router.

Note In the output, the "(?)" indicates that the router is unable to resolve an IP address to a hostname.

```
Router# show ip mroute active

Active IP Multicast Sources - sending >= 4 kbps

Group: 239.1.2.3, (?)
   Source: 10.0.0.1 (?)
     Rate: 20 pps/4 kbps(1sec), 4 kbps(last 30 secs), 4 kbps(life avg)
```

Verifying IP Multicast on the First Hop Router: Example

The following is sample output from the **show ip mroute** for a particular group. This command is used in this example to verify the packets being sent to groups from active sources on the first hop router shown in Figure 1. In the sample output, notice the packet count displayed for the Forwarding field. This field displays the packet forwarding count for sources sending to groups on the first hop router.

Note The RPF nbr 0.0.0.0 field indicates that the source of an mroute has been reached.

```
Router# show ip mroute 239.1.2.3

(*, 239.1.2.3), 00:18:10/stopped, RP 172.16.0.1, flags: SPF
  Incoming interface: Serial1/0, RPF nbr 172.31.200.2
  Outgoing interface list: Null

(10.0.0.1, 239.1.2.3), 00:18:10/00:03:22, flags: FT
  Incoming interface: Ethernet0/0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial1/0, Forward/Sparse-Dense, 00:18:10/00:03:19
```

The following is sample output from the **show ip mroute** command with the **active** keyword from the first hop router shown in Figure 1:

Note In the output, the "(?)" indicates that the router is unable to resolve an IP address to a host name.

```
Router# show ip mroute active

Active IP Multicast Sources - sending >= 4 kbps

Group: 239.1.2.3, (?)
   Source: 10.0.0.1 (?)
     Rate: 20 pps/4 kbps(1sec), 4 kbps(last 30 secs), 4 kbps(life avg)
```

Additional References

The following sections provide references related to verifying IP multicast operation.

Related Documents

Related Topic	Document Title
Overview of the IP multicast technology area	"IP Multicast Technology Overview" module
PIM-SM and SSM concepts and configuration examples	"Configuring Basic IP Multicast" module
IP multicast commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Multicast Command Reference*

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	—

MIBs

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

RFCs

RFC	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	—

Technical Assistance

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

Feature Information for Verifying IP Multicast Operation

Table 4 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the IP Multicast Features Roadmap.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 4 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 4 Feature Information for Verifying IP Multicast Operation

Feature Name	Releases	Feature Information
This table is intentionally left blank because no features were introduced or modified in this module since Cisco IOS Release 12.2(1). This table will be updated when feature information is added to this module.	—	—

Customizing IGMP

First Published: May 2, 2005
Last Updated: January 14, 2008

Internet Group Management Protocol (IGMP) is used to dynamically register individual hosts in a multicast group on a particular LAN segment. Enabling Protocol Independent Multicast (PIM) on an interface also enables IGMP operation on that interface.

This module describes ways to customize IGMP, including how to:

- Configure the router to forward multicast traffic in the absence of directly connected IGMP hosts.
- Configure IGMP state limits to control the number of multicast streams sent to a router to minimize the possibility of denial of service (DoS) attacks with IGMP packets.
- Enable an IGMP Version 3 (IGMPv3) host stack so that the router can function as a multicast network endpoint or host.
- Enable routers to track each individual host that is joined to a particular group or channel in an IGMPv3 environment.
- Control access to an SSM network using IGMP extended access lists.
- Configure an IGMP proxy that enables hosts that are not directly connected to a downstream router to join a multicast group sourced from an upstream network.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Customizing IGMP" section on page 33.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Contents

Prerequisites for Customizing IGMP

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

Restrictions for Customizing IGMP

Traffic Filtering with Multicast Groups That Are Not Configured in SSM Mode

IGMPv3 membership reports are not utilized by Cisco IOS software to filter or restrict traffic for multicast groups that are not configured in Source Specific Multicast (SSM) mode. Effectively, Cisco IOS software interprets all IGMPv3 membership reports for groups configured in dense, sparse, or bidirectional mode to be group membership reports and forwards traffic from all active sources onto the network.

Interoperability with IGMP Snooping

You must be careful when using IGMPv3 with switches that support and are enabled for IGMP snooping, because IGMPv3 messages are different from the messages used in IGMP Version 1 (IGMPv1) and Version 2 (IGMPv2). If a switch does not recognize IGMPv3 messages, then hosts will not correctly receive traffic if IGMPv3 is being used. In this case, either IGMP snooping may be disabled on the switch or the router may be configured for IGMPv2 on the interface, which would remove the ability to use SSM for host applications that cannot resort to URL Rendezvous Directory (URD) or IGMP v3lite.

Interoperability with CGMP

Networks using Cisco Group Management Protocol (CGMP) will have better group leave behavior if they are configured with IGMPv2 than IGMPv3. If CGMP is used with IGMPv2 and the switch is enabled for the CGMP leave functionality, then traffic to a port joined to a multicast group will be removed from the port shortly after the last member on that port has dropped membership to that group. This fast-leave mechanism is part of IGMPv2 and is specifically supported by the CGMP fast-leave enabled switch.

With IGMPv3, there is currently no CGMP switch support of fast leave. If IGMPv3 is used in a network, CGMP will continue to work, but CGMP fast-leave support is ineffective and the following conditions apply:

- Each time a host on a new port of the CGMP switch joins a multicast group, that port is added to the list of ports to which the traffic of this group is sent.
- If all hosts on a particular port leave the multicast group, but there are still hosts on other ports (in the same virtual LAN) joined to the group, then nothing happens. In other words, the port continues to receive traffic from that multicast group.
- Only when the last host in a virtual LAN (VLAN) has left the multicast group does forwarding of the traffic of this group into the VLAN revert to no ports on the forwarding switch.

This join behavior only applies to multicast groups that actually operate in IGMPv3 mode. If legacy hosts only supporting IGMPv2 are present in the network, then groups will revert to IGMPv2 and fast leave will work for these groups.

If fast leave is needed with CGMP-enabled switches, we recommend that you not enable IGMPv3 but configure IGMPv2 on that interface.

If you want to use SSM, you need IGMPv3 and you have two configuration alternatives, as follows:

- Configure only the interface for IGMPv2 and use IGMP v3lite and URD.
- Enable IGMPv3 and accept the higher leave latencies through the CGMP switch.

Information About Customizing IGMP

Before you customize IGMP, you should understand the following concepts:

- Role of the Internet Group Management Protocol, page 3
- IGMP Version Differences, page 4
- IGMP Join Process, page 6
- IGMP Leave Process, page 6
- IGMP Multicast Addresses, page 7

Role of the Internet Group Management Protocol

IGMP is used to dynamically register individual hosts in a multicast group on a particular LAN. Enabling PIM on an interface also enables IGMP. IGMP provides a means to automatically control and limit the flow of multicast traffic throughout your network with the use of special multicast queriers and hosts.

- A querier is a network device, such as a router, that sends query messages to discover which network devices are members of a given multicast group.
- A host is a receiver, including routers, that sends report messages (in response to query messages) to inform the querier of a host membership.

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

Hosts identify group memberships by sending IGMP messages to their local multicast router. Under IGMP, routers listen to IGMP messages and periodically send out queries to discover which groups are active or inactive on a particular subnet.

IGMP Version Differences

Table 1 provides high-level descriptions of the three IGMP versions.

Table 1 IGMP Versions

IGMP Version	Description
IGMPv1	Provides the basic query-response mechanism that allows the multicast router to determine which multicast groups are active and other processes that enable hosts to join and leave a multicast group. RFC 1112 defines the IGMPv1 host extensions for IP multicasting.
IGMPv2	Extends IGMP, allowing such capabilities as the IGMP leave process, group-specific queries, and an explicit maximum response time field. IGMPv2 also adds the capability for routers to elect the IGMP querier without dependence on the multicast protocol to perform this task. RFC 2236 defines IGMPv2.
IGMPv3	Provides for source filtering, which enables a multicast receiver host to signal to a router which groups it wants to receive multicast traffic from, and from which sources this traffic is expected. In addition, IGMPv3 supports the link local address 224.0.0.22, which is the destination IP address for IGMPv3 membership reports; all IGMPv3-capable multicast routers must listen to this address. RFC 3376 defines IGMPv3.

Note By default, enabling a PIM on an interface enables IGMPv2 on that router. IGMPv2 was designed to be as backward compatible with IGMPv1 as possible. To accomplish this backward compatibility, RFC 2236 defined special interoperability rules. If your network contains legacy IGMPv1 hosts, you should be familiar with these operability rules. For more information about IGMPv1 and IGMPv2 interoperability, see RFC 2236, *Internet Group Management Protocol, Version 2*.

Routers That Run IGMPv1

IGMPv1 routers send IGMP queries to the "all-hosts" multicast address of 224.0.0.1 to solicit multicast groups with active multicast receivers. The multicast receivers also can send IGMP reports to the router to notify it that they are interested in receiving a particular multicast stream. Hosts can send the report asynchronously or in response to the IGMP queries sent by the router. If more than one multicast receiver exists for the same multicast group, only one of these hosts sends an IGMP report message; the other hosts suppress their report messages.

In IGMPv1, there is no election of an IGMP querier. If more than one router on the segment exists, all the routers send periodic IGMP queries. IGMPv1 has no special mechanism by which the hosts can leave the group. If the hosts are no longer interested in receiving multicast packets for a particular group, they simply do not reply to the IGMP query packets sent from the router. The router continues sending query packets. If the router does not hear a response in three IGMP queries, the group times out and the router stops sending multicast packets on the segment for the group. If the host later wants to receive multicast packets after the timeout period, the host simply sends a new IGMP join to the router, and the router begins to forward the multicast packet again.

If there are multiple routers on a LAN, a designated router (DR) must be elected to avoid duplicating multicast traffic for connected hosts. PIM routers follow an election process to select a DR. The PIM router with the highest IP address becomes the DR.

The DR is responsible for the following tasks:

- Sending PIM register and PIM Join and Prune messages toward the rendezvous point (RP) to inform it about host group membership.
- Sending IGMP host-query messages.
- Sending host-query messages by default every 60 seconds in order to keep the IGMP overhead on hosts and networks very low.

Routers That Run IGMPv2

IGMPv2 improves the query messaging capabilities of IGMPv1.

The query and membership report messages in IGMPv2 are identical to the IGMPv1 messages with two exceptions:

- IGMPv2 query messages are broken into two categories: general queries (identical to IGMPv1 queries) and group-specific queries.
- IGMPv1 membership reports and IGMPv2 membership reports have different IGMP type codes.

IGMPv2 also enhances IGMP by providing support for the following capabilities:

- Querier election process—Provides the capability for IGMPv2 routers to elect the IGMP querier without having to rely on the multicast routing protocol to perform the process.
- Maximum Response Time field—A new field in query messages permits the IGMP querier to specify the maximum query-response time. This field permits the tuning of the query-response process to control response burstiness and to fine-tune leave latencies.
- Group-Specific Query messages—Permits the IGMP querier to perform the query operation on a specific group instead of all groups.
- Leave-Group messages—Provides hosts with a method of notifying routers on the network that they wish to leave the group.

Unlike IGMPv1, in which the DR and the IGMP querier are typically the same router, in IGMPv2 the two functions are decoupled. The DR and the IGMP querier are selected based on different criteria and may be different routers on the same subnet. The DR is the router with the highest IP address on the subnet, whereas the IGMP querier is the router with the lowest IP address.

Query messages are used to elect the IGMP querier as follows:

1. When IGMPv2 routers start, they each multicast a general query message to the all-systems group address of 224.0.0.1 with their interface address in the source IP address field of the message.
2. When an IGMPv2 router receives a general query message, the router compares the source IP address in the message with its own interface address. The router with the lowest IP address on the subnet is elected the IGMP querier.
3. All routers (excluding the querier) start the query timer, which is reset whenever a general query message is received from the IGMP querier. If the query timer expires, it is assumed that the IGMP querier has gone down, and the election process is performed again to elect a new IGMP querier.

By default, the timer is two times the query interval.

Routers Running IGMPv3

IGMPv3 adds support in Cisco IOS software for source filtering, which enables a multicast receiver host to signal to a router which groups it wants to receive multicast traffic from, and from which sources this traffic is expected. This membership information enables Cisco IOS software to forward traffic only from those sources from which receivers requested the traffic.

IGMPv3 supports applications that explicitly signal sources from which they want to receive traffic. With IGMPv3, receivers signal membership to a multicast group in the following two modes:

- INCLUDE mode—In this mode, the receiver announces membership to a group and provides a list of IP addresses (the INCLUDE list) from which it wants to receive traffic.
- EXCLUDE mode—In this mode, the receiver announces membership to a group and provides a list of IP addresses (the EXCLUDE list) from which it does not want to receive traffic. In other words, the host wants to receive traffic only from sources whose IP addresses are not listed in the EXCLUDE list. To receive traffic from all sources, like in the case of the Internet Standard Multicast (ISM) service model, a host expresses EXCLUDE mode membership with an empty EXCLUDE list.

IGMPv3 is the industry-designated standard protocol for hosts to signal channel subscriptions in an SSM network environment. For SSM to rely on IGMPv3, IGMPv3 must be available in the network stack portion of the operating systems running on the last hop routers and hosts and be used by the applications running on those hosts.

In IGMPv3, hosts send their membership reports to 224.0.0.22; all IGMPv3 routers, therefore, must listen to this address. Hosts, however, do not listen or respond to 224.0.0.22; they only send their reports to that address. In addition, in IGMPv3, there is no membership report suppression because IGMPv3 hosts do not listen to the reports sent by other hosts. Therefore, when a general query is sent out, all hosts on the wire respond.

IGMP Join Process

When a host wants to join a multicast group, the host sends one or more unsolicited membership reports for the multicast group it wants to join. The IGMP join process is the same in for IGMPv1 and IGMPv2 hosts.

In IGMPv3, the join process for hosts proceeds as follows:

- When a hosts wants to join a group, it sends an IGMPv3 membership report to 224.0.0.22 with an empty EXCLUDE list.
- When a host wants to join a specific channel, it sends an IGMPv3 membership report to 224.0.0.22 with the address of the specific source included in the INCLUDE list.
- When a host wants to join a group excluding particular sources, it sends an IGMPv3 membership report to 224.0.0.22 excluding those sources in the EXCLUDE list.

Note If some IGMPv3 hosts on a LAN wish to exclude a source and others wish to include the source, then the router will send traffic for the source on the LAN (that is, inclusion trumps exclusion in this situation).

IGMP Leave Process

The method that hosts use to leave a group varies depending on the version of IGMP in operation.

IGMPv1 Leave Process

There is no leave-group message in IGMPv1 to notify the routers on the subnet that a host no longer wants to receive the multicast traffic from a specific group. The host simply stops processing traffic for the multicast group and ceases responding to IGMP queries with IGMP membership reports for the group. As a result, the only way IGMPv1 routers know that there are no longer any active receivers for a particular multicast group on a subnet is when the routers stop receiving membership reports. To facilitate this process, IGMPv1 routers associate a countdown timer with an IGMP group on a subnet. When a membership report is received for the group on the subnet, the timer is reset. For IGMPv1

routers, this timeout interval is typically three times the query interval (3 minutes). This timeout interval means that the router may continue to forward multicast traffic onto the subnet for up to 3 minutes after all hosts have left the multicast group.

IGMPv2 Leave Process

IGMPv2 incorporates a leave-group message that provides the means for a host to indicate that it wishes to stop receiving multicast traffic for a specific group. When an IGMPv2 host leaves a multicast group, if it was the last host to respond to a query with a membership report for that group, it sends a leave-group message to the all-routers multicast group (224.0.0.2).

IGMPv3 Leave Process

IGMPv3 enhances the leave process by introducing the capability for a host to stop receiving traffic from a particular group, source, or channel in IGMP by including or excluding sources, groups, or channels in IGMPv3 membership reports.

IGMP Multicast Addresses

IP multicast traffic uses group addresses, which are Class D IP addresses. The high-order four bits of a Class D address are 1110. Therefore, host group addresses can be in the range 224.0.0.0 to 239.255.255.255.

Multicast addresses in the range 224.0.0.0 to 224.0.0.255 are reserved for use by routing protocols and other network control traffic. The address 224.0.0.0 is guaranteed not to be assigned to any group.

IGMP packets are transmitted using IP multicast group addresses as follows:

- IGMP general queries are destined to the address 224.0.0.1 (all systems on a subnet).
- IGMP group-specific queries are destined to the group IP address for which the router is querying.
- IGMP group membership reports are destined to the group IP address for which the router is reporting.
- IGMPv2 leave-group messages are destined to the address 224.0.0.2 (all routers on a subnet).
- IGMPv3 membership reports are destined to the address 224.0.0.22; all IGMPv3-capable multicast routers must listen to this address.

How to Customize IGMP

This section contains the following tasks:

- Configuring the Router to Forward Multicast Traffic in the Absence of Directly Connected IGMP Hosts, page 8 (optional)
- Configuring IGMP State Limits, page 9 (optional)
- Enabling the IGMPv3 Host Stack, page 11 (optional)
- Configuring IGMPv3—Explicit Tracking of Hosts, Groups, and Channels, page 13 (optional)
- Controlling Access to an SSM Network Using IGMP Extended Access Lists, page 15 (optional)
- Configuring an IGMP Proxy, page 19 (optional)

Configuring the Router to Forward Multicast Traffic in the Absence of Directly Connected IGMP Hosts

Perform this optional task to configure the router to forward multicast traffic in the absence of directly connected IGMP hosts.

Sometimes either there is no group member on a network segment or a host cannot report its group membership using IGMP. However, you may want multicast traffic to go to that network segment. The following are two ways to pull multicast traffic down to a network segment:

- Use the **ip igmp join-group** interface configuration command. With this method, the router accepts the multicast packets in addition to forwarding them. Accepting the multicast packets prevents the router from fast switching.
- Use the **ip igmp static-group** interface configuration command. With this method, the router does not accept the packets itself, but only forwards them. Hence, this method allows fast switching. When the **ip igmp static-group** command is configured, the outgoing interface will appear in the IGMP cache, but the router itself will not be a member of the group, as evidenced by lack of an "L" (local) flag in the multicast route entry.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip igmp join-group** *group-address*
   or
   ip igmp static-group {***** | *group-address* [**source** *source-address*]}
5. **end**
6. **show ip igmp interface** [*interface-type interface-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Enters interface configuration mode. • For the *type* and *number* arguments, specify an interface that is connected to hosts.

	Command or Action	Purpose
Step 4	**ip igmp join-group** *group-address* or **ip igmp static-group** {***** | *group-address* [**source** *source-address*]} **Example:** `Router(config-if)# ip igmp join-group 225.2.2.2` or **Example:** `Router(config-if)# ip igmp static-group 225.2.2.2`	Configures the router to forward multicast traffic in the absence of directly connected IGMP hosts. • Use the **ip igmp join-group** command to configure an interface on the router to join the specified group. • With this method, the router accepts the multicast packets in addition to forwarding them. Accepting the multicast packets prevents the router from fast switching. or • Use the **ip igmp static-group** command to configure static group membership entries on an interface. • With this method, the router does not accept the packets itself, but only forwards them. Hence, this method allows fast switching. The outgoing interface appears in the IGMP cache, but the router itself is not a member, as evidenced by lack of an "L" (local) flag in the multicast route entry.
Step 5	**end** **Example:** `Router(config-if)# end`	Ends the current configuration session and returns to privileged EXEC mode.
Step 6	**show ip igmp interface** [*interface-type interface-number*] **Example:** `Router# show ip igmp interface`	(Optional) Displays multicast-related information about an interface.

Configuring IGMP State Limits

Perform this optional task to limit the number of mroute states resulting from IGMP membership states per interface, per subinterface, or globally. Configuring IGMP state limits can reduce the vulnerability of a router to DoS attacks with IGMP packets. A high rate of IGMP messages sent to a router can pose a DoS attack scenario because the router processes IGMP, IGMP v3lite, and URD messages at the process level.

Feature Design of the IGMP State Limit

The IGMP State Limit feature introduces the capability to limit the number of mroute states resulting from IGMP membership states per interface, per subinterface, or globally. Membership reports exceeding the configured limits are not entered into the IGMP cache and traffic for the excess membership reports is not forwarded.

Per-interface and global IGMP limits operate independently of each other. Both per-interface and global IGMP limits can be configured on the same router. A membership report that exceeds either the per-interface or the global state limit is ignored.

Note When configuring IGMP state limits, you can only configure one global limit and one limit per interface.

The **ip igmp limit** command is used to configure IGMP state limits:

- Configuring the **ip igmp limit** command in global configuration mode configures a global limit on the number of mroute states resulting from IGMP membership states.
- Configuring the **ip igmp limit** command in interface configuration mode limits the number of mroute states resulting from IGMP membership states on a per-interface basis.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp limit** *number*
4. **interface** *type number*
5. **ip igmp limit** *number* [**except** *access-list*]
6. **end**
7. **show ip igmp interface** [*interface-type interface-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip igmp limit** *number* **Example:** `Router(config)# ip igmp limit 150`	Configures a global limit on the number of mroute states resulting from IGMP membership states.
Step 4	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Enters interface configuration mode. • For the *type* and *number* arguments, specify an interface that is connected to hosts.
Step 5	**ip igmp limit** *number* [**except** *access-list*] **Example:** `Router(config-if)# ip igmp limit 100`	Limits the number of mroute states resulting from IGMP membership states on a per-interface basis. • Use the **except** *access-list* keyword and argument to exclude certain groups or channels from being counted against the limit.

	Command or Action	Purpose
Step 6	**end** **Example:** `Router(config-if)# end`	Ends the current configuration session and returns to privileged EXEC mode.
Step 7	**show ip igmp interface** [*interface-type interface-number*] **Example:** `Router# show ip igmp interface`	(Optional) Displays multicast-related information about an interface.

Enabling the IGMPv3 Host Stack

Perform this optional task to add INCLUDE mode capability to the IGMPv3 host stack for SSM groups.

IGMPv3 Host Stack

The IGMPv3 Host Stack feature enables routers or switches to function as multicast network endpoints or hosts. The feature adds INCLUDE mode capability to the IGMPv3 host stack for SSM groups. Enabling the IGMPv3 host stack ensures that hosts on a LAN can leverage SSM by enabling the router or switch to initiate IGMPv3 joins, such as in environments where fast channel change is required in a SSM deployments.

In support of the IGMPv3 Host Stack feature, the **source** keyword and *source-address* argument were added to the **ip igmp join-group** command to add INCLUDE mode capability to the IGMPv3 host stack for SSM groups.

Note Multiple **ip igmp join-group** command configurations with different source addresses for the same group are supported.

When the IGMPv3 Host Stack feature is configured, an IGMPv3 membership report is sent when one of the following events occurs:

- When the **ip igmp join-group** command is configured for a group and source and there is no existing state for this (S, G) channel, an IGMPv3 report of group record type ALLOW_NEW_SOURCES for the source specified is sent on that interface.
- When the **no** form of the **ip igmp join-group** command is configured for a group and source and there is state for this (S, G) channel, an IGMPv3 report of group record type BLOCK_OLD_SOURCES for the source specified is sent on that interface.
- When a query is received, an IGMPv3 report is sent as defined in RFC 3376.

Note For more information about IGMPv3 group record types and membership reports, see RFC 3376, *Internet Group Management Protocol, Version 3*.

Prerequisites

- This task requires the routers to be running Cisco IOS Release 12.3(14)T or a subsequent release.
- This task assumes that the router has been configured for SSM. For information about how to configure SSM, see the "Configuring Basic IP Multicast" module in the *Cisco IOS IP Multicast Configuration Guide*.

Restrictions

- IGMP version 3 must be configured on the interface.

Note If the **ip igmp join-group** command is configured for a group and source and IGMPv3 is *not* configured on the interface, (S, G) state will be created but no IGMPv3 membership reports will be sent.

- The router must be configured for SSM. IGMPv3 membership reports will only be sent for SSM channels.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip igmp version 3**
5. **ip igmp join-group** *group-address* **source** *source-address*
6. Repeat Step 5 to provide INCLUDE mode capability for additional (S, G) channels.
7. **end**
8. **show ip igmp groups detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface FastEthernet 1`	Enters interface configuration mode. • For the *type* and *number* arguments, specify an interface that is connected to hosts.

	Command or Action	Purpose
Step 4	**ip igmp version 3** **Example:** `Router(config-if)# ip igmp version 3`	Enables IGMPv3 on the interface.
Step 5	**ip igmp join-group** *group-address* **source** *source-address* **Example:** `Router(config-if)# ip igmp join-group 232.2.2.2 source 10.1.1.1`	Configures the interface to join the specified (S, G) channel. • This command enables the router to provide INCLUDE mode capability for the (S, G) channel specified for the *group-address* and *source-address* arguments.
Step 6	Repeat Step 5 to provide INCLUDE mode capability for additional (S, G) channels.	—
Step 7	**end** **Example:** `Router(config-if)# end`	Ends the current configuration session and returns to privileged EXEC mode.
Step 1	**show ip igmp groups detail** **Example:** `Router# show ip igmp groups detail`	Displays the multicast groups with receivers that are directly connected to the router and that were learned through IGMP. • Use this command to verify that the router has received membership reports for (S, G) channels configured using the **ip igmp join group** command. When the router is correctly receiving IGMP membership reports for a channel, the "Flags:" output field will display the L and SSM flags.

Configuring IGMPv3—Explicit Tracking of Hosts, Groups, and Channels

Perform this optional task to enable a multicast router to explicitly track the membership of all multicast hosts in a particular multiaccess network. This enhancement to the Cisco IOS implementation of IGMPv3 enables the router to track each individual host that is joined to a particular group or channel.

Benefits of IGMPv3—Explicit Tracking of Hosts, Groups, and Channels

Minimal Leave Latencies

The main benefit of the IGMPv3—Explicit Tracking of Hosts, Groups, and Channels feature is to allow minimal leave latencies when a host leaves a multicast group or channel. A router configured with IGMPv3 and explicit tracking can immediately stop forwarding traffic if the last host to request to receive traffic from the router indicates that it no longer wants to receive traffic. The leave latency is thus bound only by the packet transmission latencies in the multiaccess network and the processing time in the router.

In IGMPv2, when a router receives an IGMP leave message from a host, it must first send an IGMP group-specific query to learn if other hosts on the same multiaccess network are still requesting to receive traffic. If after a specific time (in Cisco IOS software, the default value is approximately 3 seconds) no host replies to the query, the router will then stop forwarding the traffic. This query

process is required because, in IGMPv1 and IGMPv2, IGMP membership reports are suppressed if the same report has already been sent by another host in the network. Therefore, it is impossible for the router to reliably know how many hosts on a multiaccess network are requesting to receive traffic.

Faster Channel Changing

In networks where bandwidth is constrained between multicast routers and hosts (such as in DSL deployments), the bandwidth between routers and hosts is typically large enough to sustain, in general, only a limited number of multicast streams to be received in parallel. In these deployments, each host will typically join to only one multicast stream and the overall number of allowed hosts will be limited by the total bandwidth of the link. The effective leave latency in these environments defines the channel change time of the receiver application—a single host cannot receive the new multicast stream before forwarding of the old stream has stopped. If an application tries to change the channel faster than the leave latency, the application will overload the bandwidth of the access network, resulting in a temporary degradation of traffic flow for all hosts. The IGMPv3—Explicit Tracking of Hosts, Groups, and Channels feature allows for minimal leave latencies, and thus allows for fast channel changing capabilities.

Improved Diagnostics Capabilities

The IGMPv3—Explicit Tracking of Hosts, Groups, and Channels feature allows network administrators to easily determine which multicast hosts are joined to which multicast groups or channels.

Restrictions

No MIB Support

There is no Simple Network Management Protocol (SNMP) MIB to track the IGMP membership of individual hosts. The MIBs supported by Cisco IOS software reflect only the aggregate membership of a particular interface on a router.

No Minimal Leave Latency for Groups with Legacy Hosts

If one or more hosts that supports only IGMPv1 or IGMPv2 are present on a network, the leave latencies for the multicast groups to which those hosts are joined will revert to the leave latencies of the IGMP version of the hosts—approximately 3 seconds for IGMPv2 and up to 180 seconds (3 minutes) for IGMPv1. This condition affects only the multicast groups to which those legacy hosts are actually joined at any given point in time. In addition, the membership reports for these multicast groups sent by IGMPv3 hosts may revert to IGMPv1 or IGMPv2 membership reports, thus disabling explicit tracking of those host memberships.

No Explicit Tracking Support for IGMP v3lite and URD

Explicit tracking of IGMP Version 3 lite (IGMP v3lite) or URL Rendezvous Directory (URD) channel membership reports is not supported in Release 12.0(19)S or earlier releases. In these releases, the leave latency for multicast groups sending traffic to hosts using IGMP v3lite or URD will be determined by the leave latency of the version of IGMP configured on the hosts (for IGMPv3, the leave latency is typically 3 seconds when explicit tracking is not configured).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*

4. **ip igmp version 3**
5. **ip igmp explicit-tracking**
6. **end**
7. **show ip igmp membership** [*group-address* | *group-name*] [**tracked**] [**all**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Enters interface configuration mode. • For the *type* and *number* arguments, specify an interface that is connected to hosts.
Step 4	**ip igmp version 3** **Example:** `Router(config-if)# ip igmp version 3`	Enables IGMPv3.
Step 5	**ip igmp explicit-tracking** **Example:** `Router(config-if)# ip igmp explicit-tracking`	Enables explicit tracking of hosts, groups, and channels for IGMPv3.
Step 6	**end** **Example:** `Router(config-if)# end`	Ends the current configuration session and returns to privileged EXEC mode.
Step 7	**show ip igmp membership** [*group-address* | *group-name*] [**tracked**] [**all**] **Example:** `Router# show ip igmp membership`	(Optional) Displays IGMP membership information for multicast groups and (S, G) channels.

Controlling Access to an SSM Network Using IGMP Extended Access Lists

IGMPv3 includes support for source filtering, that is, the ability for a system to report interest in receiving packets only from specific source addresses, or from all but specific source addresses, sent to a particular multicast address.

Perform this optional task to control access to an SSM network by using an IGMP extended access list that filters SSM traffic based on source address, group address, or both. For more information on SSM, see the "Configuring Basic IP Multicast" module. For general information about how to configure an access list, see the "Creating an IP Access List and Applying It to an Interface" module.

Benefits of Extended Access List Support for IGMP to Support SSM in IPv4

IGMPv3 accommodates extended access lists, which allow you to leverage an important advantage of SSM in IPv4, that of basing access on source IP address. Prior to this feature, an IGMP access list accepted only a standard access list, allowing membership reports to be filtered based only on multicast group addresses.

IGMPv3 allows multicast receivers not only to join to groups, but to groups including or excluding sources. For appropriate access control, it is therefore necessary to allow filtering of IGMPv3 messages not only by group addresses reported, but by group and source addresses. IGMP extended access lists introduce this functionality. Using SSM with an IGMP extended access list (ACL) allows you to permit or deny source S and group G (S, G) in IGMPv3 reports, thereby filtering IGMPv3 reports based on source address, group address, or source and group address.

Source Addresses in IGMPv3 Reports for ASM Groups

IGMP extended access lists also can be used to permit or filter (deny) traffic based on (0.0.0.0, G), that is, (*, G) in IGMP reports that are non-SSM, such as Any Source Multicast (ASM).

Note The permit and deny statements equivalent to (*, G) are **permit host 0.0.0.0 host** *group-address* and **deny host 0.0.0.0 host group** *group-address*, respectively.

Filtering applies to IGMPv3 reports for both ASM and SSM groups, but it is most important for SSM groups because Cisco IOS IP multicast routing ignores source addresses in IGMPv3 reports for ASM groups. Source addresses in IGMPv3 membership reports for ASM groups are stored in the Cisco IOS IGMP cache (as displayed with the **show ip igmp membership** command), but PIM-based IP multicast routing considers only the ASM groups reported. Therefore, adding filtering for source addresses for ASM groups impacts only the IGMP cache for ASM groups.

How IGMP Checks an Extended Access List

When an IGMP extended access list is referenced in the **ip igmp access-group** command on an interface, the (S, G) pairs in the **permit** and **deny** statements of the extended access list are matched against the (S, G) pair of the IGMP reports received on the interface. For example, if an IGMP report with (S1, S2...Sn, G) is received, first the group (0.0.0.0, G) is checked against the access list statements. The convention (0.0.0.0, G) means (*, G), which is a wildcard source with a multicast group number. If the group is denied, the entire IGMP report is denied. If the group is permitted, each individual (S, G) pair is checked against the access list. Denied sources are taken out of the IGMP report, thereby denying the sources access to the multicast traffic.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast-routing** [**distributed**]
4. **ip pim ssm** {**default** | **range** *access-list*}

5. **ip access-list extended** *access-list-name*
6. **deny igmp** *source source-wildcard destination destination-wildcard* [*igmp-type*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
7. **permit igmp** *source source-wildcard destination destination-wildcard* [*igmp-type*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
8. **end**
9. **interface** *type number*
10. **ip igmp access-group** *access-list*
11. **ip pim sparse-mode**
12. Repeat Steps 1 through 11 on all interfaces that require access control of SSM channel membership.
13. **ip igmp version 3**
14. Repeat Step 13 on all host-facing interfaces.
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast-routing** [**distributed**] **Example:** `Router(config)# ip multicast-routing`	Enables IP multicast routing. • Use the **distributed** keyword to enable Multicast Distributed Switching.
Step 4	**ip pim ssm** {**default** | **range** *access-list*} **Example:** `Router(config)# ip pim ssm default`	Configures SSM service. • The **default** keyword defines the SSM range access list as 232/8. • The **range** keyword specifies the standard IP access list number or name that defines the SSM range.
Step 5	**ip access-list extended** *access-list*-name **Example:** `Router(config)# ip access-list extended mygroup`	Specifies an extended named IP access list.

	Command or Action	Purpose
Step 6	**deny igmp** *source source-wildcard destination destination-wildcard* [*igmp-type*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config-ext-nacl)# deny igmp host 10.1.2.3 any`	(Optional) Filters the specified source address or group address from the IGMP report, thereby restricting hosts on a subnet from membership to the (S, G) channel. • Repeat this step to restrict hosts on a subnet membership to other (S, G) channels. (These sources should be more specific than a subsequent **permit** statement because any sources or groups not specifically permitted are denied.) • Remember that the access list ends in an implicit **deny** statement. • This example creates a deny statement that filters all groups for source 10.1.2.3, which effectively denies the source.
Step 7	**permit igmp** *source source-wildcard destination destination-wildcard* [*igmp-type*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config-ext-nacl)# permit igmp any any`	Allows a source address or group address in an IGMP report to pass the IP access list. • You must have at least one **permit** statement in an access list. • Repeat this step to allow other sources to pass the IP access list. • This example allows group membership to sources and groups not denied by prior **deny** statements.
Step 8	**end** **Example:** `Router(config-ext-nacl)# end`	Ends the current configuration session and returns to privileged EXEC mode.
Step 9	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Selects an interface that is connected to hosts on which IGMPv3 can be enabled.
Step 10	**ip igmp access-group** *access-list* **Example:** `Router(config-if)# ip igmp access-group mygroup`	Applies the specified access list to IGMP reports.
Step 11	**ip pim sparse-mode** **Example:** `Router(config-if)# ip pim sparse-mode`	Enables PIM-SM on the interface. **Note** You must use sparse mode.
Step 12	Repeat Steps 1 through 11 on all interfaces that require access control of SSM channel membership.	—
Step 13	**ip igmp version 3** **Example:** `Router(config-if)# ip igmp version 3`	Enables IGMPv3 on this interface. The default version of IGMP is set to Version 2. Version 3 is required by SSM.

	Command or Action	Purpose
Step 14	Repeat Step 13 on all host-facing interfaces.	—
Step 15	**end** **Example:** `Router(config-if)# end`	Ends the current configuration session and returns to privileged EXEC mode.

Configuring an IGMP Proxy

Perform this optional task to configure unidirectional link (UDL) routers to use the IGMP proxy mechanism. An IGMP proxy enables hosts in a unidirectional link routing (UDLR) environment that are not directly connected to a downstream router to join a multicast group sourced from an upstream network.

To configure an IGMP proxy, you will need to perform the following tasks:

- Configuring the Upstream UDL Router for IGMP UDLR, page 21
- Configuring the Downstream UDL Router for IGMP UDLR with IGMP Proxy Support, page 22

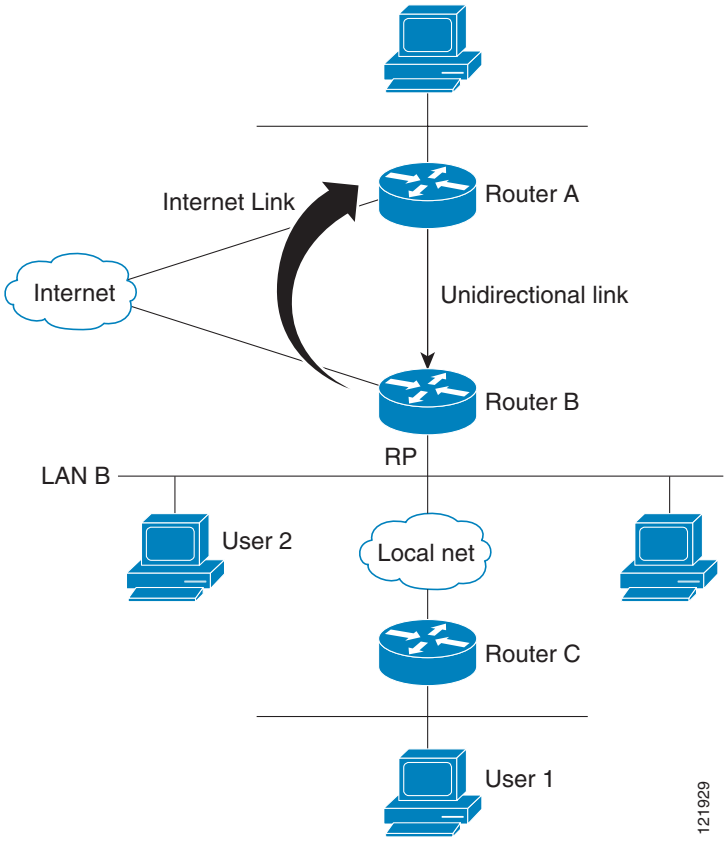
Figure 1 illustrates a sample topology that shows two UDLR scenarios:

- Traditional UDL routing scenario—A UDL router with directly connected receivers.
- IGMP proxy scenario—UDL router without directly connected receivers.

Note IGMP UDLs are needed on the upstream and downstream routers. For more information about IGMP UDLs, see the "Configuring IP Multicast Over Unidirectional Links" module in the *Cisco IOS IP Multicast Configuration Guide*.

Note For an example of this task, see the "Configuring an IGMP Proxy: Example" section.

Figure 1 IGMP Proxy Topology



Scenario 1—Traditional UDLR Scenario (UDL Router with Directly Connected Receivers)

For scenario 1, no IGMP proxy mechanism is needed. In this scenario, the following sequence of events occurs:

1. User 2 sends an IGMP membership report requesting interest in group G.
2. Router B receives the IGMP membership report, adds a forwarding entry for group G on LAN B, and proxies the IGMP report to Router A, which is the UDLR upstream router.
3. The IGMP report is then proxied across the Internet link.
4. Router A receives the IGMP proxy and maintains a forwarding entry on the unidirectional link.

Scenario 2—IGMP Proxy Scenario (UDL Router without Directly Connected Receivers)

For scenario 2, the IGMP proxy mechanism is needed to enable hosts that are not directly connected to a downstream router to join a multicast group sourced from an upstream network. In this scenario, the following sequence of events occurs:

1. User 1 sends an IGMP membership report requesting interest in group G.
2. Router C sends a PIM Join message hop-by-hop to the RP (Router B).
3. Router B receives the PIM Join message and adds a forwarding entry for group G on LAN B.
4. Router B periodically checks its mroute table and proxies the IGMP membership report to its upstream UDL router across the Internet link.
5. Router A creates and maintains a forwarding entry on the unidirectional link (UDL).

In an enterprise network, it is desirable to be able to receive IP multicast traffic via satellite and forward the traffic throughout the network. With unidirectional link routing (UDLR) alone, scenario 2 would not be possible because receiving hosts must be directly connected to the downstream router, Router B. The IGMP proxy mechanism overcomes this limitation by creating an IGMP report for (*, G) entries in the multicast forwarding table. To make this scenario functional, therefore, you must enable IGMP report forwarding of proxied (*, G) multicast static route (mroute) entries (using the **ip igmp mroute-proxy** command) and enable the mroute proxy service (using the **ip igmp proxy-service** command) on interfaces leading to PIM-enabled networks with potential members.

Note Because PIM messages are not forwarded upstream, each downstream network and the upstream network have a separate domain.

Prerequisites

Before configuring an IGMP proxy, ensure that the following conditions exist:

- All routers on the IGMP UDL have the same subnet address. If all routers on the UDL cannot have the same subnet address, the upstream router must be configured with secondary addresses to match all the subnets that the downstream routers are attached to.
- This task assumes that IP multicast has been enabled and that the PIM interfaces have been configured using the tasks described in the "Configuring Basic IP Multicast" module.

  When enabling PIM on the interfaces for the IGMP proxy scenario, keep in mind the following guidelines:

  - Use PIM sparse mode (PIM-SM) when the interface is operating in a sparse-mode region and you are running static RP, bootstrap (BSR), or Auto-RP with the Auto-RP listener capability.
  - Use PIM sparse-dense mode when the interface is running in a sparse-dense mode region and you are running Auto-RP without the Auto-RP listener capability.
  - Use PIM dense mode (PIM-DM) for this step when the interface is operating in dense mode and is, thus, participating in a dense-mode region.
  - Use PIM-DM with the proxy-register capability when the interface is receiving source traffic from a dense-mode region that needs to reach receivers that are in a sparse-mode region.

Configuring the Upstream UDL Router for IGMP UDLR

Perform this task to configure the upstream UDL router for IGMP UDLR (Router A in Figure 1).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip igmp unidirectional-link**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Enters interface configuration mode. • For the *type* and *number* arguments, specify the interface to be used as the UDL on the upstream router.
Step 4	**ip igmp unidirectional-link** **Example:** `Router(config-if)# ip igmp unidirectional-link`	Configures IGMP on the interface to be unidirectional for IGMP UDLR.
Step 5	**end** **Example:** `Router(config-if)# end`	Ends the current configuration session and returns to privileged EXEC mode.

Configuring the Downstream UDL Router for IGMP UDLR with IGMP Proxy Support

Perform this task to configure the downstream UDL router for IGMP UDLR with IGMP proxy support (Router B in Figure 2).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip igmp unidirectional-link**
5. **exit**
6. **interface** *type number*
7. **ip igmp mroute-proxy** *type number*
8. **exit**
9. **interface** *type number*
10. **ip igmp helper-address udl** *interface-type interface-number*
11. **ip igmp proxy-service**

12. **end**
13. **show ip igmp interface**
14. **show ip igmp udlr**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Enters interface configuration mode. • For the *type* and *number* arguments, specify the interface to be used as the UDL on the downstream router for IGMP UDLR.
Step 4	**ip igmp unidirectional-link** **Example:** `Router(config-if)# ip igmp unidirectional-link`	Configures IGMP on the interface to be unidirectional for IGMP UDLR.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Enters interface configuration mode. • For the *type* and *number* arguments, select an interface that is facing the nondirectly hosts.
Step 7	**ip igmp mroute-proxy** *type number* **Example:** `Router(config-if)# ip igmp mroute-proxy loopback 0`	Enables IGMP report forwarding of proxied (*, G) multicast static route (mroute) entries. • This step is performed to enable the forwarding of IGMP reports to a proxy service interface for all (*, G) forwarding entries in the multicast forwarding table. • In this example, the **ip igmp mroute-proxy** command is configured on Ethernet interface 1 to request that IGMP reports be sent to loopback interface 0 for all groups in the mroute table that are forwarded to Ethernet interface 1.
Step 8	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 9	**interface** *type number* **Example:** `Router(config)# interface loopback 0`	Enters interface configuration mode for the specified interface. • In this example, loopback interface 0 is specified.
Step 10	**ip igmp helper-address udl** *interface-type interface-number* **Example:** `Router(config-if)# ip igmp helper-address udl ethernet 0`	Configures IGMP helpering for UDLR. • This step allows the downstream router to helper IGMP reports received from hosts to an upstream router connected to a UDL associated with the interface specified for the *interface-type* and *interface-number* arguments. • In the example topology, IGMP helpering is configured over loopback interface 0 on the downstream router. Loopback interface 0, thus, is configured to helper IGMP reports from hosts to an upstream router connected to Ethernet interface 0.
Step 11	**ip igmp proxy-service** **Example:** `Router(config-if)# ip igmp proxy-service`	Enables the mroute proxy service. • When the mroute proxy service is enabled, the router periodically checks the static mroute table for (*, G) forwarding entries that match interfaces configured with the **ip igmp mroute-proxy** command (see Step 7) based on the IGMP query interval. Where there is a match, one IGMP report is created and received on this interface. **Note** The **ip igmp proxy-service** command is intended to be used with the **ip igmp helper-address** (UDL) command. • In this example, the **ip igmp proxy-service** command is configured on loopback interface 0 to enable the forwarding of IGMP reports out the interface for all groups on interfaces registered through the **ip igmp mroute-proxy** command (see Step 7).
Step 12	**end** **Example:** `Router(config-if)# end`	Ends the current configuration session and returns to privileged EXEC mode.
Step 13	**show ip igmp interface** **Example:** `Router# show ip igmp interface`	(Optional) Displays multicast-related information about an interface.
Step 14	**show ip igmp udlr** **Example:** `Router# show ip igmp udlr`	(Optional) Displays UDLR information for directly connected multicast groups on interfaces that have a UDL helper address configured.

Configuration Examples for Customizing IGMP

This section provides the following configuration examples:

Configuring the Router to Forward Multicast Traffic in the Absence of Directly Connected IGMP Hosts: Examples

The following example shows how to configure a router to forward multicast traffic in the absence of directly connected IGMP hosts using the **ip igmp join-group** command. With this method, the router accepts the multicast packets in addition to forwarding them. Accepting the multicast packets prevents the router from fast switching.

In this example, Fast Ethernet interface 0/0 on the router is configured to join the group 225.2.2.2:

```
interface FastEthernet0/0
 ip igmp join-group 225.2.2.2
```

The following example shows how to configure a router to forward multicast traffic in the absence of directly connected IGMP hosts using the **ip igmp static-group** command. With this method, the router does not accept the packets itself, but only forwards them. Hence, this method allows fast switching. The outgoing interface appears in the IGMP cache, but the router itself is not a member, as evidenced by lack of an “L” (local) flag in the multicast route entry.

In this example, static group membership entries for group 225.2.2.2 are configured on Fast Ethernet interface 0/1:

```
interface FastEthernet0/0
 ip igmp static-group 225.2.2.2
```

Configuring IGMP State Limits: Example

The following example shows how to configure an IGMP state limit. In this example, the number of IGMP membership reports on Ethernet interface 0 is limited to 100 reports and IGMP membership reports permitted by access list 199 do not count toward the configured state limit.

```
interface ethernet 0
 ip igmp limit 100 except 199
```

The following sample output from the **show ip igmp interface** command illustrates the IGMP limit of 100 IGMP membership reports and no reports permitted by access list 199 count toward the limit:

```
Router# show ip igmp interface

Ethernet0 is up, line protocol is up
  Internet address is 192.168.37.6, subnet mask is 255.255.255.0
```

```
  IGMP is enabled on interface
  IGMP query interval is 60 seconds
  Inbound IGMP access group is not set
  ip igmp limit 100 except 199
  Multicast routing is enabled on interface
  Multicast TTL threshold is 0
  Multicast designated router (DR) is 192.168.37.33
  No multicast groups joined
Ethernet1 is up, line protocol is up
  Internet address is 192.168.36.129, subnet mask is 255.255.255.0
  IGMP is enabled on interface
  IGMP query interval is 60 seconds
  Inbound IGMP access group is not set
  ip igmp limit 100 except 199
  Multicast routing is enabled on interface
  Multicast TTL threshold is 0
  Multicast designated router (DR) is 192.168.36.131
  Multicast groups joined: 225.2.2.2 226.2.2.2
Tunnel0 is up, line protocol is up
  Internet address is 10.1.37.2, subnet mask is 255.255.0.0
  IGMP is enabled on interface
  IGMP query interval is 60 seconds
  Inbound IGMP access group is not set
  ip igmp limit 100 except 199
  Multicast routing is enabled on interface
  Multicast TTL threshold is 0
 No multicast groups joined
```

Enabling the IGMPv3 Host Stack: Example

The following example shows how to add INCLUDE mode capability to the IGMPv3 host stack for SSM groups:

```
interface FastEthernet0/0
 ip igmp join-group 232.2.2.2 source 10.1.1.1
 ip igmp join-group 232.2.2.2 source 10.5.5.5
 ip igmp join-group 232.2.2.2 source 10.5.5.6
 ip igmp join-group 232.2.2.4 source 10.5.5.5
 ip igmp join-group 232.2.2.4 source 10.5.5.6
 ip igmp version 3
```

Based on the configuration presented in this example, the following is sample output from the **debug igmp** command. The messages confirm that IGMPv3 membership reports are being sent after IGMPv3 and SSM are enabled:

```
Router# debug igmp

*May 4 23:48:34.251: IGMP(0): Group 232.2.2.2 is now in the SSM range, changing
*May 4 23:48:34.251: IGMP(0): Building v3 Report on Ethernet0/0
*May 4 23:48:34.251: IGMP(0): Add Group Record for 232.2.2.2, type 5
*May 4 23:48:34.251: IGMP(0): Add Source Record 10.1.1.1
*May 4 23:48:34.251: IGMP(0): Add Source Record 10.5.5.5
*May 4 23:48:34.251: IGMP(0): Add Source Record 10.5.5.6
*May 4 23:48:34.251: IGMP(0): Add Group Record for 232.2.2.2, type 6
*May 4 23:48:34.251: IGMP(0): No sources to add, group record removed from report
*May 4 23:48:34.251: IGMP(0): Send unsolicited v3 Report with 1 group records on
FastEthernet0/0
*May 4 23:48:34.251: IGMP(0): Group 232.2.2.4 is now in the SSM range, changing
*May 4 23:48:34.251: IGMP(0): Building v3 Report on Ethernet0/0
*May 4 23:48:34.251: IGMP(0): Add Group Record for 232.2.2.4, type 5
*May 4 23:48:34.251: IGMP(0): Add Source Record 10.5.5.5
```

```
*May 4 23:48:34.251: IGMP(0): Add Source Record 10.5.5.6
*May 4 23:48:34.251: IGMP(0): Add Group Record for 232.2.2.4, type 6
*May 4 23:48:34.251: IGMP(0): No sources to add, group record removed from report
*May 4 23:48:34.251: IGMP(0): Send unsolicited v3 Report with 1 group records on
FastEthernet0/0
*May 4 23:48:35.231: IGMP(0): Building v3 Report on Ethernet0/0
*May 4 23:48:35.231: IGMP(0): Add Group Record for 232.2.2.2, type 5
*May 4 23:48:35.231: IGMP(0): Add Source Record 10.1.1.1
*May 4 23:48:35.231: IGMP(0): Add Source Record 10.5.5.5
*May 4 23:48:35.231: IGMP(0): Add Source Record 10.5.5.6
*May 4 23:48:35.231: IGMP(0): Add Group Record for 232.2.2.2, type 6
*May 4 23:48:35.231: IGMP(0): No sources to add, group record removed from report
*May 4 23:48:35.231: IGMP(0): Send unsolicited v3 Report with 1 group records on
FastEthernet0/0
*May 4 23:48:35.231: IGMP(0): Building v3 Report on Ethernet0/0
*May 4 23:48:35.231: IGMP(0): Add Group Record for 232.2.2.4, type 5
*May 4 23:48:35.231: IGMP(0): Add Source Record 10.5.5.5
*May 4 23:48:35.231: IGMP(0): Add Source Record 10.5.5.6
*May 4 23:48:35.231: IGMP(0): Add Group Record for 232.2.2.4, type 6
*May 4 23:48:35.231: IGMP(0): No sources to add, group record removed from report
*May 4 23:48:35.231: IGMP(0): Send unsolicited v3 Report with 1 group records on
FastEthernet0/0
```

The following is sample output from the **show ip igmp groups** command with the **detail** keyword for this configuration example scenario. The **show ip igmp groups** command can be used to verify that the router has received membership reports for (S, G) channels configured using the **ip igmp join group** command. When the router is correctly receiving IGMP membership reports for a channel, the "Flags:" output field will display the L and SSM flags.

```
Router# show ip igmp groups detail

Flags: L - Local, U - User, SG - Static Group, VG - Virtual Group,
       SS - Static Source, VS - Virtual Source

Interface:      FastEthernet0/0
Group:          232.2.2.2
Flags:          L SSM
Uptime:         00:04:12
Group mode:     INCLUDE
Last reporter:  10.4.4.7
Group source list: (C - Cisco Src Report, U - URD, R - Remote, S - Static,
                    V - Virtual, Ac - Accounted towards access control limit,
                    M - SSM Mapping, L - Local)
  Source Address   Uptime    v3 Exp   CSR Exp   Fwd  Flags
  10.1.1.1         00:04:10  stopped   stopped    Yes  L
  10.5.5.5         00:04:12  stopped   stopped    Yes  L
  10.5.5.6         00:04:12  stopped   stopped    Yes  L

Interface:      FastEthernet0/0
Group:          232.2.2.3
Flags:          L SSM
Uptime:         00:04:12
Group mode:     INCLUDE
Last reporter:  10.4.4.7
Group source list: (C - Cisco Src Report, U - URD, R - Remote, S - Static,
                    V - Virtual, Ac - Accounted towards access control limit,
                    M - SSM Mapping, L - Local)
  Source Address   Uptime    v3 Exp   CSR Exp   Fwd  Flags
  10.5.5.5         00:04:14  stopped   stopped   Yes  L
  10.5.5.6         00:04:14  stopped   stopped   Yes  L
```

Configuring IGMPv3—Explicit Tracking of Hosts, Groups, and Channels: Example

The following example shows how to enable explicit tracking. The example shows a basic configuration for enabling IP multicast with SSM, IGMPv3, and explicit tracking.

```
ip multicast-routing
interface ethernet 0
 description access network to desktop systems
 ip address 10.1.0.1 255.255.255.0
 ip pim sparse-dense-mode
 ip mroute-cache
 ip igmp version 3
 ip igmp explicit-tracking
interface ethernet 1
 description backbone interface no connected hosts
 ip address 10.10.0.1 255.255.255.0
 ip pim sparse-dense-mode
 ip mroute-cache
ip pim ssm default
```

Controlling Access to an SSM Network Using IGMP Extended Access Lists: Examples

This section contains the following configuration examples for controlling access to an SSM network using IGMP extended access lists:

- Denying All States for a Group G: Example, page 28
- Denying All States for a Source S: Example, page 29
- Permitting All States for a Group G: Example, page 29
- Permitting All States for a Source S: Example, page 29
- Filtering a Source S for a Group G: Example, page 29

Note Keep in mind that access lists are very flexible: there are many combinations of permit and deny statements one could use in an access list to filter multicast traffic. The examples in this section simply provide a few examples of how it can be done.

Denying All States for a Group G: Example

The following example shows how to deny all states for a group G. In this example, FastEthernet interface 0/0 is configured to filter all sources for SSM group 232.2.2.2 in IGMPv3 reports, which effectively denies this group.

```
ip access-list extended test1
 deny igmp any host 232.2.2.2
 permit igmp any any
!
interface FastEthernet0/0
 ip igmp access-group test1
```

Denying All States for a Source S: Example

The following example shows how to deny all states for a source S. In this example, Ethernet interface 1/1 is configured to filter all groups for source 10.2.1.32 in IGMPv3 reports, which effectively denies this source.

```
ip access-list extended test2
 deny igmp host 10.2.1.32 any
 permit igmp any any
!
interface Ethernet1/1
 ip igmp access-group test2
```

Permitting All States for a Group G: Example

The following example shows how to permit all states for a group G. In this example, Ethernet interface 1/2 is configured to accept all sources for SSM group 232.1.1.10 in IGMPv3 reports, which effectively accepts this group altogether.

```
ip access-list extended test3
 permit igmp any host 232.1.1.10
!
interface Ethernet1/2
 ip igmp access-group test3
```

Permitting All States for a Source S: Example

The following example shows how to permit all states for a source S. In this example, Ethernet interface 1/2 is configured to accept all groups for source 10.6.23.32 in IGMPv3 reports, which effectively accepts this source altogether.

```
ip access-list extended test4
 permit igmp host 10.6.23.32 any
!
interface Ethernet1/2
 ip igmp access-group test4
```

Filtering a Source S for a Group G: Example

The following example shows how to filter a particular source S for a group G. In this example, Ethernet interface 0/3 is configured to filter source 232.2.2.2 for SSM group 232.2.30.30 in IGMPv3 reports.

```
ip access-list extended test5
 deny igmp host 10.4.4.4 host 232.2.30.30
 permit igmp any any
!
interface Ethernet0/3
 ip igmp access-group test5
```

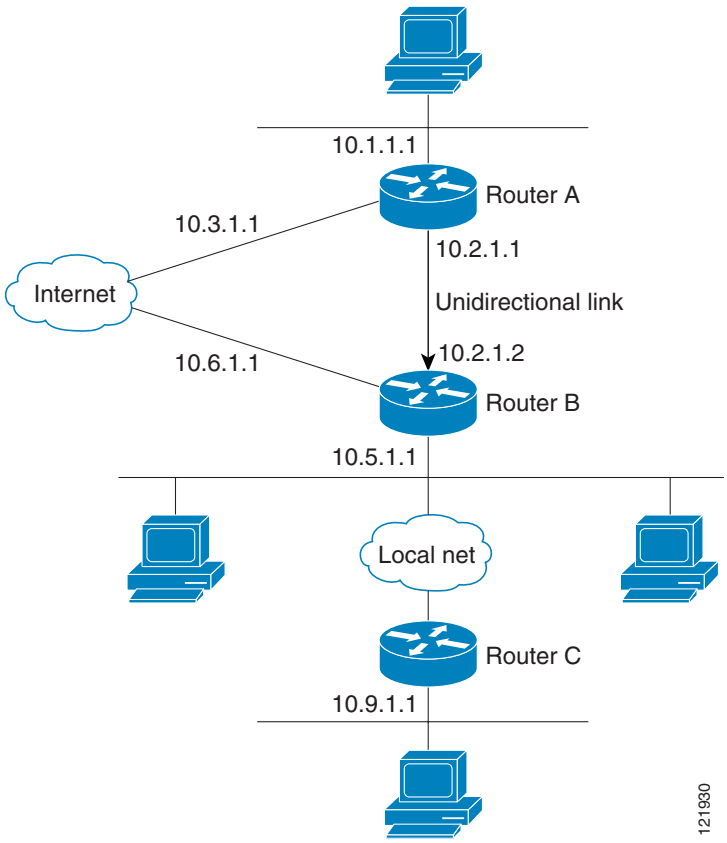
Configuring an IGMP Proxy: Example

The following example shows how to configure the upstream UDL router for IGMP UDLR and the downstream UDL router for IGMP UDLR with IGMP proxy support. The IGMP proxy mechanism is needed to enable hosts that are not directly connected to a downstream router to join a multicast group sourced from an upstream network.

The example is based on the topology illustrated in Figure 2. In this example topology, Router A is the upstream router and Router B is the downstream router.

Note For more details about configuring an IGMP proxy, see the "Configuring an IGMP Proxy" section.

Figure 2 IGMP Proxy Example Topology



Router A Configuration

```
interface ethernet 0
ip address 10.1.1.1 255.255.255.0
ip pim dense-mode
!
interface ethernet 1
ip address 10.2.1.1 255.255.255.0
ip pim dense-mode
ip igmp unidirectional-link
!
interface ethernet 2
ip address 10.3.1.1 255.255.255.0
```

Router B Configuration

```
ip pim rp-address 10.5.1.1 5
access-list 5 permit 239.0.0.0 0.255.255.255
!
interface loopback 0
ip address 10.7.1.1 255.255.255.0
ip pim dense-mode
```

```
ip igmp helper-address udl ethernet 0
ip igmp proxy-service
!
interface ethernet 0
ip address 10.2.1.2 255.255.255.0
ip pim dense-mode
ip igmp unidirectional-link
!
interface ethernet 1
ip address 10.5.1.1 255.255.255.0
ip pim sparse-mode
ip igmp mroute-proxy loopback 0
!
interface ethernet 2
ip address 10.6.1.1 255.255.255.0
```

Additional References

The following sections provide references related to customizing IGMP.

Related Documents

Related Topic	Document Title
Overview of the IP multicast technology area	"IP Multicast Technology Overview" module in the *Cisco IOS IP Multicast Configuration Guide*
Basic IP multicast concepts, configuration tasks, and examples	"Configuring Basic IP Multicast" module in the *Cisco IOS IP Multicast Configuration Guide*
IGMP UDLR concepts, configuration tasks, and examples	"Configuring IP Multicast over Unidirectional Links" module in the *Cisco IOS IP Multicast Configuration Guide*
IP multicast commands: complete command syntax, command mode, command history, command defaults, usage guidelines, and examples	*Cisco IOS IP Multicast Command Reference*

Standards

Standard	Title
No new or modified standards are supported by these features, and support for existing standards has not been modified by these features.	—

MIBs

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

RFCs

RFC	Title
RFC 1112	*Host extensions for IP multicasting*
RFC 2236	*Internet Group Management Protocol, Version 2*
RFC 3376	*Internet Group Management Protocol, Version 3*

Technical Assistance

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

Feature Information for Customizing IGMP

Table 2 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Releases 12.2(1) or 12.0(3)S or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the "IP Multicast Features Roadmap."

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 2 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 2 Feature Information for Customizing IGMP

Feature Name	Releases	Feature Information
Extended ACL Support for IGMP to Support SSM in IPv4	12.0(19)S 12.3(7)T 12.2(25)S 12.2(27)SBC 12.2(33)SRA 12.2(33)SXH	The Extended ACL Support for IGMP to Support SSM in IPv4 feature enables IGMPv3 to accommodate extended access lists. IGMPv3 support of extended access lists allows you to leverage an important advantage of SSM in IPv4, that of filtering IGMPv3 reports based on source address, group address, or both. The following sections provide information about this feature: • Controlling Access to an SSM Network Using IGMP Extended Access Lists, page 15 • Controlling Access to an SSM Network Using IGMP Extended Access Lists: Examples, page 28 The following command was introduced by this feature: **ip igmp access-group**.
Extended ACL support for IGMP to support SSM in IPv4	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Table 2 Feature Information for Customizing IGMP (continued)

Feature Name	Releases	Feature Information
IGMPv3 Host Stack	12.3(14)T	The IGMPv3 Host Stack feature enables routers and switches to function as multicast network endpoints or hosts. The feature adds INCLUDE mode capability to the IGMPv3 host stack for SSM groups. Enabling the IGMPv3 host stack ensures that hosts on a LAN can leverage SSM by enabling the router to initiate IGMPv3 joins, such as in environments where fast channel change is required in a SSM deployments. The following sections provide information about this feature: • Enabling the IGMPv3 Host Stack, page 11 • Enabling the IGMPv3 Host Stack: Example, page 26 The following command was modified by this feature: **ip igmp join-group**.
IGMP State Limit	12.2(14)S 12.2(15)T	The IGMP State Limit feature introduces the capability to limit the number of mroute states resulting from IGMP membership states per interface, per subinterface, or globally. Membership reports exceeding the configured limits are not entered into the IGMP cache and traffic for the excess membership reports is not forwarded. The following sections provide information about this feature: • Configuring IGMP State Limits, page 9 • Configuring IGMP State Limits: Example, page 25 The following commands were introduced or modified by this feature: **ip igmp limit** (global), **ip igmp limit** (interface), **show ip igmp interface**.

Table 2 Feature Information for Customizing IGMP (continued)

Feature Name	Releases	Feature Information
IGMP State Limit	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
IGMPv3—Explicit Tracking Host, Group, and Channel	12.0(19)S 12.2(8)T 12.2(14)S	This IGMPv3—Explicit Tracking Host, Group, and Channel feature enables a multicast router to explicitly track the membership of all multicast hosts in a particular multiaccess network. This enhancement to the Cisco IOS implementation of IGMPv3 enables the router to track each individual host that is joined to a particular group or channel. The following sections provide information about this feature: • Configuring IGMPv3—Explicit Tracking of Hosts, Groups, and Channels, page 13 • Configuring IGMPv3—Explicit Tracking of Hosts, Groups, and Channels: Example, page 28 The following commands were introduced by this feature: **ip igmp explicit-tracking**, **show ip igmp membership**.

Optimizing PIM Sparse Mode in a Large IP Multicast Deployment

This module describes how to optimize Protocol Independent Multicast (PIM) sparse mode for a large deployment of IP multicast. You can set a limit on the rate of PIM register messages sent in order to limit the load on the designated router and RP, you can reduce the PIM router query message interval to achieve faster convergence, and you can delay or prevent the use of the shortest path tree.

Module History

This module was first published on May 2, 2005, and last updated on May 2, 2005.

Finding Feature Information in This Document

Not all features may be supported in your Cisco IOS software release. Use the "Feature Information for Optimizing PIM Sparse Mode in a Large IP Multicast Deployment" to find information about feature support and configuration.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Optimizing PIM Sparse Mode in a Large IP Multicast Deployment

This module assumes you have met the following prerequisites:

- You have PIM sparse mode running in your network.
- You understand the concepts in the "IP Multicast Technology Overview" module.
- If you plan to use a group list to control which groups the shortest-path tree (SPT) threshold applies to, you have configured your access list before performing the task.

Information About Optimizing PIM Sparse Mode in a Large IP Multicast Deployment

Before optimizing PIM sparse mode for a large deployment, you should understand the following concepts:

- PIM Registering Process, page 2
- PIM Designated Router, page 3
- PIM Sparse-Mode Register Messages, page 3
- Preventing Use of Shortest-Path Tree to Reduce Memory Requirement, page 4

PIM Registering Process

IP multicast sources do not use a signaling mechanism to announce their presence. Sources just send their data into the attached network, as opposed to receivers that use Internet Group Management Protocol (IGMP) to announce their presence. If a source sends traffic to a multicast group configured in PIM sparse mode (PIM-SM), the Designated Router (DR) leading toward the source must inform the rendezvous point (RP) about the presence of this source. If the RP has downstream receivers that want to receive the multicast traffic (natively) from this source and has not joined the shortest path leading toward the source, then the DR must send the traffic from the source to the RP. The PIM registering process, which is individually run for each (S, G) entry, accomplishes these tasks between the DR and RP.

The registering process begins when a DR creates a new (S, G) state. The DR encapsulates all the data packets that match the (S, G) state into PIM register messages and unicasts those register messages to the RP.

If an RP has downstream receivers that want to receive register messages from a new source, the RP can either continue to receive the register messages through the DR or join the shortest path leading toward the source. By default, the RP will join the shortest path, because delivery of native multicast traffic provides the highest throughput. Upon receipt of the first packet that arrives natively through the shortest path, the RP will send a register-stop message back to the DR. When the DR receives this register-stop message, it will stop sending register messages to the RP.

If an RP has no downstream receivers that want to receive register messages from a new source, the RP will not join the shortest path. Instead, the RP will immediately send a register-stop message back to the DR. When the DR receives this register-stop message, it will stop sending register messages to the RP.

Once a routing entry is established for a source, a periodic reregistering takes place between the DR and RP. One minute before the multicast routing table state times out, the DR will send one dataless register message to the RP each second that the source is active until the DR receives a register-stop message from the RP. This action restarts the timeout time of the multicast routing table entry, typically resulting in one reregistering exchange every 2 minutes. Reregistering is necessary to maintain state, to recover from lost state, and to keep track of sources on the RP. It will take place independently of the RP joining the shortest path.

PIM Version 1 Compatibility

If an RP is running PIM Version 1, it will not understand dataless register messages. In this case, the DR will not send dataless register messages to the RP. Instead, approximately every 3 minutes after receipt of a register-stop message from the RP, the DR encapsulates the incoming data packets from the source into register messages and sends them to the RP. The DR continues to send register messages until it receives another register-stop message from the RP. The same behavior occurs if the DR is running PIM Version 1.

When a DR running PIM Version 1 encapsulates data packets into register messages for a specific (S, G) entry, the entry is process-switched, not fast-switched or hardware-switched. On platforms that support these faster paths, the PIM registering process for an RP or DR running PIM Version 1 may lead to periodic out-of-order packet delivery. For this reason, we recommend upgrading your network from PIM Version 1 to PIM Version 2.

PIM Designated Router

Routers configured for IP multicast send PIM hello messages to determine which router will be the designated router (DR) for each LAN segment (subnet). The hello messages contain the router's IP address, and the router with the highest IP address becomes the DR.

The DR sends Internet Group Management Protocol (IGMP) host query messages to all hosts on the directly connected LAN. When operating in sparse mode, the DR sends source registration messages to the rendezvous point (RP).

By default, multicast routers send PIM router query messages every 30 seconds. By enabling a router to send PIM hello messages more often, the router can discover unresponsive neighbors more quickly. As a result, the router can implement failover or recovery procedures more efficiently. It is appropriate to make this change only on redundant routers on the edge of the network.

PIM Sparse-Mode Register Messages

Dataless register messages are sent at a rate of one message per second. Continuous high rates of register messages might occur if a DR is registering bursty sources (sources with high data rates) and if the RP is not running PIM Version 2.

By default, PIM sparse-mode register messages are sent without limiting their rate. Limiting the rate of register messages will limit the load on the DR and RP, at the expense of dropping those register messages that exceed the set limit. Receivers may experience data packet loss within the first second in which packets are sent from bursty sources.

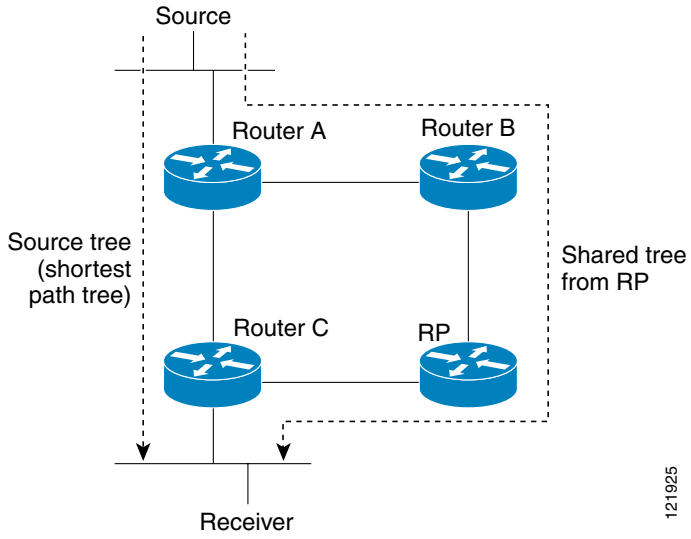
Preventing Use of Shortest-Path Tree to Reduce Memory Requirement

Understanding PIM shared tree and source tree will help you understand how preventing the use of the shortest-path tree can reduce memory requirements.

PIM Shared Tree and Source Tree (Shortest-Path Tree)

By default, members of a multicast group receive data from senders to the group across a single data distribution tree rooted at the rendezvous point (RP). This type of distribution tree is called shared tree, as shown in Figure 1. Data from senders is delivered to the RP for distribution to group members joined to the shared tree.

Figure 1 Shared Tree versus Source Tree (Shortest-Path Tree)



If the data rate warrants, leaf routers on the shared tree may initiate a switch to the data distribution tree rooted at the source. This type of distribution tree is called a shortest-path tree (SPT) or source tree. By default, the software switches to a source tree upon receiving 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 the RP.
2. The RP puts the link to Router C in its outgoing interface list.
3. Source sends data; Router A encapsulates data in a register message and sends it to the RP.
4. The RP forwards data down the shared tree to Router C and sends a Join message toward the source. At this point, data may arrive twice at Router C, once encapsulated and once natively.
5. When data arrives natively (through multicast) at the RP, the RP sends a register-stop message to Router A.
6. By default, reception of the first data packet prompts Router C to send a Join message toward the source.

7. When Router C receives data on (S, G), it sends a Prune message for the source up the shared tree.
8. The RP deletes the link to Router C from the outgoing interface of (S, G). The RP triggers a Prune message toward the 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 sent by the designated router that is directly connected to a source and are received by the RP for the group.

Multiple sources sending to groups use the shared tree.

Benefit of Preventing or Delaying the Use of the Shortest-Path Tree

The switch from shared to source tree happens upon the arrival of the first data packet at the last hop router (Router C in Figure 1). This switch occurs because the **ip pim spt-threshold** command controls that timing, and its default setting is 0 kbps.

The shortest-path tree requires more memory than the shared tree, but reduces delay. You might want to prevent or delay its use to reduce memory requirements. Instead of allowing the leaf router to move to the shortest-path tree immediately, you can prevent use of the SPT or specify that the traffic must first reach a threshold.

You can configure when a PIM leaf router should join the shortest-path tree for a specified group. If a source sends at a rate greater than or equal to the specified *kbps* rate, the router triggers a PIM Join message toward the source to construct a source tree (shortest-path tree). If the **infinity** keyword is specified, all sources for the specified group use the shared tree, never switching to the source tree.

How to Optimize PIM Sparse Mode in a Large IP Multicast Deployment

This section contains the following procedure:

- Optimizing PIM Sparse Mode in a Large Deployment, page 5 (optional)

Optimizing PIM Sparse Mode in a Large Deployment

Consider performing this task if your deployment of IP multicast is large.

Steps 3, 5, and 6 in this task are independent of each other and are therefore considered optional. Any one of these steps will help optimize PIM sparse mode. If you are going to perform Step 5 or 6, you must perform Step 4. Step 6 applies only to a designated router; changing the PIM query interval is only appropriate on redundant routers on the edge of the PIM domain.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip pim register-rate-limit** *rate*
4. **ip pim spt-threshold** {*kbps* | **infinity**} [**group-list** *access-list*]

5. **interface** *type number*
6. **ip pim query-interval** *period* [**msec**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip pim register-rate-limit** *rate* **Example:** `Router(config)# ip pim register-rate-limit 10`	(Optional) Sets a limit on the maximum number of PIM sparse mode register messages sent per second for each (S, G) routing entry. • Use this command to limit the number of register messages that the designated router (DR) will allow for each (S, G) entry. • By default, there is no maximum rate set. • Configuring this command will limit the load on the DR and RP at the expense of dropping those register messages that exceed the set limit. • Receivers may experience data packet loss within the first second in which register messages are sent from bursty sources.
Step 4	**ip pim spt-threshold** {*kbps* | **infinity**} [**group-list** *access-list*] **Example:** `Router(config)# ip pim spt-threshold infinity group-list 5`	(Optional) Specifies the threshold that must be reached before moving to the shortest-path tree. • The default value is **0**, which causes the router to join the SPT immediately upon the first data packet it receives. • Specifying the **infinity** keyword causes the router never to move to the shortest-path tree; it remains on the shared tree. This keyword applies to a multicast environment of "many-to-many" communication. • The group list is a standard access list that controls which groups the SPT threshold applies to. If a value of 0 is specified or the group list is not used, the threshold applies to all groups. • In the example, group-list 5 is already configured to permit the multicast groups 239.254.2.0 and 239.254.3.0: access-list 5 permit 239.254.2.0 0.0.0.255 access-list 5 permit 239.254.3.0 0.0.0.255

	Command or Action	Purpose
Step 5	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Configures an interface. • If you do not want to change the default values of the PIM SPT threshold or the PIM query interval, do not perform this step; you are done with this task.
Step 6	**ip pim query-interval** *period* [**msec**] **Example:** `Router(config-if)# ip pim query-interval 1`	(Optional) Configures the frequency at which multicast routers send PIM router query messages. • Perform this step only on redundant routers on the edge of a PIM domain. • The default query interval is 30 seconds. • The *period* argument is in seconds unless the **msec** keyword is specified. • Set the query interval to a smaller number of seconds for faster convergence, but keep in mind the trade-off between faster convergence and higher CPU and bandwidth usage.

Configuration Examples for Optimizing PIM Sparse Mode in a Large IP Multicast Deployment

This section provides the following configuration example:

- Optimizing PIM Sparse Mode in a Large IP Multicast Deployment: Example, page 8

Optimizing PIM Sparse Mode in a Large IP Multicast Deployment: Example

The following example shows how to:

- Set the query interval to 1 second for faster convergence.
- Configure the router to never move to the SPT but to remain on the shared tree.
- Set a limit of 10 PIM sparse mode register messages sent per second for each (S, G) routing entry.

```
interface ethernet 0
 ip pim query-interval 1
.
.
.
!
ip pim spt-threshold infinity
ip pim register-rate-limit 10
!
```

Where to Go Next

- To configure basic IP multicast, see the "Configuring Basic IP Multicast" module.
- To verify multicast operation, see the "Verifying IP Multicast Operation" module.

- To customize IGMP, see the "Customizing IGMP" module.
- To load split IP multicast traffic, see the "Load Splitting IP Multicast Traffic" module.
- To connect non-IP multicast areas, see the "Tunneling to Connect Non-IP Multicast Areas" module.
- To configure IP multicast over ATM point-to-multipoint VCs, see the "Configuring IP Multicast over ATM Point-to-Multipoint VCs" module.
- To configure IP multicast for operation in a switched Ethernet network, see the "Constraining IP Multicast in a Switched Ethernet Network" module.
- To configure IP multicast over unidirectional links, see the "Configuring IP Multicast over Unidirectional Links" module.
- To configure IP multicast for operation in a Virtual Private Network, see the "Configuring Multicast-VPN" module.
- To monitor and maintain IP multicast, see the "Monitoring and Maintaining IP Multicast" module.
- To validate a multicast test packet, see the "Using the Multicast Routing Monitor" module.

Additional References

The following sections provide references related to optimizing PIM sparse mode.

Related Documents

Related Topic	Document Title
IP multicast concepts and tasks	*Cisco IOS IP Multicast Configuration Guide*, Release 12.4
IP multicast commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Multicast Command Reference*, Release 12.4

MIBs

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

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 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/public/support/tac/home.shtml

Feature Information for Optimizing PIM Sparse Mode in a Large IP Multicast Deployment

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Releases 12.2(1) or later appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for specific commands was introduced, see the command reference documents.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator (http://www.cisco.com/go/fn). You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click Cancel at the login dialog box and follow the instructions that appear.

Table 1 Feature Information for Optimizing PIM Sparse Mode in a Large IP Multicast Deployment

Feature Name	Releases	Feature Configuration Information
This table is intentionally left blank because no features were introduced or modified in Cisco IOS Release 12.2(1) or later. This table will be updated when feature information is added to this module.	—	—

Multicast Subsecond Convergence

First Published: July 22, 2002
Last Updated: February 11, 2008

The Multicast Subsecond Convergence feature comprises a comprehensive set of features and protocol enhancements that provide for improved scalability and convergence in multicast-based services. This feature set provides for the ability to scale to larger services levels and to recover multicast forwarding after service failure in subsecond time frames.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Multicast Subsecond Convergence" section on page 12.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Multicast Subsecond Convergence

Service providers must have a multicast-enabled core in order to use the Cisco Multicast Subsecond Convergence feature.

Restrictions for Multicast Subsecond Convergence

Routers that use the subsecond designated router (DR) failover enhancement need to be able to process hello interval information arriving in milliseconds. Routers that are congested or do not have enough CPU cycles to process the hello interval may assume that the Protocol Independent Multicast (PIM) neighbor is disconnected, although this may not be the case.

Information About Multicast Subsecond Convergence

To configure the Multicast Subsecond Convergence feature, you must understand the following concepts:

Benefits of Multicast Subsecond Convergence

- The scalability components improve on the efficiency of handling increases (or decreases) in service users (receivers) and service load (sources or content).
- New algorithms and processes (such as aggregated join messages, which deliver up to 1000 individual messages in a single packet) reduce the time to reach convergence by a factor of 10.
- Multicast subsecond convergence improves service availability for large multicast networks.
- Multicast users such as financial services firms and brokerages receive better quality of service (QoS), because multicast functionality is restored in a fraction of the time previously required.

Multicast Subsecond Convergence Scalability Enhancements

The Multicast Subsecond Convergence feature provides scalability enhancements that improve on the efficiency of handling increases (or decreases) in service users (receivers) and service load (sources or content). Scalability enhancements in this release include the following:

- Improved Internet Group Management Protocol (IGMP) and PIM state maintenance through new timer management techniques
- Improved scaling of the Multicast Source Discovery Protocol (MSDP) Source-Active (SA) cache

The scalability enhancements provide the following benefits:

- Increased potential PIM multicast route (mroute), IGMP, and MSDP SA cache state capacity
- Decreased CPU usage

PIM Router Query Messages

Multicast subsecond convergence allows you to send PIM router query messages (PIM hellos) every few milliseconds. The PIM hello message is used to locate neighboring PIM routers. Before the introduction of this feature, you could send the PIM hellos every few seconds. By enabling a router to send PIM hello messages more often, this feature allows the router to discover unresponsive neighbors more quickly. As a result, the router can implement failover or recovery procedures more efficiently.

Reverse Path Forwarding

Unicast Reverse Path Forwarding (RPF) helps to mitigate problems caused by the introduction of malformed or forged IP source addresses into a network by discarding IP packets that lack a verifiable IP source address. Malformed or forged source addresses can indicate denial-of-service (DoS) attacks based on source IP address spoofing.

RPF uses access control lists (ACLs) in determining whether to drop or forward data packets that have malformed or forged IP source addresses. An option in the ACL commands allows system administrators to log information about dropped or forwarded packets. Logging information about forged packets can help in uncovering information about possible network attacks.

Per-interface statistics can help system administrators quickly discover the interface serving as the entry point for an attack on the network.

Triggered RPF Checks

Multicast subsecond convergence provides the ability to trigger a check of RPF changes for mroute states. This check is triggered by unicast routing changes. By performing a triggered RPF check, users can set the periodic RPF check to a relatively high value (for example, 10 seconds) and still fail over quickly.

The triggered RPF check enhancement reduces the time needed for service to be restored after disruption, such as for single service events (for example, in a situation with one source and one receiver) or as the service scales along any parameter (for example, many sources, many receivers, and many interfaces). This enhancement decreases in time-to-converge PIM (mroute), IGMP, and MSDP (SA cache) states.

Topology Changes and Multicast Routing Recovery

The Multicast Subsecond Convergence feature set enhances both enterprise and service provider network backbones by providing almost instantaneous recovery of multicast paths after unicast routing recovery.

Because PIM relies on the unicast routing table to calculate its RPF when a change in the network topology occurs, unicast protocols first need to calculate options for the best paths for traffic, and then multicast can determine the best path.

Multicast subsecond convergence allows multicast protocol calculations to finish almost immediately after the unicast calculations are completed. As a result, multicast traffic forwarding is restored substantially faster after a topology change.

How to Configure Multicast Subsecond Convergence

This section contains the following procedures:

- Modifying the Periodic RPF Check Interval, page 4 (optional)
- Configuring PIM RPF Failover Intervals, page 5 (optional)
- Modifying the PIM Router Query Message Interval, page 6 (optional)
- Verifying Multicast Subsecond Convergence Configurations, page 7 (optional)

Modifying the Periodic RPF Check Interval

Perform this task to modify the intervals at which periodic RPF checks occur.

RPF Checks

PIM is designed to forward IP multicast traffic using the standard unicast routing table. PIM uses the unicast routing table to decide if the source of the IP multicast packet has arrived on the optimal path from the source. This process, the RPF check, is protocol-independent because it is based on the contents of the unicast routing table and not on any particular routing protocol.

Restrictions

Cisco recommends that users keep the default values for the **ip rpf interval** command. The default values allow subsecond RPF failover. The default interval at which periodic RPF checks occur is 10 seconds.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast rpf interval** *seconds* [**list** *access-list* | **route-map** *route-map*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast rpf interval** *seconds* [**list** *access-list* | **route-map** *route-map*] **Example:** `Router(config)# ip multicast rpf interval 10`	Configures the periodic RPF check intervals to occur at a specified interval, in seconds.

What to Do Next

Proceed to the "Configuring PIM RPF Failover Intervals" section on page 5 to configure the intervals at which PIM RPF failover will be triggered by changes in the routing tables. Proceed to the "Modifying the PIM Router Query Message Interval" section on page 6 to modify the interval at which IGMP host query messages are sent. Proceed to the "Verifying Multicast Subsecond Convergence Configurations" section on page 7 to display information about and to verify information regarding the Multicast Subsecond Convergence feature.

Configuring PIM RPF Failover Intervals

Perform this task to configure the intervals at which PIM RPF failover will be triggered by changes in the routing tables.

RPF Failover

In an unstable unicast routing environment that uses triggered RPF checks, the environment could be constantly triggering RPF checks, which places a burden on the resources of the router. To avoid this problem, use the **ip multicast rpf backoff** command to prevent a second triggered RPF check from occurring for the length of time configured. That is, the PIM "backs off" from another triggered RPF check for a minimum amount of milliseconds as configured by the user.

If the backoff period expires without further routing table changes, PIM then scans for routing changes and accordingly establishes multicast RPF changes. However, if more routing changes occur during the backoff period, PIM doubles the backoff period to avoid overloading the router with PIM RPF changes while the routing table is still converging.

Restrictions

Cisco recommends that users keep the default values for the **ip multicast rpf backoff** command. The default values allow subsecond RPF failover.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast rpf backoff** *minimum maximum* [**disable**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast rpf backoff** *minimum maximum* [**disable**] **Example:** `Router(config)# ip multicast rpf backoff 100 2500`	Configures the minimum and the maximum backoff intervals.

What to Do Next

Proceed to the "Modifying the PIM Router Query Message Interval" section on page 6 to modify the interval at which IGMP host query messages are sent. Proceed to the "Verifying Multicast Subsecond Convergence Configurations" section on page 7 to display information about and to verify information regarding the Multicast Subsecond Convergence feature.

Modifying the PIM Router Query Message Interval

Perform this task to modify the PIM router query message interval.

PIM Router Query Messages

Router query (hello) messages are used to elect a PIM designated router. The designated router is responsible for sending IGMP host query messages. By default, multicast routers send PIM router query messages every 30 seconds.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/port*
4. **ip pim query-interval** *period* [**msec**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot/port* **Example:** `Router(config)# interface ethernet 1/0`	Specifies the interface and enters interface configuration mode.
Step 4	**ip pim query-interval** *period* [**msec**] **Example:** `Router(config-if)# ip pim query-interval 45`	Configures the frequency at which multicast routers send PIM router query messages.

What to Do Next

Proceed to the "Verifying Multicast Subsecond Convergence Configurations" section on page 7 to display information about and to verify information regarding the Multicast Subsecond Convergence feature.

Verifying Multicast Subsecond Convergence Configurations

Perform this task to display detailed information about and to verify information regarding the Multicast Subsecond Convergence feature.

SUMMARY STEPS

1. **enable**
2. **show ip pim interface** *type number*
3. **show ip pim neighbor**
4. **show ip rpf events**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

Step 2 **show ip pim interface** *type number*

Use this command to display information about interfaces configured for PIM.

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

```
Router# show ip pim interface Ethernet 1/0

Address          Interface                Ver/   Nbr    Query  DR     DR
                                          Mode   Count  Intvl  Prior
172.16.1.4       Ethernet1/0              v2/S   1      100 ms 1      172.16.1.4
```

Step 3 **show ip pim neighbor**

Use this command to display the PIM neighbors discovered by the Cisco IOS software.

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

```
Router# show ip pim neighbor

PIM Neighbor Table
Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
172.16.1.3        Ethernet1/0              00:03:41/250 msec v2    1 / S
```

Step 4 **show ip rpf events**

Use this command to display information regarding the last 15 triggered multicast RPF check events.

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

```
Router# show ip rpf events

Last 15 triggered multicast RPF check events

RPF backoff delay:500 msec
RPF maximum delay:5 sec

DATE/TIME             BACKOFF    PROTOCOL   EVENT         RPF CHANGES
Mar 7 03:24:10.505    500 msec   Static     Route UP        0
Mar 7 03:23:11.804    1000 sec   BGP        Route UP        3
Mar 7 03:23:10.796    500 msec   ISIS       Route UP        0
Mar 7 03:20:10.420    500 msec   ISIS       Route Down      3
Mar 7 03:19:51.072    500 msec   Static     Route Down      0
Mar 7 02:46:32.464    500 msec   Connected  Route UP        3
Mar 7 02:46:24.052    500 msec   Static     Route Down      0
Mar 7 02:46:10.200    1000 sec   Connected  Route UP        3
Mar 7 02:46:09.060    500 msec   OSPF       Route UP        3
Mar 7 02:46:07.416    500 msec   OSPF       Route Down      0
Mar 7 02:45:50.423    500 msec   EIGRP      Route UP        3
Mar 7 02:45:09.679    500 msec   EIGRP      Route Down      0
Mar 7 02:45:06.322    500 msec   EIGRP      Route Down      2
Mar 7 02:33:09.424    500 msec   Connected  Route UP        0
Mar 7 02:32:28.307    500 msec   BGP        Route UP        3
```

Configuration Examples for Multicast Subsecond Convergence

This section provides the following configuration examples

Modifying the Periodic RPF Check Interval Example

In the following example, the **ip multicast rpf interval** has been set to 10 seconds. This command does not show up in **show running-config** output unless the interval value has been configured to be the nondefault value.

```
!
ip multicast-routing
ip multicast rpf interval 10
.
.
.
interface Ethernet0/0
 ip address 172.16.2.1 255.255.255.0
.
.
.
ip pim sparse-mode
!
```

Configuring PIM RPF Failover Intervals Example

In the following example, the **ip multicast rpf backoff** command has been configured with a minimum backoff interval value of 100 and a maximum backoff interval value of 2500. This command does not show up in **show running-config** command output unless the interval value has been configured to be the nondefault value.

```
!
ip multicast-routing
.
.
.
ip multicast rpf backoff 100 2500
!
!

interface Ethernet0/0
 ip address 172.16.2.1 255.255.255.0
.
.
.
ip pim sparse-mode

!
```

Modifying the PIM Router Query Message Interval Example

In the following example, the **ip pim query-interval** command has been set to 100 milliseconds. This command does not show up in **show running-config** command output unless the interval value has been configured to be the nondefault value.

```
!
interface Ethernet0/0
 ip address 172.16.2.1 255.255.255.0
ip pim query-interval 100 msec
ip pim sparse-mode
```

Additional References

The following sections provide references related to the Multicast Subsecond Convergence feature.

Related Documents

Related Topic	Document Title
PIM-SM and SSM concepts and configuration examples	"Configuring Basic IP Multicast" module
PIM-SM optimization concepts and configuration examples	"Optimizing PIM Sparse Mode in a Large IP Multicast Deployment" module
Cisco IOS IP multicast commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	*Cisco IOS IP Multicast Command Reference*

Standards

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

MIBs

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

RFCs

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

Technical Assistance

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

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IP Multicast Command Reference* at http://www.cisco.com/en/US/docs/ios/ipmulti/command/reference/imc_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

- **debug ip mrouting**
- **debug ip pim**
- **ip multicast rpf backoff**
- **ip multicast rpf interval**
- **ip pim query-interval**
- **show ip pim interface**
- **show ip pim neighbor**
- **show ip rpf events**

Feature Information for Multicast Subsecond Convergence

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Multicast Subsecond Convergence

Feature Name	Releases	Feature Information
Multicast Subsecond Convergence	12.0(22)S 12.2(14)S 12.2(15)T	The Multicast Subsecond Convergence feature comprises a comprehensive set of features and protocol enhancements that provide for improved scalability and convergence in multicast-based services. This feature set provides for the ability to scale to larger services levels and to recover multicast forwarding after service failure in subsecond time frames. The following commands were introduced or modified: **debug ip mrouting**, **debug ip pim**, **ip multicast rpf backoff**, **ip multicast rpf interval**, **ip pim query-interval**, **show ip pim interface**, **show ip pim neighbor**, **show ip rpf events**.
Multicast Subsecond Convergence	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Glossary

convergence—Speed and ability of a group of internetworking devices running a specific routing protocol to agree on the topology of an internetwork after a change in that topology.

DR—designated router. OSPF router that generates link-state advertisements (LSAs) for a multiaccess network and has other special responsibilities in running Open Shortest Path First (OSPF). Each multiaccess OSPF network that has at least two attached routers has a designated router that is elected by the OSPF Hello protocol. The designated router enables a reduction in the number of adjacencies required on a multiaccess network, which in turn reduces the amount of routing protocol traffic and the size of the topological database.

Internet Group Management Protocol (IGMP)—Internet Group Management Protocol. Used by IP hosts to report their multicast group memberships to an adjacent multicast router.

MBONE—multicast backbone. Multicast backbone of the Internet. MBONE is a virtual multicast network composed of multicast LANs and the point-to-point tunnels that interconnect them.

multicast—Single packets copied by the network and sent to a specific subset of network addresses. These addresses are specified in the Destination Address Field.

multicast address—Single address that refers to multiple network devices. Synonymous with group address.

Multicast Source Discovery Protocol (MSDP)—A mechanism to connect multiple Protocol Independent Multicast sparse mode (PIM-SM) domains. MSDP allows multicast sources for a group to be known to all rendezvous points in different domains.

PIM—Protocol Independent Multicast. Multicast routing architecture that allows the addition of IP multicast routing on existing IP networks. PIM is unicast routing protocol independent and can be operated in two modes: dense and sparse.

Reverse Path Forwarding (RPF)—Multicasting technique in which a multicast datagram is forwarded out of all but the receiving interface if the receiving interface is the one used to forward unicast datagrams to the source of the multicast datagram.

Note Refer to the *Internetworking Terms and Acronyms* for terms not included in this glossary.

PIM Dense Mode State Refresh

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

This feature module describes the Protocol Independent Multicast (PIM) Dense Mode (DM) State Refresh feature, which is an extension to the dense operational mode of the PIM Version 2 multicast routing architecture. This feature module includes the following sections:

Feature Overview

PIM dense mode builds source-based multicast distribution trees that operate on a flood and prune principle. Multicast packets from a source are flooded to all areas of a PIM dense mode network. PIM routers that receive multicast packets and have no directly connected multicast group members or PIM neighbors send a prune message back up the source-based distribution tree toward the source of the packets. As a result, subsequent multicast packets are not flooded to pruned branches of the distribution tree. However, the pruned state in PIM dense mode times out approximately every 3 minutes and the entire PIM dense mode network is reflooded with multicast packets and prune messages. This reflooding of unwanted traffic throughout the PIM dense mode network consumes network bandwidth.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

The PIM Dense Mode State Refresh feature keeps the pruned state in PIM dense mode from timing out by periodically forwarding a control message down the source-based distribution tree. The control message refreshes the prune state on the outgoing interfaces of each router in the distribution tree.

Benefits

The PIM Dense Mode State Refresh feature keeps the pruned state in PIM dense mode from timing out, which saves network bandwidth by greatly reducing the reflooding of unwanted multicast traffic to pruned branches of the PIM dense mode network. This feature also enables PIM routers in a PIM dense mode multicast network to recognize topology changes (sources joining or leaving a multicast group) before the default 3-minute state refresh timeout period.

Restrictions

All routers in a PIM dense mode network must run a Cisco IOS software release, such as Cisco IOS Release 12.1(5)T, that supports the PIM Dense Mode State Refresh feature to process and forward state refresh control messages.

The origination interval for the state refresh control message must be the same for all PIM routers on the same LAN. Specifically, the same origination interval must be configured on each router interface that is directly connected to the LAN.

Related Features and Technologies

The PIM Dense Mode State Refresh feature is an extension of the PIM Version 2 multicast routing architecture, which is documented in the Release 12.1 *Cisco IOS IP and IP Routing Configuration Guide* and the *Cisco IOS IP and IP Routing Command Reference*.

Supported Platforms

The PIM Dense Mode State Refresh feature is supported on the following platforms:

- Cisco 800 series
- Cisco 1400 series
- Cisco 1600 series
- Cisco 1700 series
- Cisco 2500 series
- Cisco 2600 series
- Cisco 3600 series
- Cisco 4000 series (Cisco 4000, 4000-M, 4500, 4500-M, 4700, 4700-M)
- Cisco 7000 series
- Cisco 7200 series
- Cisco 7500 series
- Cisco AS5800 access server

- Cisco AS5400 series
- Cisco AS5300 series
- Cisco AS5200 series
- Cisco MC3810 series
- Cisco MGX8850 WAN edge switch
- Cisco uBR7200 series

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

No new or modified MIBs are supported by this feature.

To obtain lists of MIBs supported by platform and Cisco IOS releases, and to download MIB modules, go to the Cisco MIB web site on Cisco Connection Online (CCO) at http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml.

RFCs

No new or modified RFCs are supported by this feature.

Prerequisites

You must have PIM dense mode enabled on an interface before configuring the PIM Dense Mode State Refresh feature.

Configuration Tasks

There are no configuration tasks for enabling the PIM Dense Mode State Refresh feature. By default, all PIM routers that are running a Cisco IOS software release that supports the PIM Dense Mode State Refresh feature automatically process and forward state refresh control messages. To disable the processing and forwarding of state refresh control messages on a PIM router, use the **ip pim state-refresh disable** global configuration command.

The origination of state refresh control messages is disabled by default. To configure the origination of the control messages on a PIM router, use the following commands beginning in global configuration mode:

Command	Purpose
`Router(config)# interface type number`	Specifies an interface and places the router in interface configuration mode.
`Router(config-if)# ip pim state-refresh origination-interval [interval]`	Configures the origination of the PIM Dense Mode State Refresh control message. Optionally, you can configure the number of seconds between control messages by using the *interval* argument. The default interval is 60 seconds. The interval range is 4 seconds to 100 seconds.

Verifying PIM Dense Mode State Refresh Configuration

Use the **show ip pim interface** [*type number*] **detail** and the **show ip pim neighbor** [*interface*] commands to verify that the PIM Dense Mode State Refresh feature is configured correctly. The following output of the **show ip pim interface** [*type number*] **detail** command indicates that processing, forwarding, and origination of state refresh control messages is enabled.

```
Router# show ip pim interface fastethernet 0/1 detail

FastEthernet0/1 is up, line protocol is up
  Internet address is 172.16.8.1/24
  Multicast switching:process
  Multicast packets in/out:0/0
  Multicast boundary:not set
  Multicast TTL threshold:0
  PIM:enabled
    PIM version:2, mode:dense
    PIM DR:172.16.8.1 (this system)
    PIM neighbor count:0
    PIM Hello/Query interval:30 seconds
→   PIM State-Refresh processing:enabled
→   PIM State-Refresh origination:enabled, interval:60 seconds
    PIM NBMA mode:disabled
    PIM ATM multipoint signalling:disabled
    PIM domain border:disabled
  Multicast Tagswitching:disabled
```

The S in the Mode field of the following **show ip pim neighbor** [*interface*] command output indicates that the neighbor has the PIM Dense Mode State Refresh feature configured.

```
Router# show ip pim neighbor

PIM Neighbor Table
Neighbor          Interface          Uptime/Expires    Ver   DR
Address                                                    Priority/Mode
→ 172.16.5.1        Ethernet1/1        00:09:03/00:01:41 v2    1 / B S
```

Monitoring and Maintaining PIM DM State Refresh

Following are the PIM Dense Mode State Refresh control messages that are sent and received by a PIM router after the **debug ip pim** privileged EXEC command is configured for multicast group 239.0.0.1:

```
Router# debug ip pim 239.0.0.1
```

```
*Mar  1 00:25:10.416:PIM:Originating refresh message for
(172.16.8.3,239.0.0.1)
*Mar  1 00:25:10.416:PIM:Send SR on Ethernet1/1 for (172.16.8.3,239.0.0.1)
TTL=9
```

The following output from the **show ip mroute** command displays are the resulting prune timer changes for interface Ethernet 1/0 and multicast group 239.0.0.1. (The following output assumes that the **debug ip pim** privileged EXEC command has already been configured on the router.) In the first output from the **show ip mroute** command, the prune timer reads 00:02:06. The debug messages indicate that a PIM Dense Mode State Refresh control message is received and sent on Ethernet interface 1/0, and that other PIM Dense Mode State Refresh routers were discovered. In the second output from the **show ip mroute** command, the prune timer has been reset to 00:02:55.

```
Router# show ip mroute 239.0.0.1

(172.16.8.3, 239.0.0.1), 00:09:50/00:02:06, flags:PT
  Incoming interface:Ethernet1/1, RPF nbr 172.16.5.2
  Outgoing interface list:
→   Ethernet1/0, Prune/Dense, 00:09:43/00:02:06

Router#
*Mar  1 00:32:06.657:PIM:SR on iif from 172.16.5.2 orig 172.16.8.1 for
(172.16.8.3,239.0.0.1)
*Mar  1 00:32:06.661:     flags:prune-indicator
*Mar  1 00:32:06.661:PIM:Cached metric is [0/0]
*Mar  1 00:32:06.661:PIM:Keep RPF nbr 172.16.5.2
*Mar  1 00:32:06.661:PIM:Send SR on Ethernet1/0 for (172.16.8.3,239.0.0.1)
TTL=8
*Mar  1 00:32:06.661:     flags:prune-indicator

Router# show ip mroute 239.0.0.1

(172.16.8.3, 239.0.0.1), 00:10:01/00:02:55, flags:PT
  Incoming interface:Ethernet1/1, RPF nbr 172.16.5.2
  Outgoing interface list:
→   Ethernet1/0, Prune/Dense, 00:09:55/00:02:55
```

Configuration Examples

The following example is for a PIM router that is originating, processing, and forwarding PIM Dense Mode State Refresh control messages on Fast Ethernet interface 0/1 every 60 seconds:

```
ip multicast-routing

interface FastEthernet0/1
 ip address 172.16.8.1 255.255.255.0
 ip pim state-refresh origination-interval 60
 ip pim dense-mode
```

The following example is for a PIM router that is just processing and forwarding PIM Dense Mode State Refresh control messages on Fast Ethernet interface 1/1:

```
ip multicast-routing

interface FastEthernet1/1
 ip address 172.16.7.3 255.255.255.0
 ip pim dense-mode
```

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IP Multicast Command Reference* at http://www.cisco.com/en/US/docs/ios/ipmulti/command/reference/imc_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

- **ip pim state-refresh disable**
- **ip pim state-refresh origination-interval**
- **show ip pim interface**

Multicast VPN

Configuring Multicast VPN

First Published: May 2, 2005
Last Updated: August 21, 2007

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

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

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

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Configuring Multicast VPN" section on page 21.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Configuring Multicast VPN

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

Restrictions for Configuring Multicast VPN

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

Information About Configuring Multicast VPN

Before you configure MVPN, you should understand the following concepts:

Multicast VPN Operation

MVPN IP allows a service provider to configure and support multicast traffic in a Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) environment. This feature 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.

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

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

Multicast VPN Routing and Forwarding and Multicast Domains

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

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

Multicast Distribution Trees

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

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

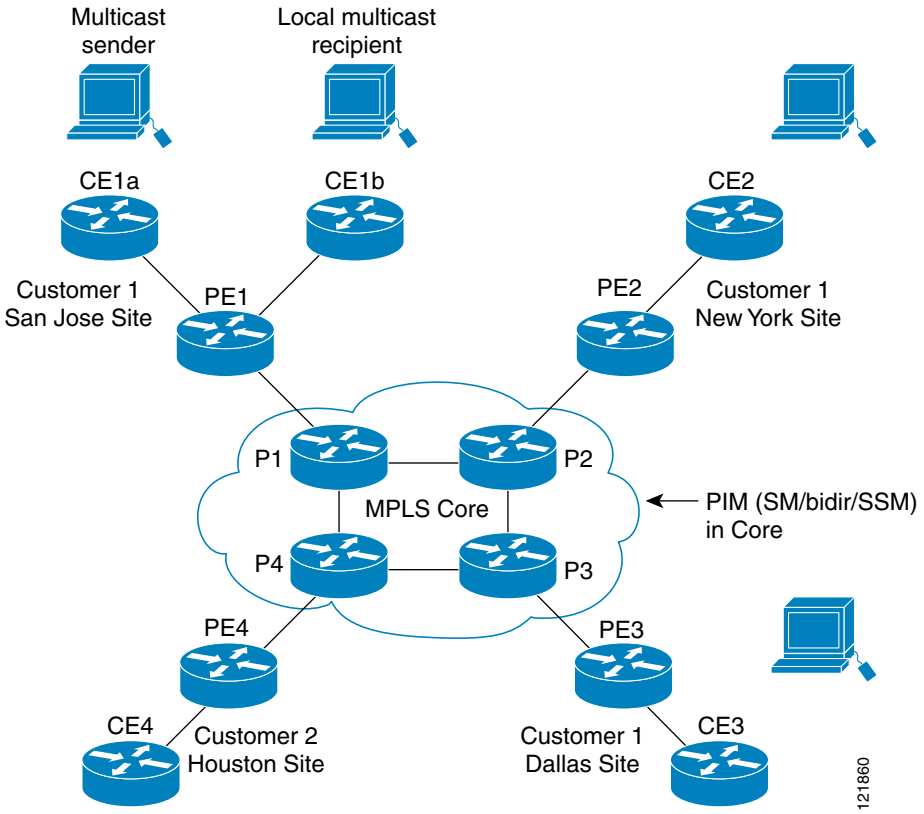
MVPN also supports the dynamic creation of MDTs for high-bandwidth transmission. Data MDTs are a feature unique to Cisco IOS software. Data MDTs are intended for high-bandwidth sources such as full-motion video inside the VPN to ensure optimal traffic forwarding in the MPLS VPN core. The threshold at which the data MDT is created can be configured on a per-router or a per-VRF basis. When the multicast transmission exceeds the defined threshold, the sending PE router creates the data MDT and sends a User Datagram Protocol (UDP) message, which contains information about the data MDT to all routers on the default MDT. The statistics to determine whether a multicast stream has exceeded the data MDT threshold are examined once every second; on distributed platforms such as the Cisco 7500 and Cisco 12000, those statistics are updated to the route processor every 10 seconds. After a PE router sends the UDP message, it waits 3 more seconds before switching over; 13 seconds is the worst case switchover time; 3 seconds is the best case.

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

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

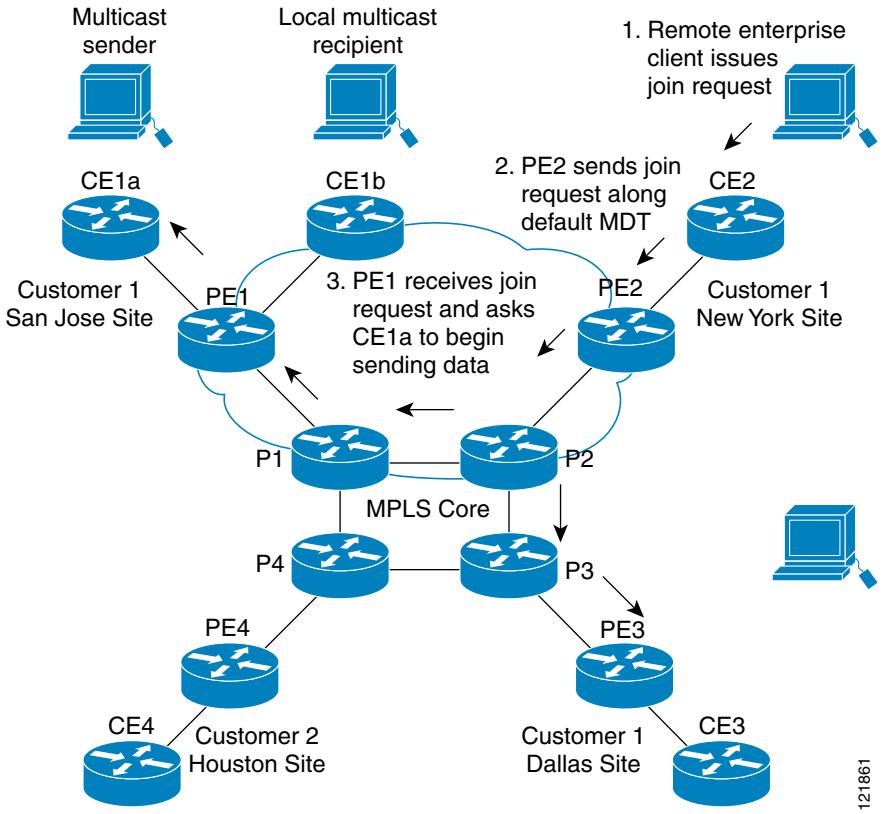
The default MDT for the enterprise customer consists of provider routers P1, P2, and P3 and their associated PE routers. PE4 is not part of the default MDT, because it is associated with a different customer. Figure 1 shows that no data flows along the default MDT, because no one outside of San Jose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

Figure 1 Default Multicast Distribution Tree Overview



An employee in New York joins the multicast session. The PE router associated with the New York site sends a join request that flows across the default MDT for the multicast domain of the customer. PE1, the PE router associated with the multicast session source, receives the request. Figure 2 depicts that the PE router forwards the request to the CE router associated with the multicast source (CE1a).

Figure 2 Initializing the Data MDT



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

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

Multicast Tunnel Interface

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

Multicast Distributed Switching Support for Multicast VPN

Multicast distributed switching (MDS) is supported for MVPN on the Cisco 7500 series routers. When MDS is configured, ensure that all interfaces enabled for IP multicast have MDS enabled correctly—verify that no interface has the **no ip mroute-cache** command configured (including loopback interfaces).

Benefits of Multicast VPN

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

How to Configure Multicast VPN

This section contains the following procedures.

Configuring a Default MDT Group for a VRF

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast-routing** **Example:** `Router(config)# ip multicast-routing`	Enables multicast routing.
Step 4	**ip multicast-routing vrf** *vrf-name* **Example:** `Router(config)# ip multicast-routing vrf vrf1`	Supports the Multicast Virtual Private Network (VPN) routing and forwarding (VRF) instance.
Step 5	**ip vrf** *vrf-name* **Example:** `Router(config)# ip vrf vrf1`	Enters VRF configuration mode and defines the VPN routing instance by assigning a VRF name.
Step 6	**mdt default** *group-address* **Example:** `Router(config-vrf)# mdt default 232.0.0.1`	Configures the multicast group address range for data multicast distribution tree (MDT) groups for a VRF. A tunnel interface is created as a result of this command. By default, the destination address of the tunnel header is the *group-address* argument.

Configuring the MDT Address Family in BGP for Multicast VPN

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

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

BGP Advertisement Methods for Multicast VPN Support

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

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

Table 1 lists the BGP advertisement methods for sending the source PE address and the default MDT group that are available (by Cisco IOS release).

Table 1 BGP Advertisement Methods for MVPN

Cisco IOS Release	BGP Advertisement Method
• Release 12.0(29)S • Release 12.2(33)SRA1 • Release 12.2(31)SB2 • Release 12.2(33)SXH	Extended Communities
• Release 12.0(29)S and later 12.0S releases • Release 12.2(31)SB2 and later 12.2SB releases • Release 12.2(33)SRA and later 12.2SR releases • Release 12.2(33)SXH and later 12.2SX releases	BGP address family MDT SAFI

BGP Extended Community

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

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

BGP MDT SAFI

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

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

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

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

Auto-Migration to the MDT SAFI

In Cisco IOS Release 12.0(30)S3, auto-migration to the MDT SAFI functionality was introduced to ease the migration to the MDT SAFI. This functionality was integrated into Cisco IOS Releases 12.2(33)SRA1, 12.2(31)SB2, and 12.2(33)SXH. When migrating a Cisco IOS release to the MDT SAFI, existing VPNv4 neighbors will be automatically configured for the MDT SAFI upon bootup neighbors based on the presence of an existing default MDT configuration (that is, pre-MDT SAFI configurations will be automatically converted to an MDT SAFI configuration upon bootup). In addition, when a default MDT configuration exists and a VPNv4 neighbor in BGP is configured, a similar neighbor in the IPv4 MDT address family will be automatically configured.

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

Guidelines for Configuring the MDT SAFI

- We recommended that you configure the MDT SAFI on all routers that participate in the MVPN. Even though the benefits of the MDT SAFI are for SSM tree building, the MDT SAFI must also be configured when using MVPN with the default MDT group for PIM-SM. From the multicast point of view, the MDT SAFI is not required for MVPN to work within a PIM-SM core. However, in certain scenarios, the new address family must be configured in order to create the MTI. Without this notification, the MTI would not be created and MVPN would not function (even with PIM-SM).
- For backward compatible sessions, extended communities must be enabled on all MDT SAFI peers. In a pure MDT SAFI environment there is no need to configure extended communities explicitly for MVPN. However, extended communities will be needed for VPNv4 interior BGP (iBGP) sessions to relay the route-target. In a hybrid (MDT SAFI and pre-MDT SAFI) environment, extended communities must be configured to send the embedded source in the VPNv4 address and the MDT group address to MDT SAFI neighbors.

Guidelines for Upgrading a Network to Support the MDT SAFI

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

1. Upgrade the PEs in the MVPN to a Cisco IOS release that supports the MDT SAFI. Upon bootup, the PE configurations will be auto-migrated to the MDT SAFI. For more information about the auto-migration to the MDT SAFI functionality, see the "Auto-Migration to the MDT SAFI" section.
2. After the PEs have been upgraded, upgrade the RRs and enable the MDT SAFI for all peers providing MVPN service. Enabling or disabling the MDT SAFI will reset the BGP peer relationship for all address families; thus, a loss of routing information may occur.

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

Supported Policy

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

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

Prerequisites

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

Restrictions

The following policy configuration parameters are not supported:

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 65535`	Enters router configuration mode and creates a BGP routing process.
Step 4	**address-family ipv4 mdt** **Example:** `Router(config-router)# address-family ipv4 mdt`	Enters address family configuration to create an IP MDT address family session.
Step 5	**neighbor** *neighbor-address* **activate** **Example:** `Router(config-router-af)# neighbor 192.168.1.1 activate`	Enables the MDT address family for this neighbor.
Step 6	**neighbor** *neighbor-address* **send-community** [**both** | **extended** | **standard**] **Example:** `Router(config-router-af)# neighbor 192.168.1.1 send-community extended`	Enables community and (or) extended community exchange with the specified neighbor.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Router(config-router-af)# exit`	Exits address family configuration mode and returns to router configuration mode.
Step 8	**address-family vpnv4** **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode to create a VPNv4 address family session.
Step 9	**neighbor** *neighbor-address* **activate** **Example:** `Router(config-router-af)# neighbor 192.168.1.1 activate`	Enables the VPNv4 address family for this neighbor.
Step 10	**neighbor** *neighbor-address* **send-community** [**both** | **extended** | **standard**] **Example:** `Router(config-router-af)# neighbor 192.168.1.1 send-community extended`	Enables community and (or) extended community exchange with the specified neighbor.
Step 11	**end** **Example:** `Router(config-router-af)# end`	Exits address-family configuration mode and enters privileged EXEC mode.

Configuring the Data Multicast Group

Perform this task to configure a data MDT group.

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

Prerequisites

- Before configuring a default MDT group, the VPN must be configured for multicast routing as described in the "Configuring a Default MDT Group for a VRF" section on page 6.
- All access lists needed when using the tasks in this module should be configured prior to beginning the configuration task. For information about how to configure an access list, see the "Creating an IP Access List and Applying It to an Interface" module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **mdt data** *group-address-range wildcard-bits* [**threshold** *threshold-value*] [**list** *access-list*]

5. **mdt log-reuse**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Router(config)# ip vrf vrf1`	Enters VRF configuration mode and defines the VPN routing instance by assigning a VRF name.
Step 4	**mdt data** *group-address-range wildcard-bits* [**threshold** *threshold-value*] [**list** *access-list*] **Example:** `Router(config-vrf)# mdt data 232.0.1.0 0.0.0.255 threshold 500 list 101`	Configures the multicast group address range for data MDT groups. • This command configures a range of alternative multicast destination addresses for the tunnel header. The destination address chosen depends on the traffic profile (that is, the source and destination match the specified access list and the rate of the traffic has exceeded the bandwidth threshold value). • The threshold is in kbps.
Step 5	**mdt log-reuse** **Example:** `Router(config-vrf)# mdt log-reuse`	(Optional) Enables the recording of data multicast distribution tree (MDT) reuse and generates a syslog message when a data MDT has been reused.
Step 6	**exit** **Example:** `Router(config-vrf)# exit`	Returns to global configuration mode.

Configuring Multicast Routes and Information

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

Prerequisites

- Before configuring a default MDT group, the VPN must be configured for multicast routing as described in the "Configuring a Default MDT Group for a VRF" section on page 6.
- All access lists needed when using the tasks in this module should be configured prior to beginning the configuration task. For information about how to configure an access list, see the "Creating an IP Access List and Applying It to an Interface" module.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast vrf** *vrf-name* **route-limit** *limit* [*threshold*] **Example:** `Router(config)# ip multicast vrf cisco route-limit 200000 20000`	Sets the mroute limit and the threshold parameters.
Step 4	**ip multicast mrinfo-filter** *access-list* **Example:** `Router(config)# ip multicast mrinfo-filter 4`	Filters the multicast router information request packets for all sources specified in the access list.

Verifying Information for the MDT Default Group

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

SUMMARY STEPS

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

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

- Enter your password if prompted.

Example:

```
Router> enable
```

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

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

```
Router# show ip msdp peer 224.135.250.116

MSDP Peer 224.135.250.116 (rtp5-rp1.cisco.com), AS 109 (configured AS)
Description:
 Connection status:
   State: Up, Resets: 9, Connection source: Loopback2 (228.69.199.17)
   Uptime(Downtime): 1d10h, Messages sent/received: 436765/429062
   Output messages discarded: 0
   Connection and counters cleared 1w2d     ago
 SA Filtering:
   Input (S,G) filter: none, route-map: none
   Input RP filter: none, route-map: none
   Output (S,G) filter: none, route-map: none
   Output RP filter: none, route-map: none
 SA-Requests:
   Input filter: none
   Sending SA-Requests to peer: disabled
 Peer ttl threshold: 0
 SAs learned from this peer: 32, SAs limit: 500
 Input queue size: 0, Output queue size: 0
```

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

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

```
Router# show ip msdp summary

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

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

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

```
Router# show ip pim mdt bgp

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

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

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

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

```
Router# show ip pim mdt send

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

Step 6 show ip pim mdt history

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

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

   MDT-data send history for VRF - vrf1 for the past 20 minutes


MDT-data group        Number of reuse
     10.9.9.8           3
     10.9.9.9           2
```

Configuration Examples for Multicast VPN

This section contains the following configuration examples:

Configuring MVPN and SSM: Example

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

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

Enabling a VPN for Multicast Routing: Example

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

```
ip multicast-routing vrf1
```

Configuring the MDT Address Family in BGP for Multicast VPN: Example

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

```
!
ip vrf test
 rd 55:2222
 route-target export 55:2222
 route-target import 55:2222
 mdt default 232.0.0.1
!
ip multicast-routing
ip multicast-routing vrf test
!
router bgp 55
.
.
```

```
.
!
 address-family vpnv4
 neighbor 192.168.1.1 activate
 neighbor 192.168.1.1 send-community-both
!
 address-family ipv4 mdt
 neighbor 192.168.1.1 activate
 neighbor 192.168.1.1 send-community-both
!
```

Configuring the Multicast Group Address Range for Data MDT Groups: Example

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

```
ip vrf blue
 rd 55:1111
 route-target both 55:1111
 mdt default 239.1.1.1
 mdt data 239.1.2.0 0.0.0.3
 end
show ip vrf blue
  Name                             Default RD          Interfaces
  blue                             55:1111
```

Configuring the IP Source Address of Register Messages: Example

In the following example, the IP source address of the register message is configured to the Ethernet interface 1 of a DR:

```
ip pim register-source Ethernet1/0/1
```

Configuring an MSDP Peer: Example

In the following example, an MSDP peer is configured with a VPN routing instance named vrf1 and a source of 10.10.0.1 from Ethernet interface 1:

```
ip msdp vrf vrf1 peer 10.10.0.1 connect-source E1/0/1
```

Limiting the Number of Multicast Routes: Example

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

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

Additional References

The following sections provide references related to the configuring Multicast VPN.

Related Documents

Related Topic	Document Title
Extranet MVPN concepts, tasks, and configuration examples	"Configuring Multicast VPN Extranet Support" module
Inter-AS MVPN concepts, tasks, and configuration examples	"Configuring Multicast VPN Inter-AS Support" module
MVPN MIB concepts and tasks	"Multicast VPN MIB" module
IP multicast commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Multicast Command Reference,* Release 12.4

Standards

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

MIBs

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

RFCs

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

Technical Assistance

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

Feature Information for Configuring Multicast VPN

Table 2 lists the release history for this feature.

Table 2 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Releases 12.2(1) or 12.0(3)S or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the *IP Multicast Features Roadmap*.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 2 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 2 Feature Information for Configuring Multicast VPN

Feature Name	Releases	Feature Information
Multicast VPN—IP Multicast Support of MPLS VPNs	12.0(23)S 12.2(13)T 12.2(14)S 12.0(25)S1 12.0(26)S 12.0(32)SY	The Multicast VPN feature provides the ability to support multicast over a Layer 3 Virtual Private Network (VPN). As enterprises extend the reach of their multicast applications, service providers can accommodate these enterprises over their Multiprotocol Label Switching (MPLS) core network. IP multicast is used to stream video, voice, and data to an MPLS VPN network core. This entire module provides information about this feature.

Multicast VPN MIB

First Published: August 9, 2004
Last Updated: August 21, 2007

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

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Multicast VPN MIB" section on page 7.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Prerequisites for Multicast VPN MIB

- Before performing the tasks in this module, you must configure MVPN. For information, see the "Configuring Multicast-VPN" chapter in the *Cisco IOS Multicast Configuration Guide*.
- You must configure SNMP on the routers on which the MVPN MIB is to be used. See the "Configuring the Router to Send MVRF Trap Notifications" task for more information. For more information about configuring an SNMP server, see the "Configuring SNMP Support" chapter in the *Cisco IOS Network Management Configuration Guide*.

Restrictions for Multicast VPN MIB

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

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Information About Multicast VPN MIB

To configure the Multicast VPN MIB feature, you should understand the following concepts:

Overview of the MVPN MIB

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

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

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

MVPN Information Retrieval Using SNMP and the MVPN MIB

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

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

MVPN MIB Objects

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

- The state of the MVRFs including the name of the MVRF, whether they are active, and the number of active multicast-enabled interfaces

- MDT default group address and encapsulation information
- Next hop information used to receive Border Gateway Protocol (BGP) MDT updates for Source Specific Multicast (SSM) mode
- Traffic threshold that determines switchover to an MDT data group
- Type of MDT group being used for a given (S, G) multicast route entry that exists on each configured MVRF, source address, and group address of the multicast route entry
- Source and group address used for encapsulation
- Information on MDT data groups currently joined
- Information on MVPN-specific MDT tunnels present in the device
- Trap notifications enabled on the router

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

How to Configure Multicast VPN MIB

This section contains the following required procedure:

- Configuring the Router to Send MVRF Trap Notifications, page 3 (required)

Configuring the Router to Send MVRF Trap Notifications

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

MVRF Trap Notifications

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

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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server community** *string* **ro**
   or
   snmp-server community *string* **rw**

4. **snmp-server host** {*hostname* | *ip-address*} **version 2c** *community-string*
5. **snmp-server enable traps mvpn**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server community** *string* **ro** or **snmp-server community** *string* **rw** **Example:** `Router(config)# snmp-server community public ro` or `Router(config)# snmp-server community public rw`	Sets up the community access string to permit access to SNMP. • The *string* argument is a community string that consists of from 1 to 32 alphanumeric characters and functions much like a password, permitting access to the SNMP protocol. Blank spaces are not permitted in the community string. • Specifying the **snmp-server community** command with the **ro** keyword configures read-only access. SNMP management stations using this string only can retrieve MIB objects. or • Specifying the **snmp-server community** command with the **rw** keyword configures read-write access. SNMP management stations using this string can retrieve and modify MIB objects.
Step 4	**snmp-server host** {*hostname*	*ip-address*} **version 2c** *community-string* **Example:** `Router(config)# snmp-server host 192.168.1.1 version 2c public`	Specifies the recipient of an SNMP notification operation.
Step 5	**snmp-server enable traps mvpn** **Example:** `Router(config)# snmp-server enable traps mvpn`	Enables the router to send MVRF trap notifications.
Step 6	**end** **Example:** `Router(config)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Configuration Examples for Multicast VPN MIB

This section provides the following configuration example:

Configuring the Router to Send MVRF Trap Notifications: Example

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

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

Additional References

The following sections provide references related to the Multicast VPN MIB feature.

Related Documents

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

Standards

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

MIBs

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

RFCs

RFC	Title
No new or modified RFCs are supported by this feature.	—

Technical Assistance

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

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IP Multicast Command Reference* at http://www.cisco.com/en/US/docs/ios/ipmulti/command/reference/imc_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*..

- **snmp-server enable traps mvpn**

Feature Information for Multicast VPN MIB

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Multicast VPN MIB

Feature Name	Releases	Feature Information
Multicast VPN MIB	12.0(29)S 12.3(14)T 12.2(33)SRA 12.2(33)SXH	The Multicast VPN MIB feature introduces the capability for SNMP monitoring of an MVPN using the MVPN MIB (CISCO-MVPN-MIB). The following command was introduced by this feature: **snmp server enable traps mvpn**

Configuring Multicast VPN Inter-AS Support

First Published: November 8, 2004
Last Updated: July 11, 2008

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

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Configuring Multicast VPN Inter-AS Support" section on page 71.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Configuring Multicast VPN Inter-AS Support

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

Restrictions for Configuring Multicast VPN Inter-AS Support

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

Information About Multicast VPN Inter-AS Support

To configure the Multicast VPN Inter-AS Support feature, you should understand the following concepts:

MVPN Inter-AS Support Overview

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

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

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

Benefits of MVPN Inter-AS Support

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

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

MVPN Inter-AS Support Implementation Requirements

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

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

    The Option A model assumes direct connectivity between PE routers of different autonomous systems. The PE routers are attached by multiple physical or logical interfaces, each of which is associated with a given VPN (through a VRF instance). Each PE router, therefore, treats the adjacent PE router like a customer edge (CE) router, and the standard Layer 3 MPLS VPN mechanisms are used for route redistribution with each autonomous system; that is, the PEs use exterior BGP (eBGP) to distribute unlabeled IPv4 addresses to each other.

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

  - Option B—VPNv4 route exchange between ASBRs

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

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

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

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

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

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

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

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

Limitations That Prevented Option B and Option C Support Prior to the Introduction of Multicast VPN Inter-AS Support

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

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

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

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

MVPN Inter-AS Support for Option A

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

- For Option A, native IP forwarding is used by the PE routers between autonomous systems; therefore, Option A does not require support for inter-AS MDTs.
- For Option A, the MDT is limited to one autonomous system; therefore, the issues associated with managing MDT group addresses between autonomous systems and RPF for inter-AS sources never applied to Option A.

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

MVPN Inter-AS Support Solution for Options B and C

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

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

BGP Connector Attribute

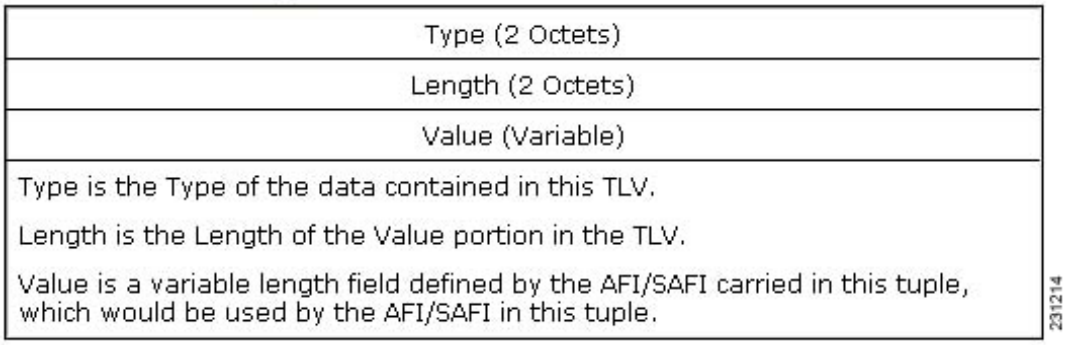
In an adjacent autonomous system, a PE router that wants to join a particular source of the default MDT for a given MVPN must know the originator's address of the source PE router. This presents some challenges for Option B inter-AS deployments because the originator next hop for VPNv4 routes is rewritten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

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

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

The format of the BGP connector attribute is shown in Figure 1.

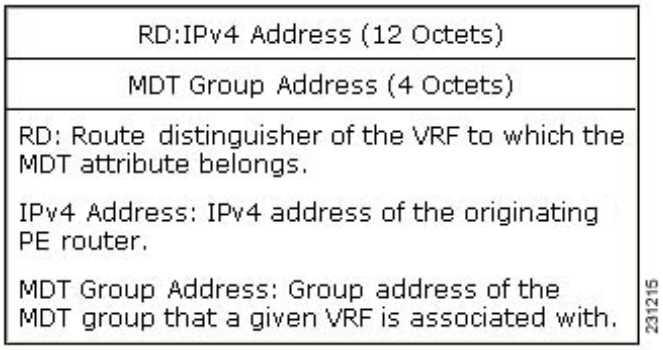
Figure 1 BGP Connector Attribute



BGP MDT SAFI Updates for MVPN Inter-AS Support

The BGP MDT SAFI is specifically designed to carry the address of the source PE router to which a PIM join should be sent for the MDT group contained in the PIM join. The format of the Network Layer Reachability Information (NLRI) carried in this SAFI is {RD:PE-IP-address}. The BGP MDT SAFI is capable of being advertised across autonomous system boundaries. Each MDT group is carried in the MP_REACH attribute using the format shown in Figure 2.

Figure 2 MDT SAFI Format



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

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

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

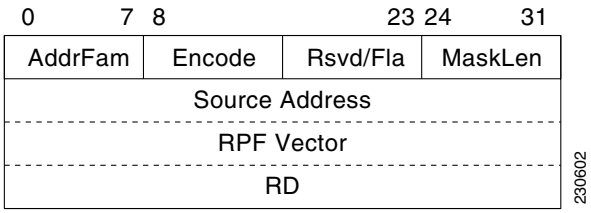
PIM RPF Vector

Normally, in an MVPN environment, PIM sends join messages containing the IP address of upstream PE routers that are sources of a given MDT group. To be able to perform RPF checks, however, P routers must have IPv4 reachability to source PE routers in remote autonomous systems. This behavior is not the case with inter-AS Options B and C because the autonomous systems do not exchange any of their IGP routes, including those of their local PE routers. However, P routers do have reachability to the BGP next hop of the BGP MDT update received with the BGP MDT SAFI updates at the PE routers. Therefore, if the PE routers add the remote PE router IP address (as received within the BGP MDT SAFI) and the BGP next-hop address of this address within the PIM join, the P routers can perform an RPF check on the BGP next-hop address rather than the original PE router address, which, in turn, allows the P router to forward the join toward the ASBR that injected the MDT SAFI updates for a remote autonomous system. This functionality is generally referred to as the *PIM RPF Vector*; the actual vector that is inserted into PIM joins is referred to as the *RPF Vector* or the *Proxy Vector*. The PIM RPF Vector, therefore, enables P routers to determine the exit ASBR to a source PE router in a remote autonomous system. Having received the join that contains a RPF Vector, an ASBR can then determine that the next-hop address is in fact itself and can perform an RPF check based on the originating PE router address carried in the PIM join.

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

The format of this PIM RPF Vector encoding in PIM join and prune messages is shown in Figure 3.

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



Note RPF Vectors can be used natively in an IP environment (that is, in a non-VPN environment). Use of RPF Vectors in a native environment is outside the scope of this module. For more information about the use of RPF Vectors in a native environment, see *The RPF Vector TLV* internet draft.

Originators of an RPF Vector

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

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

Note Using the RPF Vector in a native IP network environment is outside the scope of this module. For more information about the use of RPF Vectors in a native environment, see *The RPF Vector TLV* internet draft.

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

Recipients of an RPF Vector

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

ASBR Receipt of an RPF Vector

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

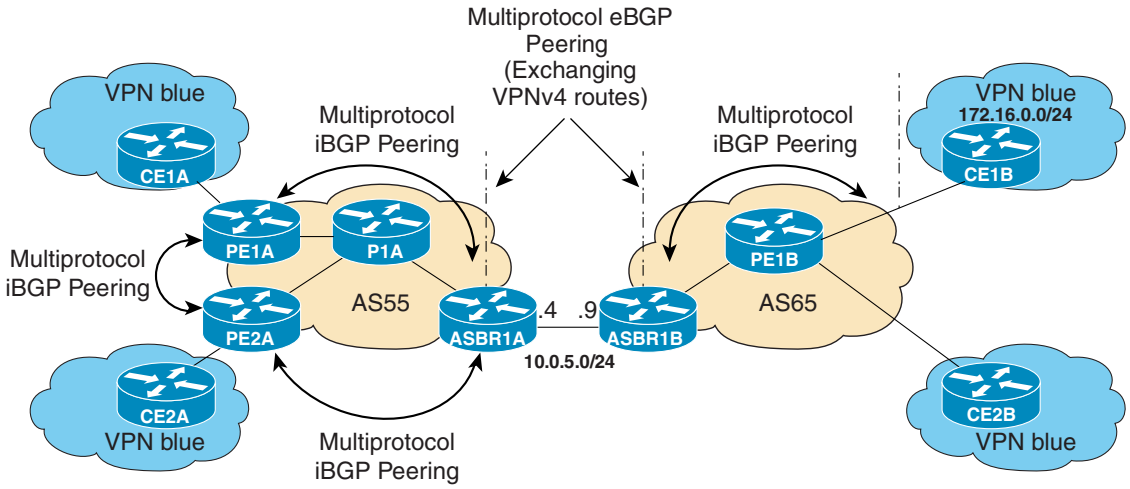
Interoperability with RPF Vector

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

MVPN Inter-AS MDT Establishment for Option B

This section describes the sequence of events that leads to the establishment of an inter-AS MDT between the autonomous systems in the sample inter-AS Option B topology illustrated in Figure 4. For this topology, assume that all the routers have been configured properly to support all extensions associated with the Multicast VPN Inter-AS Support feature.

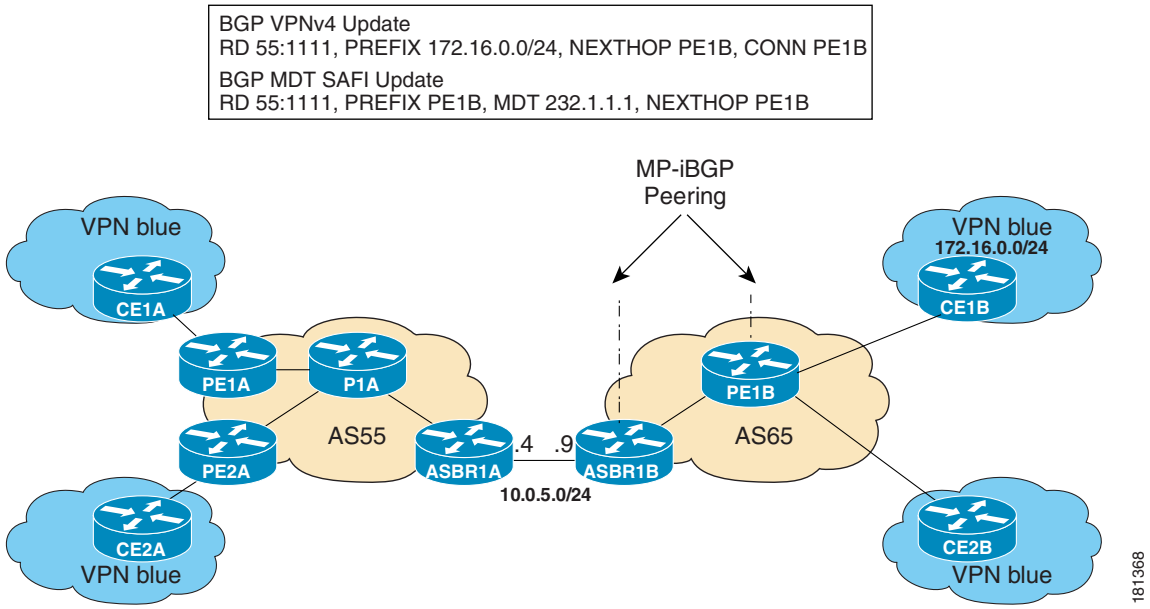
Figure 4 MVPN Inter-AS Support Option B Sample Topology



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

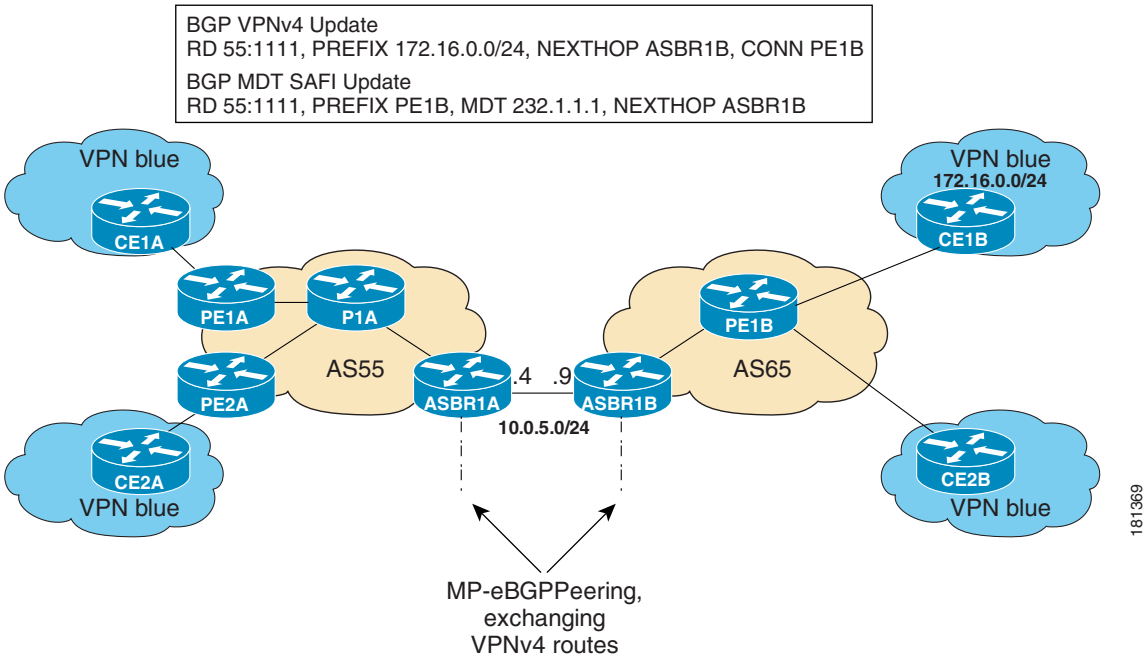
1. As illustrated in Figure 5, PE1B advertises the default MDT information for VPN blue using the BGP MDT SAFI with itself (PE1B) as the next hop.

Figure 5 BGP Updates from PE1B to ASBR1B



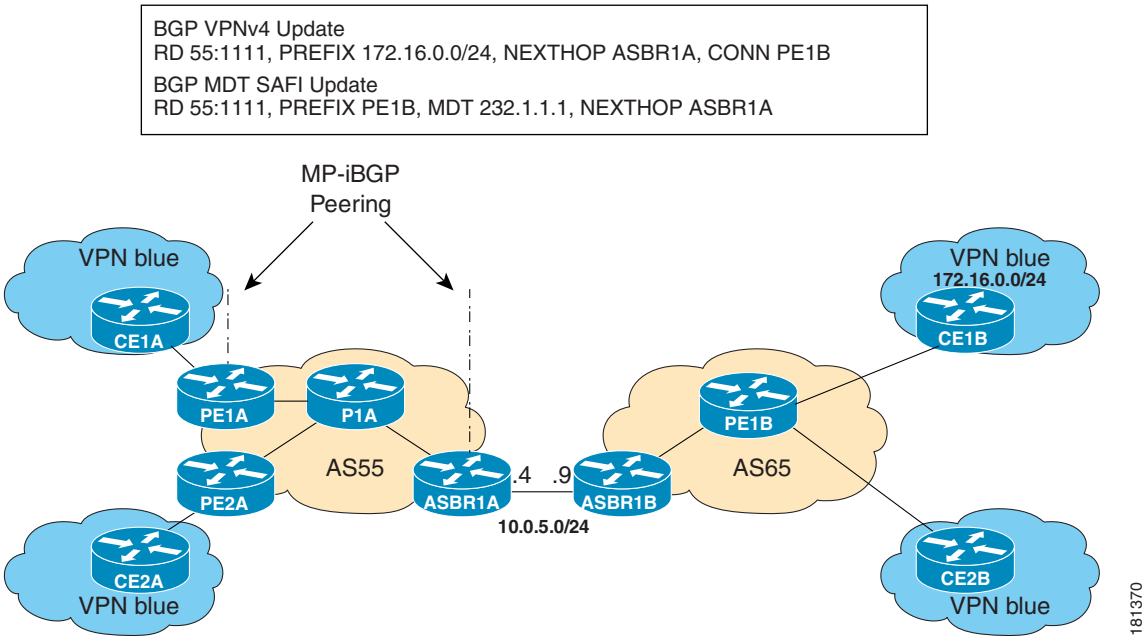
2. As illustrated in Figure 6, ASBR1B receives the MDT SAFI information and, in turn, advertises this information to ASBR1A with itself (ASBR1B) as the next hop.

Figure 6 BGP Updates from ASBR1B to ASBR1A



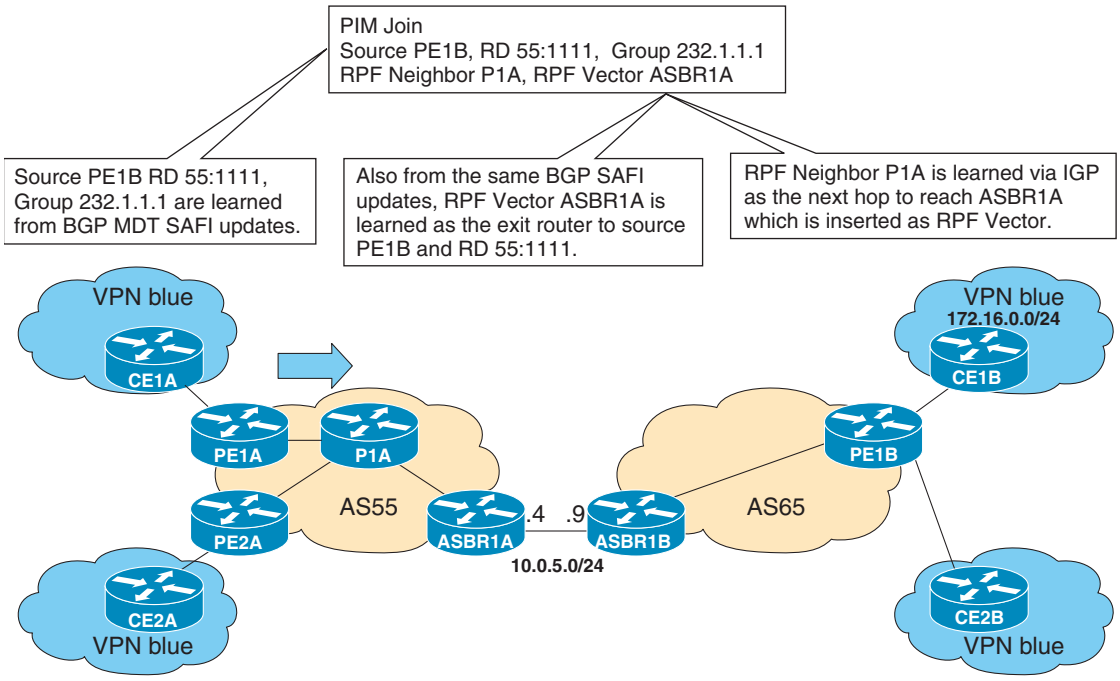
3. As illustrated in Figure 7, ASBR1A advertises the MDT SAFI to PE1A with itself (ASBR1A) as the next hop.

Figure 7 BGP Updates from ASBR1A to PE1A



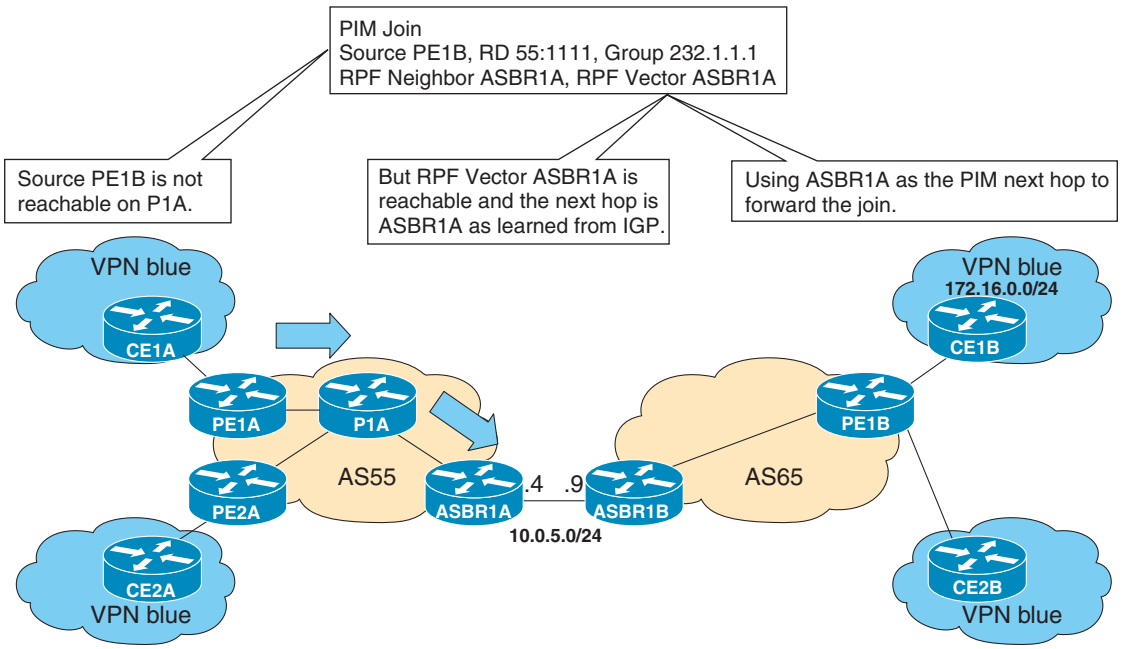
4. As illustrated in Figure 8, PE1A learns the source PE router, the RD, and the default MDT group address from BGP MDT SAFI updates. In addition, from the same BGP MDT SAFI updates, PE1A learns that the RPF Vector, ASBR1A, is the exit router to source PE1B RD 55:1111. PE1A learns that P1A is an RPF neighbor through an IGP. PE1A then inserts the RPF Vector into the PIM join and sends the PIM join that is destined for source PE1B to P1A.

Figure 8 SSM Default PIM Join from PE1A to P1A



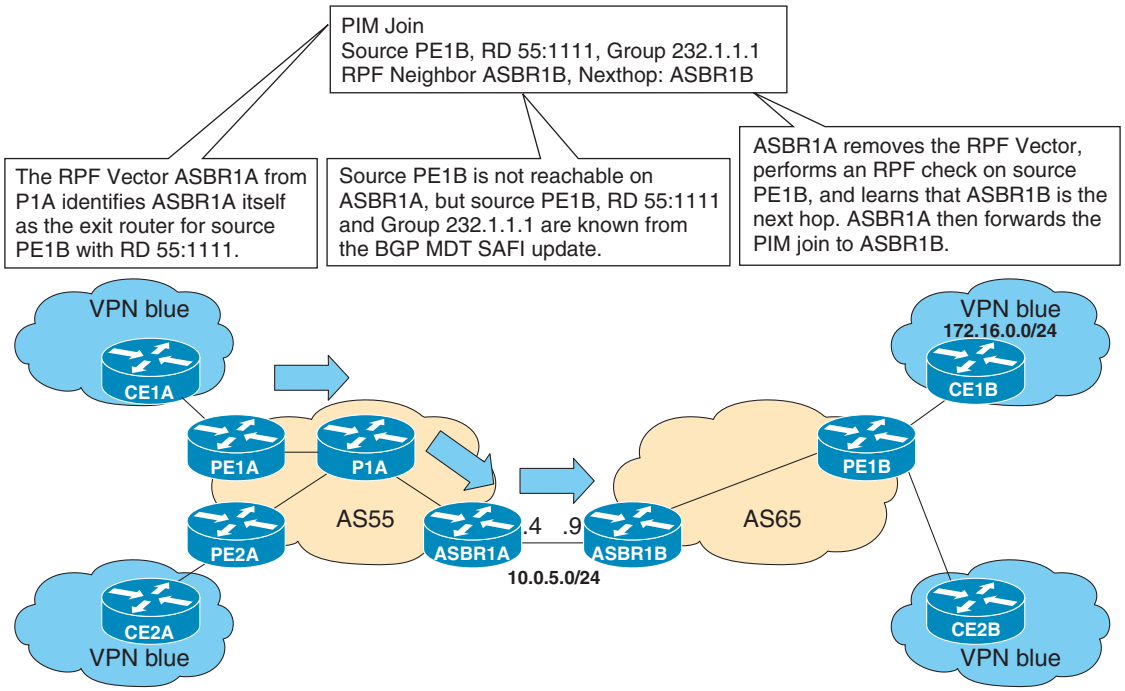
5. As illustrated in Figure 9, source PE1B is not reachable on P1A, but the RPF Vector ASBR1A is reachable, and the next hop is ASBR1A, as learned from the IGP running in the core. P1A then forwards the PIM join to ASBR1A.

Figure 9 SSM Default MDT PIM Join from P1A to ASBR1A



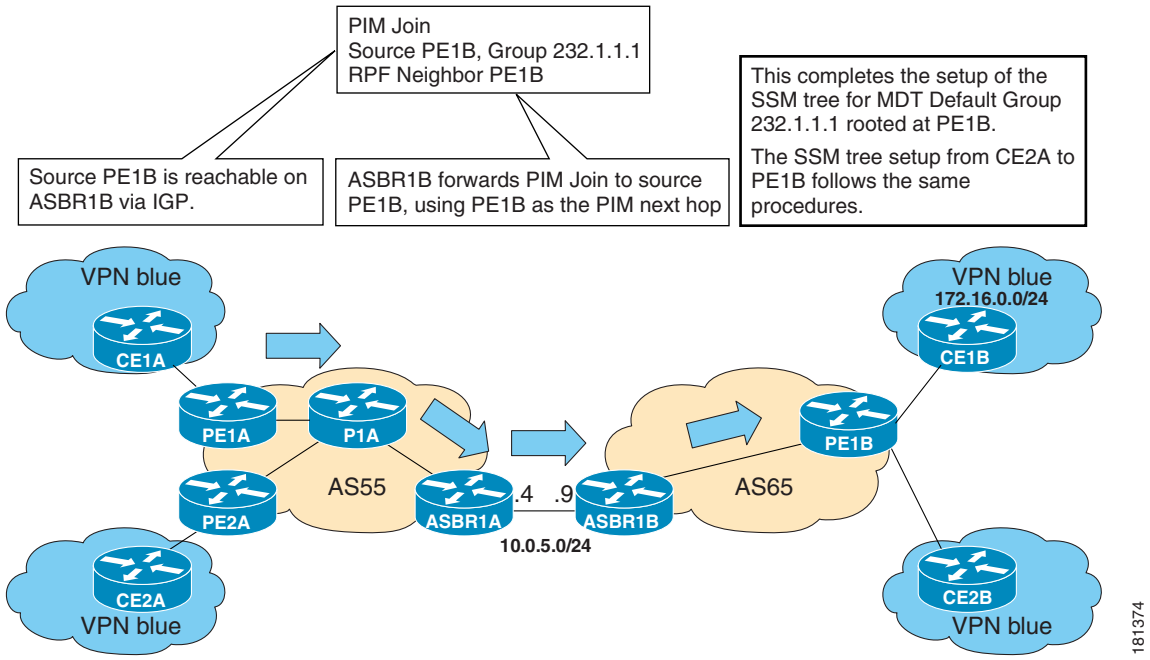
6. As illustrated in Figure 10, the RPF Vector, ASBR1A, is contained in the PIM join sent from P1A to ASBR1A. When ASBR1A receives the RPF Vector, it learns that it is the exit router for source PE1B with RD 55:1111. Source PE1B is not reachable on ASBR1A, but source PE1B, RD 55:1111, and group 232.1.1.1 are known from the BGP MDT SAFI updates. The RPF neighbor P1A is learned from the IGP running in the core as the next hop to reach ASBR1A, which is inserted as the RPF Vector. ASBR1A then forwards the PIM join for source PE1B to ASBR1B.

Figure 10 SSM Default MDT PIM Join from ASBR1A to ASBR1B



7. As illustrated in Figure 11, source PE1B is reachable on ASBR1B through the IGP running in AS65. ASBR1B forwards the PIM join to source PE1B, using PE1B as the next hop. At this point, the setup of the SSM tree for MDT default group 232.1.1.1 rooted at PE1B is complete. The SSM MDT default group rooted at PE1B, thus, has been established. The SSM trees for the MDT default groups rooted at PE1A and PE2A follow the same procedures.

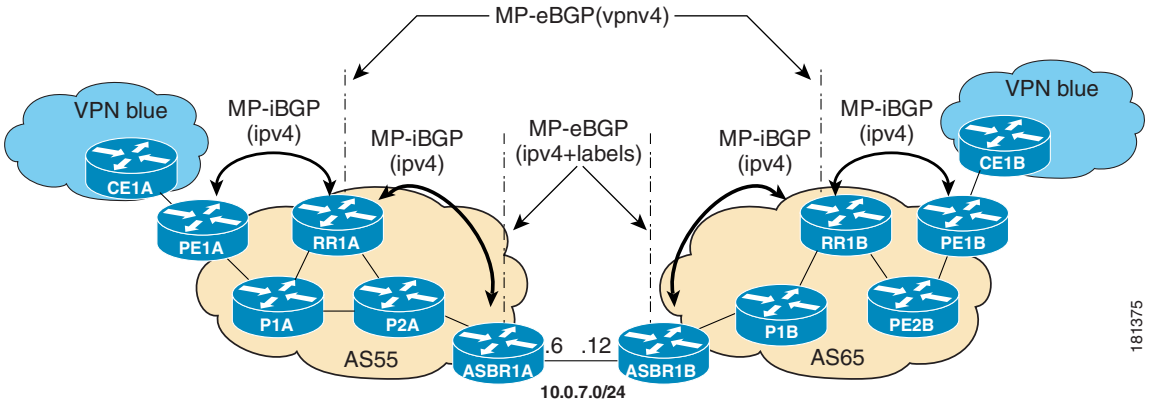
Figure 11 SSM Default MDT PIM Join from ASBR1B to PE1B



MVPN Inter-AS MDT Establishment for Option C

This section describes the sequence of events that leads to the establishment of an inter-AS MDT between the autonomous systems in the sample inter-AS Option C topology illustrated in Figure 12. For this topology, assume that all the routers have been configured properly to support all features associated with the Multicast VPN Inter-AS Support feature.

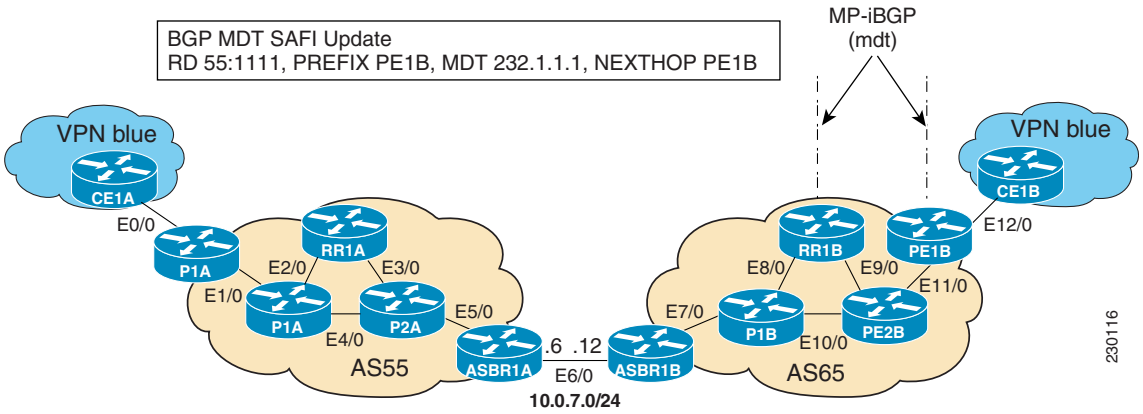
Figure 12 MVPN Inter-AS Support Option C Sample Topology



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

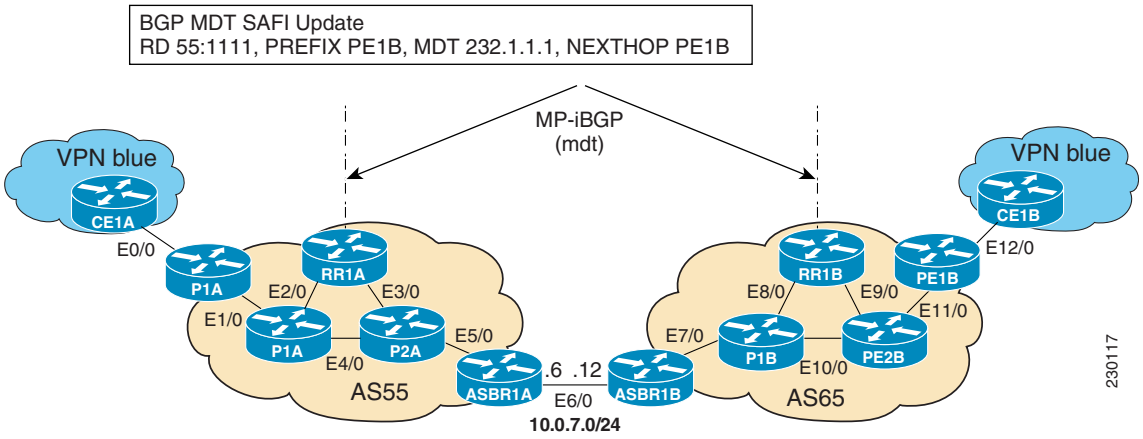
1. As illustrated in Figure 13, PE1B advertises the default MDT information for VPN blue to RR1B within the BGP MDT SAFI.

Figure 13 BGP MDT SAFI Update from PE1B to RR1B



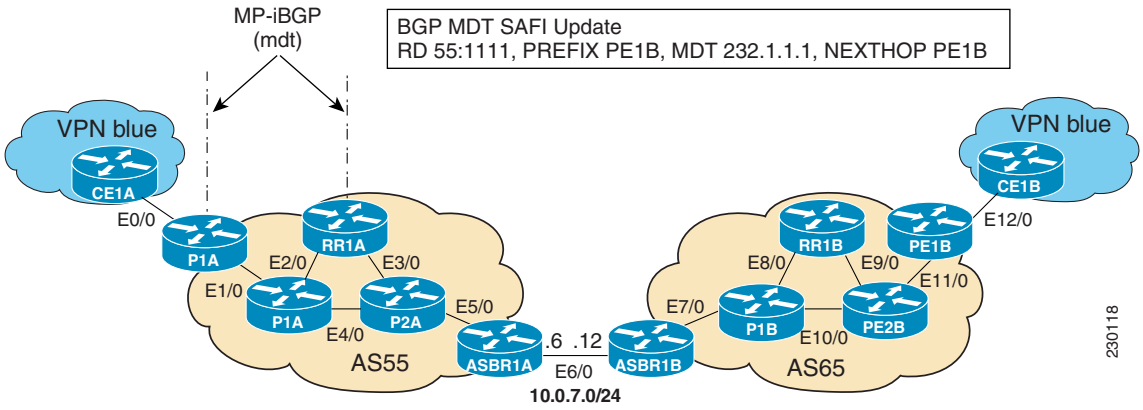
2. As illustrated in Figure 14, RR1B receives the MDT SAFI information, and, in turn, advertises this information to RR1A.

Figure 14 BGP MDT SAFI Update from RR1B to RR1A



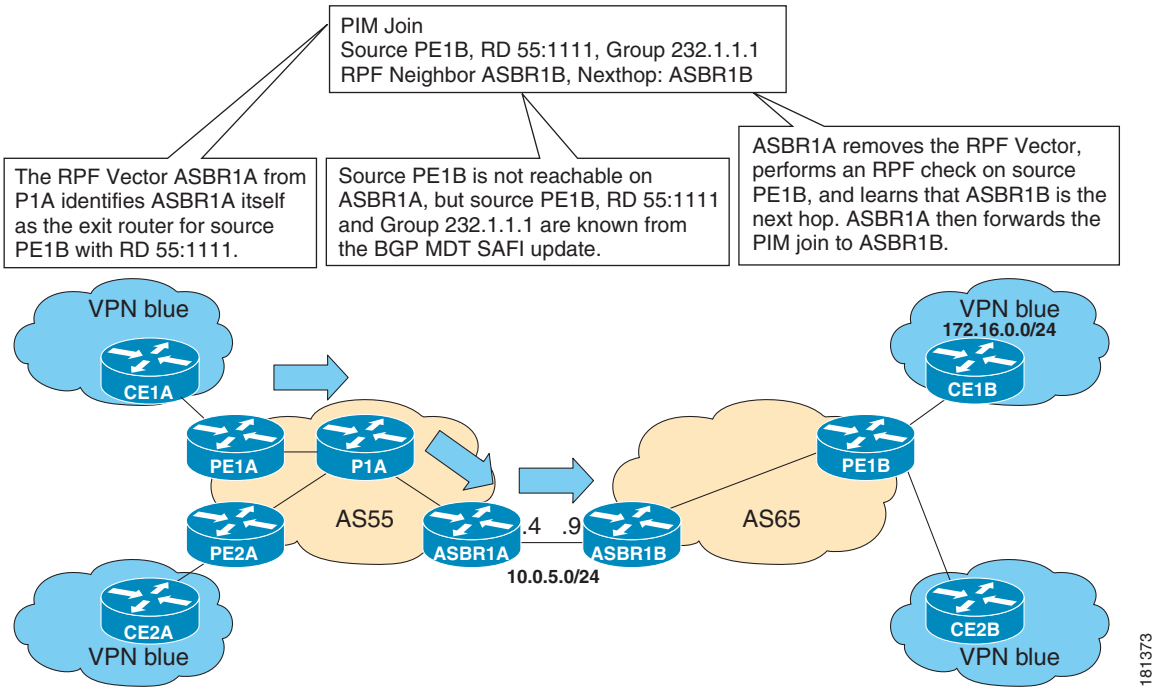
3. As illustrated in Figure 15, RR1A receives the MDT SAFI information, and, in turn, advertises this information to PE1A.

Figure 15 BGP MDT SAFI Update from RR1A to PE1A



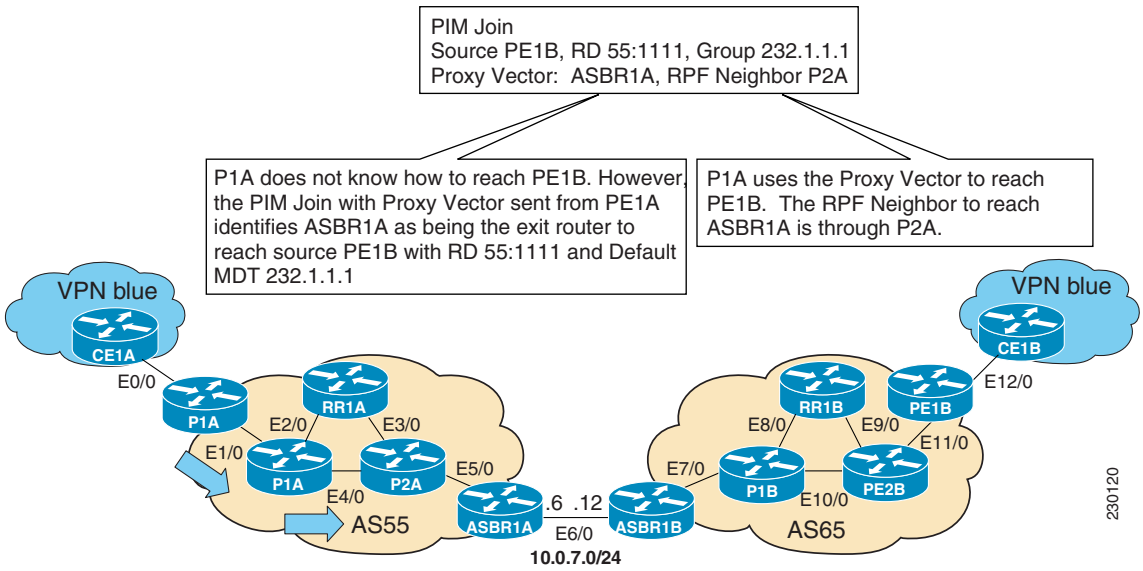
4. As illustrated in Figure 16, PE1A sends a PIM Join with the Proxy Vector that identifies ASBR1A as the exit router to reach source PE1B with RD 55:1111 and Default MDT 232.1.1.1. The Proxy Vector provides P1A and P2A a hint on how to reach PE1B in the absence of a route to reach PE1B. Source PE1B is reachable through RPF neighbor P1A through BGP IPv4 learned updates on PE1A.

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



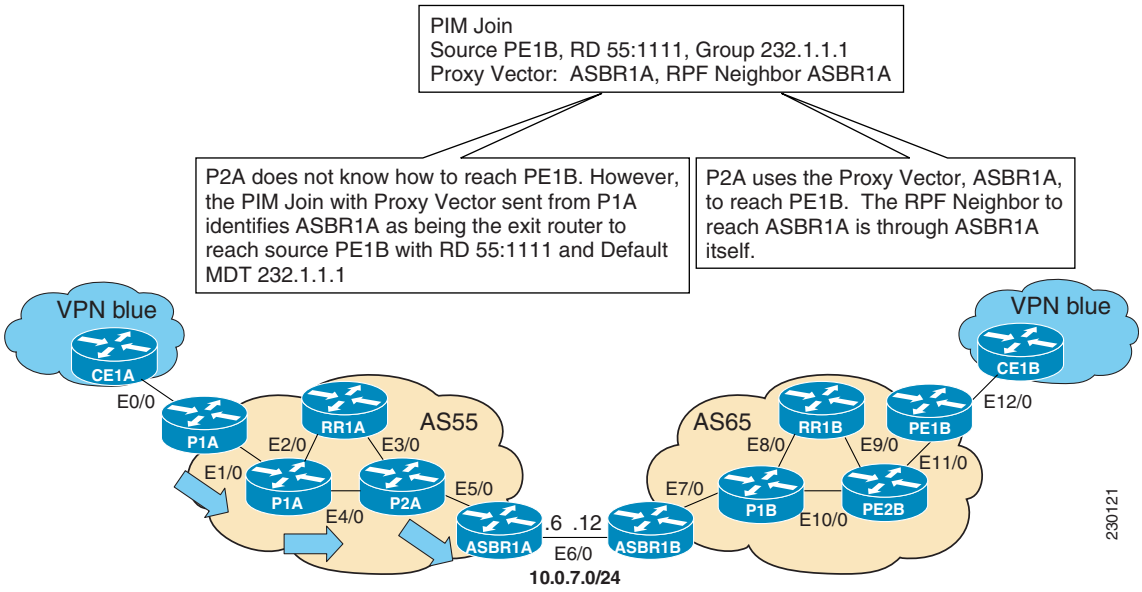
5. As illustrated in Figure 17, P1A does not know how to reach PE1B. However, the PIM join with the Proxy Vector sent from PE1A identifies ASBR1A as being the exit router to reach source PE1B with RD 55:1111 and Default MDT 232.1.1.1. P1A uses the Proxy Vector to reach PE1B. The RPF neighbor to reach ASBR1A is through P2A. P1A, thus, forwards the PIM SSM join to P2A.

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



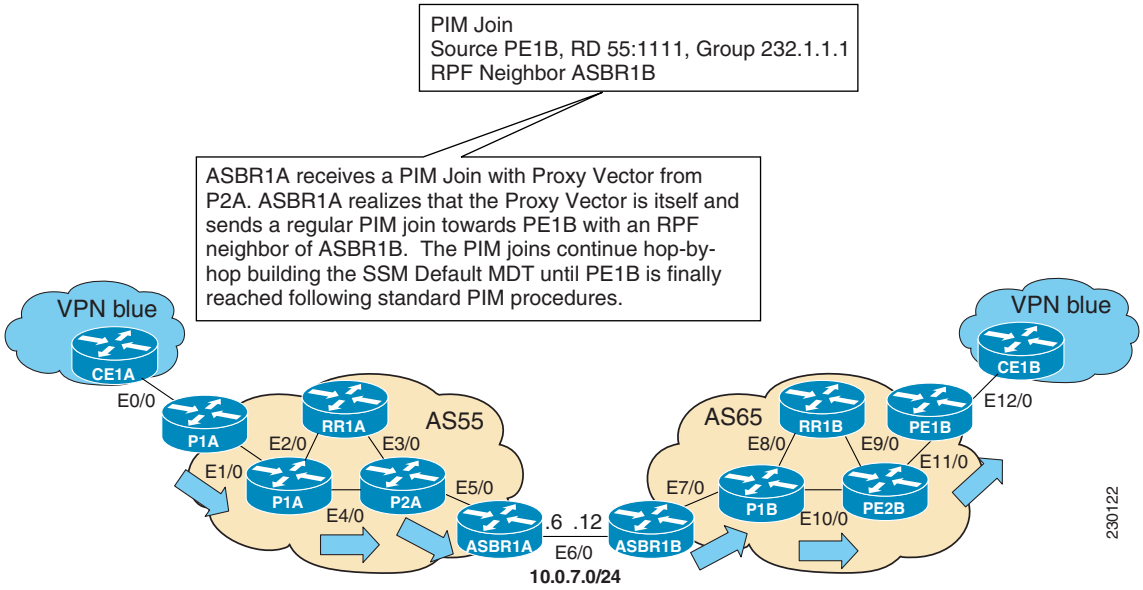
6. As illustrated in Figure 18, P2A does not know how to reach PE1B. However, the PIM join with the Proxy Vector sent from P1A identifies ASBR1A as being the exit router to reach source PE1B with RD 55:1111 and Default MDT 232.1.1.1. P2A uses the Proxy Vector, ASBR1A, to reach PE1B. The RPF neighbor to reach ASBR1B is through ASBR1A (that is, itself).

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



7. As illustrated in Figure 19, ASBR1A receives a PIM Join with Proxy Vector from P2A. ASBR1A realizes that the Proxy Vector is itself and sends a regular PIM join towards PE1B with an RPF neighbor of ASBR1B. The PIM joins continue hop-by-hop building the SSM Default MDT until PE1B is finally reached following standard PIM procedures.

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



How to Configure Multicast VPN Inter-AS Support

This section contains the following tasks:

- Configuring the MDT Address Family in BGP for Multicast VPN Inter-AS Support, page 24 (required)
- Displaying Information about IPv4 MDT Sessions in BGP, page 26 (optional)
- Clearing IPv4 MDT Peering Sessions in BGP, page 26 (optional)
- Configuring a PE Router to Send BGP MDT Updates to Build the Default MDT for MVPN Inter-AS Support (Option B), page 28 (required)
- Configuring a PE Router to Send BGP MDT Updates to Build the Default MDT for MVPN Inter-AS Support (Option C), page 30 (required)
- Verifying the Establishment of Inter-AS MDTs in Option B and Option C Deployments, page 32 (optional)

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

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

Supported Policy

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

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

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

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

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

Prerequisites

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

Restrictions

The following policy configuration parameters are not supported:

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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*

4. **address-family ipv4 mdt**
5. **neighbor** *neighbor-address* **activate**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 65535`	Enters router configuration mode and creates a BGP routing process.
Step 4	**address-family ipv4 mdt** **Example:** `Router(config-router)# address-family ipv4 mdt`	Enters address family configuration to create an IP MDT address family session.
Step 5	**neighbor** *neighbor-address* **activate** **Example:** `Router(config-router-af)# neighbor 192.168.1.1 activate`	Enables the MDT address family for this neighbor.
Step 6	**end** **Example:** `Router(config-router-af)# end`	Exits address-family configuration mode and enters privileged EXEC mode.

Displaying Information about IPv4 MDT Sessions in BGP

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

SUMMARY STEPS

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

DETAILED STEPS

Step 1 **enable**

Use this command to enable privileged EXEC mode.

- Enter your password if prompted.

```
Router> enable
```

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

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

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

```
Router# show ip bgp ipv4 mdt all

BGP table version is 2, local router ID is 10.1.0.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 55:1111 (default for vrf blue)
*> 10.5.5.5/32      10.1.0.1                 55             0 23 24 25 54 ?
*  10.9.9.9/32      0.0.0.0                                  0 ?
```

Clearing IPv4 MDT Peering Sessions in BGP

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ip bgp ipv4 mdt** *as-number* [**in** [**prefix-filter**]] [**out**] [**soft** [**in** [**prefix-filter**] | **out**]] or **clear ip bgp ipv4 mdt peer-group** *peer-group-name* [**in** [**prefix-filter**]] [**out**] [**soft** [**in** [**prefix-filter**] | **out**]] or **clear ip bgp ipv4 mdt update-group** [*index-group* | *neighbor-address*] **Example:** `Router# clear ip bgp ipv4 mdt 65700` or `Router# clear ip bgp ipv4 mdt peer-group test soft in` or `Router# clear ip bgp ipv4 mdt update-group 3`	(Optional) Resets IPv4 MDT address-family sessions • Specifying the **clear ip bgp ipv4 mdt** command with the *as-number* argument resets IPv4 MDT address-family sessions associated with the specified autonomous system. • The example for this form of the **clear ip bgp ipv4 mdt** command shows how to initiate a hard reset for all IPv4 MDT address family sessions in the autonomous system numbered 65700. or • Specifying the **clear ip bgp ipv4 mdt** command with the **peer-group** keyword resets IPv4 MDT address-family sessions for all members of a BGP peer group. • The example for this form of the **clear ip bgp ipv4 mdt** command shows how to initiate a soft reset for inbound MDT address family sessions with members of the peer group test. Outbound sessions are unaffected. or • Specifying the **clear ip bgp ipv4 mdt** command with the **update-group** keyword resets IPv4 MDT address-family sessions for all the members of a BGP update group. • The example for this form of the **clear ip bgp ipv4 mdt** command shows how to reset for all members of the update group 3.

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip multicast vrf** *vrf-name* **rpf proxy rd vector** **Example:** `Router(config)# ip multicast vrf blue rpf proxy rd vector`	Enables the RPF Vector on the exit router in a specific VPN. • Use the **rd** keyword to configure PE routers to include the RD value of the VPN associated with the PIM RPF Vector encoding inserted into PIM join and prune messages. Because ASBRs in Option B deployments change the next hop of the originating PE router for a given MDT group, including the RD value in the PIM RPF Vector encoding enables the ASBR to perform a lookup on the RD value for a prefix, which, in turn, enables the ASBR to identify which VPN the RPF Vector is intended for. **Note** In an Option B deployment, you must enter the **ip multicast rpf proxy** command with the **rd** keyword for MVPN inter-AS support. The **rd** keyword is not required for MVPN inter-AS support Option C deployments.
Step 4	**router bgp** *as-number* **Example:** `Router(config)# router bgp 101`	Enters router configuration mode for the specified routing process.
Step 5	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 192.168.1.1 remote-as 45000`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 6	**address-family ipv4 mdt** **Example:** `Router(config-router)# address-family ipv4 mdt`	Enters address family configuration to create an IPv4 MDT address family session.
Step 7	**neighbor** *neighbor-address* **activate** **Example:** `Router(config-router-af)# neighbor 192.168.1.1 activate`	Enables the MDT address family for this neighbor.
Step 8	**neighbor** *neighbor-address* **next-hop-self** **Example:** `Router(config-router-af)# neighbor 192.168.1.1 next-hop-self`	Disables next hop processing of BGP updates on the router.
Step 9	**exit** **Example:** `Router(config-router-af)# exit`	Exits address family configuration mode and returns to router configuration mode.

	Command or Action	Purpose
Step 10	**address-family vpnv4** **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode to create a VPNv4 address family session.
Step 11	**neighbor** *neighbor-address* **activate** **Example:** `Router(config-router-af)# neighbor 192.168.1.1 activate`	Enables the VPNv4 address family for this neighbor.
Step 12	**neighbor** *neighbor-address* **send-community extended** **Example:** `Router(config-router-af)# neighbor 192.168.1.1 send-community extended`	Enables the standard and extended community attributes to be sent to this neighbor.
Step 13	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast rpf proxy vector** **Example:** `Router(config)# ip multicast rpf proxy vector`	Enables the RPF Vector on the exit router. • Perform this step if there is no IGP reachability between PEs in different autonomous systems.
Step 4	**router bgp** *as-number* **Example:** `Router(config)# router bgp 101`	Enters router configuration mode for the specified routing process.
Step 5	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 192.168.1.1 remote-as 45000`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 6	**address-family ipv4 mdt** **Example:** `Router(config-router)# address-family ipv4 mdt`	Enters address family configuration to create an IPv4 MDT address family session.
Step 7	**neighbor** *neighbor-address* **activate** **Example:** `Router(config-router-af)# neighbor 192.168.1.1 activate`	Enables the MDT address family for this neighbor.
Step 8	**neighbor** *neighbor-address* **send-community extended** **Example:** `Router(config-router-af)# neighbor 192.168.1.1 send-community extended`	Enables the standard and extended community attributes to be sent to this neighbor.
Step 9	**exit** **Example:** `Router(config-router-af)# exit`	Exits address family configuration mode and returns to router configuration mode.

	Command or Action	Purpose
Step 10	**address-family vpnv4** **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode to create a VPNv4 address family session.
Step 11	**neighbor** *neighbor-address* **activate** **Example:** `Router(config-router-af)# neighbor 192.168.1.1 activate`	Enables the VPNv4 address family for this neighbor.
Step 12	**neighbor** *neighbor-address* **send-community extended** **Example:** `Router(config-router-af)# neighbor 192.168.1.1 send-community extended`	Enables the standard and extended community attributes to be sent to this neighbor.
Step 13	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

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

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

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

SUMMARY STEPS

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

DETAILED STEPS

Step 1 **enable**

Use this command to enable privileged EXEC mode.

- Enter your password if prompted.

```
Router> enable
```

Step 2 **show ip mroute proxy**

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

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

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

```
Router# show ip mroute proxy

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

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

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

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

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

```
Router# show ip pim neighbor

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

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

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

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

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

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

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

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

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

```
Router# show ip pim mdt bgp

MDT (Route Distinguisher + IPv4)              Router ID         Next Hop
  MDT group 232.1.1.1
   55:1111:192.168.0.2                        192.168.0.2       192.168.0.2
   55:1111:192.168.0.8                        192.168.0.4       192.168.0.4
```

Configuration Examples for Multicast VPN Inter-AS Support

This section provides the following configuration examples:

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

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

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

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

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

```
!
```

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

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

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

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

```
 exit-address-family
 !
 address-family ipv4 mdt
 neighbor 10.252.252.10 activate
 neighbor 10.252.252.10 send-community extended
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.252.252.10 activate
 neighbor 10.252.252.10 send-community extended
 exit-address-family
 !
.
.
.
!
ip pim ssm default
!
```

Configuring Back-to-Back ASBR PEs (Option A): Example

The following example shows how to configure support for MVPN inter-AS support Option A. This configuration example is based on the sample inter-AS network Option A topology illustrated in Figure 20.

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

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

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

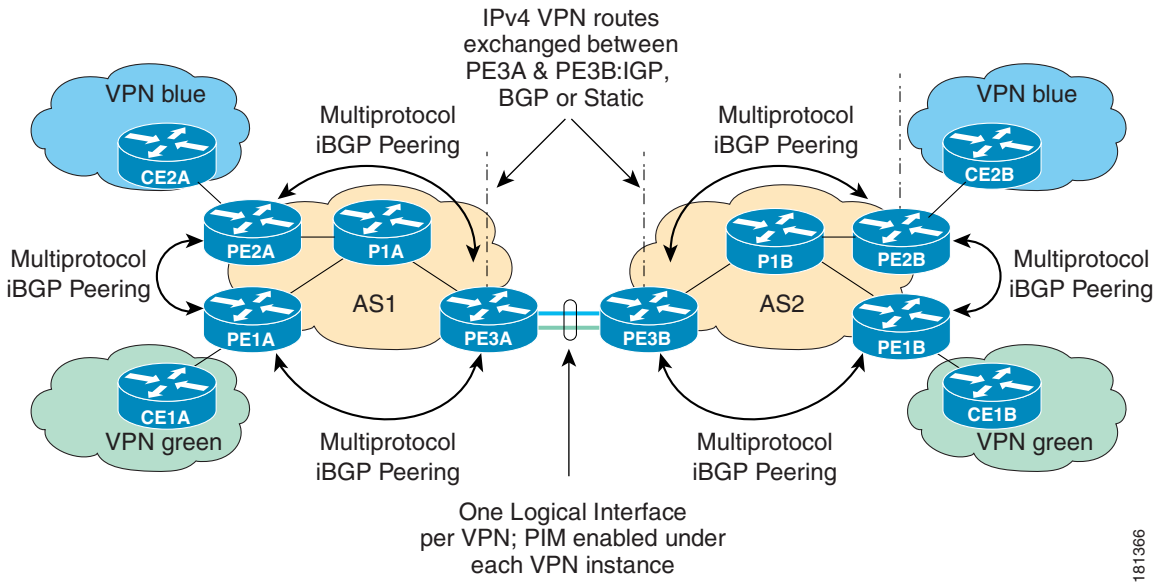


Table 1 provides information about the topology used for the Option A configuration example presented in this section.

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

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

PE1A

```
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname PE1A
!
boot-start-marker
boot-end-marker
!
!
ip subnet-zero
ip cef
no ip domain-lookup
ip vrf green
```

```
 rd 55:2222
 route-target export 55:2222
 route-target import 55:2222
 mdt default 232.2.2.2
!
ip multicast-routing
ip multicast-routing vrf green
mpls label protocol ldp
!
!
!
interface Loopback0
 ip address 10.1.1.1 255.255.255.255
 no ip directed-broadcast
 ip router isis
 ip pim sparse-mode
!
interface Ethernet0/0
 ip vrf forwarding green
 ip address 172.25.11.1 255.255.255.0
 no ip directed-broadcast
 ip pim sparse-mode
 tag-switching ip
!
interface Ethernet1/0
 ip address 172.30.41.1 255.255.255.0
 no ip directed-broadcast
 ip router isis
 ip pim sparse-mode
 tag-switching ip
!
router isis
 net 49.0000.0000.1111.00
!
router rip
 version 2
 !
 address-family ipv4 vrf green
 version 2
 network 172.25.0.0
 no auto-summary
 exit-address-family
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.1.1.2 remote-as 1
 neighbor 10.1.1.2 update-source Loopback0
 neighbor 10.1.1.3 remote-as 1
 neighbor 10.1.1.3 update-source Loopback0
 no auto-summary
 !
 address-family ipv4 mdt
 neighbor 10.1.1.2 activate
 neighbor 10.1.1.3 activate
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.1.1.2 activate
 neighbor 10.1.1.2 send-community extended
 neighbor 10.1.1.3 activate
 neighbor 10.1.1.3 send-community extended
 exit-address-family
 !
```

```
 address-family ipv4 vrf green
 redistribute rip metric 1
 no synchronization
 exit-address-family
!
ip classless
!
ip pim ssm default
ip pim vrf green send-rp-announce Ethernet0/0 scope 32
ip pim vrf green send-rp-discovery Ethernet0/0 scope 32
ip pim vrf green register-rate-limit 2
!
!
!
control-plane
!
!
line con 0
line aux 0
line vty 0 4
 login
!
no cns aaa enable
end
```

PE2A

```
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname PE2A
!
boot-start-marker
boot-end-marker
!
!
ip subnet-zero
ip cef
no ip domain-lookup
ip vrf blue
 rd 55:1111
 route-target export 55:1111
 route-target import 55:1111
 mdt default 232.1.1.1
!
ip multicast-routing
ip multicast-routing vrf blue
mpls label protocol ldp
!
!
!
interface Loopback0
 ip address 10.1.1.2 255.255.255.255
 no ip directed-broadcast
 ip router isis
 ip pim sparse-mode
!
interface Ethernet0/0
 ip vrf forwarding blue
 ip address 172.17.12.2 255.255.255.0
 no ip directed-broadcast
```

```
 ip pim sparse-mode
 tag-switching ip
!
interface Ethernet1/0
 no ip address
 no ip directed-broadcast
 shutdown
!
interface Ethernet2/0
 ip address 172.19.142.2 255.255.255.0
 no ip directed-broadcast
 ip router isis
 ip pim sparse-mode
 tag-switching ip
!
router isis
 net 49.0000.0000.2222.00
!
router rip
 version 2
 !
 address-family ipv4 vrf blue
 version 2
 network 172.17.0.0
 no auto-summary
 exit-address-family
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.1.1.1 remote-as 1
 neighbor 10.1.1.1 update-source Loopback0
 neighbor 10.1.1.3 remote-as 1
 neighbor 10.1.1.3 update-source Loopback0
 no auto-summary
 !
 address-family ipv4 mdt
 neighbor 10.1.1.1 activate
 neighbor 10.1.1.3 activate
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.1.1.1 activate
 neighbor 10.1.1.1 send-community extended
 neighbor 10.1.1.3 activate
 neighbor 10.1.1.3 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf blue
 redistribute rip metric 1
 no synchronization
 exit-address-family
!
ip classless
!
ip pim ssm default
ip pim vrf blue send-rp-announce Ethernet0/0 scope 32
ip pim vrf blue send-rp-discovery Ethernet0/0 scope 32
ip pim vrf blue ssm default
!
!
!
control-plane
!
```

```
!
line con 0
line aux 0
line vty 0 4
 login
!
no cns aaa enable
end
```

PE3A

```
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname PE3A
!
boot-start-marker
boot-end-marker
!
!
ip subnet-zero
ip cef
no ip domain-lookup
ip vrf blue
 rd 55:1111
 route-target export 55:1111
 route-target import 55:1111
 mdt default 232.1.1.1
!
ip vrf green
 rd 55:2222
 route-target export 55:2222
 route-target import 55:2222
 mdt default 232.2.2.2
!
ip multicast-routing
ip multicast-routing vrf blue
ip multicast-routing vrf green
mpls label protocol ldp
!
!
!
interface Loopback0
 ip address 10.1.1.3 255.255.255.255
 no ip directed-broadcast
 ip router isis
 ip pim sparse-mode
!
interface Ethernet0/0
 no ip address
 no ip directed-broadcast
 shutdown
!
interface Ethernet1/0
 no ip address
 no ip directed-broadcast
 shutdown
!
interface Ethernet2/0
 no ip address
 no ip directed-broadcast
```

```
 shutdown
!
interface Ethernet3/0
 ip address 192.168.143.3 255.255.255.0
 no ip directed-broadcast
 ip router isis
 ip pim sparse-mode
 tag-switching ip
!
interface Ethernet4/0
 ip vrf forwarding blue
 ip address 172.20.34.3 255.255.255.0
 no ip directed-broadcast
 ip pim sparse-dense-mode
 tag-switching ip
!
interface Ethernet5/0
 ip vrf forwarding green
 ip address 172.23.35.3 255.255.255.0
 no ip directed-broadcast
 ip pim sparse-dense-mode
 tag-switching ip
!
router eigrp 1
 !
 address-family ipv4 vrf blue
 network 172.20.0.0
 no auto-summary
 exit-address-family
!
router isis
 net 49.0000.0000.3333.00
!
router rip
 version 2
 !
 address-family ipv4 vrf green
 version 2
 redistribute bgp 1 metric 2
 network 172.23.0.0
 no auto-summary
 exit-address-family
 !
 address-family ipv4 vrf blue
 version 2
 redistribute bgp 1 metric 1
 network 172.20.0.0
 no auto-summary
 exit-address-family
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.1.1.1 remote-as 1
 neighbor 10.1.1.1 update-source Loopback0
 neighbor 10.1.1.2 remote-as 1
 neighbor 10.1.1.2 update-source Loopback0
 no auto-summary
 !
 address-family ipv4 mdt
 neighbor 10.1.1.1 activate
 neighbor 10.1.1.2 activate
 exit-address-family
 !
```

```
 address-family vpnv4
 neighbor 10.1.1.1 activate
 neighbor 10.1.1.1 send-community extended
 neighbor 10.1.1.2 activate
 neighbor 10.1.1.2 send-community extended
 bgp redistribute-internal
 exit-address-family
 !
 address-family ipv4 vrf green
 no synchronization
 bgp redistribute-internal
 exit-address-family
 !
 address-family ipv4 vrf blue
 redistribute rip
 no synchronization
 bgp redistribute-internal
 exit-address-family
!
ip classless
!
ip pim ssm default
ip pim vrf blue ssm default
!
!
!
control-plane
!
!
line con 0
line aux 0
line vty 0 4
 login
!
no cns aaa enable
end
```

PE3B

```
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname PE3B
!
boot-start-marker
boot-end-marker
!
!
ip subnet-zero
ip cef
no ip domain-lookup
ip vrf blue
 rd 55:1111
 route-target export 55:1111
 route-target import 55:1111
 mdt default 232.1.1.1
!
ip vrf green
 rd 55:2222
 route-target export 55:2222
 route-target import 55:2222
```

```
 mdt default 232.2.2.2
!
ip multicast-routing
ip multicast-routing vrf blue
ip multicast-routing vrf green
mpls label protocol ldp
!
!
!
interface Loopback0
 ip address 10.2.2.3 255.255.255.255
 no ip directed-broadcast
 ip router isis
 ip pim sparse-mode
!
interface Ethernet0/0
 no ip address
 no ip directed-broadcast
 shutdown
!
interface Ethernet1/0
 no ip address
 no ip directed-broadcast
 shutdown
!
interface Ethernet2/0
 no ip address
 no ip directed-broadcast
 shutdown
!
interface Ethernet3/0
 ip address 172.16.43.3 255.255.255.0
 no ip directed-broadcast
 ip router isis
 ip pim sparse-mode
 tag-switching ip
!
interface Ethernet4/0
 ip vrf forwarding blue
 ip address 172.20.34.4 255.255.255.0
 no ip directed-broadcast
 ip pim sparse-dense-mode
 tag-switching ip
!
interface Ethernet5/0
 ip vrf forwarding green
 ip address 172.23.35.4 255.255.255.0
 no ip directed-broadcast
 ip pim sparse-dense-mode
 tag-switching ip
!
router isis
 net 49.0000.0000.3333.00
!
router rip
 version 2
 !
 address-family ipv4 vrf green
 version 2
 network 172.23.0.0
 no auto-summary
 exit-address-family
 !
 address-family ipv4 vrf blue
```

```
 version 2
 network 172.20.0.0
 no auto-summary
 exit-address-family
!
router bgp 2
 no synchronization
 bgp log-neighbor-changes
 redistribute rip metric 1
 neighbor 10.2.2.1 remote-as 2
 neighbor 10.2.2.1 update-source Loopback0
 neighbor 10.2.2.2 remote-as 2
 neighbor 10.2.2.2 update-source Loopback0
 no auto-summary
 !
 address-family ipv4 mdt
 neighbor 10.2.2.1 activate
 neighbor 10.2.2.2 activate
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.2.2.1 activate
 neighbor 10.2.2.1 send-community extended
 neighbor 10.2.2.2 activate
 neighbor 10.2.2.2 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf green
 redistribute rip
 no synchronization
 exit-address-family
 !
 address-family ipv4 vrf blue
 redistribute rip
 no synchronization
 exit-address-family
!
ip classless
!
ip pim ssm default
ip pim vrf blue ssm default
!
!
!
control-plane
!
!
line con 0
line aux 0
line vty 0 4
 login
!
no cns aaa enable
end
```

PE2B

```
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname PE2B
```

```
!
boot-start-marker
boot-end-marker
!
!
ip subnet-zero
ip cef
no ip domain-lookup
ip vrf blue
 rd 55:1111
 route-target export 55:1111
 route-target import 55:1111
 mdt default 232.1.1.1
!
ip multicast-routing
ip multicast-routing vrf blue
mpls label protocol ldp
!
!
!
interface Loopback0
 ip address 10.2.2.2 255.255.255.255
 no ip directed-broadcast
 ip router isis
 ip pim sparse-mode
!
interface Ethernet0/0
 ip vrf forwarding blue
 ip address 172.18.22.2 255.255.255.0
 no ip directed-broadcast
 ip pim sparse-mode
 tag-switching ip
!
interface Ethernet1/0
 no ip address
 no ip directed-broadcast
 shutdown
!
interface Ethernet2/0
 ip address 172.19.42.2 255.255.255.0
 no ip directed-broadcast
 ip router isis
 ip pim sparse-mode
 tag-switching ip
!
router isis
 net 49.0000.0000.2222.00
!
router rip
 !
 address-family ipv4 vrf blue
 network 172.18.0.0
 no auto-summary
 exit-address-family
!
router bgp 2
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.2.2.1 remote-as 2
 neighbor 10.2.2.1 update-source Loopback0
 neighbor 10.2.2.3 remote-as 2
 neighbor 10.2.2.3 update-source Loopback0
 no auto-summary
 !
```

```
 address-family ipv4 mdt
 neighbor 10.2.2.1 activate
 neighbor 10.2.2.3 activate
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.2.2.1 activate
 neighbor 10.2.2.1 send-community extended
 neighbor 10.2.2.3 activate
 neighbor 10.2.2.3 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf blue
 no synchronization
 exit-address-family
!
ip classless
!
ip pim ssm default
ip pim vrf blue ssm default
!
!
!
control-plane
!
!
line con 0
line aux 0
line vty 0 4
 login
!
no cns aaa enable
end
```

PE1B

```
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname PE1B
!
boot-start-marker
boot-end-marker
!
!
ip subnet-zero
ip cef
no ip domain-lookup
ip vrf green
 rd 55:2222
 route-target export 55:2222
 route-target import 55:2222
 mdt default 232.2.2.2
!
ip multicast-routing
ip multicast-routing vrf green
mpls label protocol ldp
!
!
!
interface Loopback0
```

```
 ip address 10.2.2.1 255.255.255.255
 no ip directed-broadcast
 ip router isis
 ip pim sparse-mode
!
interface Ethernet0/0
 ip vrf forwarding green
 ip address 172.25.111.1 255.255.255.0
 no ip directed-broadcast
 ip pim sparse-mode
 tag-switching ip
!
interface Ethernet1/0
 ip address 172.30.141.1 255.255.255.0
 no ip directed-broadcast
 ip router isis
 ip pim sparse-mode
 tag-switching ip
!
router isis
 net 49.0000.0000.1111.00
!
router rip
 version 2
 !
 address-family ipv4 vrf green
 version 2
 network 172.25.0.0
 no auto-summary
 exit-address-family
!
router bgp 2
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.2.2.2 remote-as 2
 neighbor 10.2.2.2 update-source Loopback0
 neighbor 10.2.2.3 remote-as 2
 neighbor 10.2.2.3 update-source Loopback0
 no auto-summary
 !
 address-family ipv4 mdt
 neighbor 10.2.2.2 activate
 neighbor 10.2.2.3 activate
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.2.2.2 activate
 neighbor 10.2.2.2 send-community extended
 neighbor 10.2.2.3 activate
 neighbor 10.2.2.3 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf green
 no synchronization
 exit-address-family
!
ip classless
!
ip pim ssm default
!
!
!
control-plane
!
```

```
!
line con 0
line aux 0
line vty 0 4
 login
!
no cns aaa enable
end
```

Configuring the Exchange of VPNv4 Routes Directly Between ASBRs (Option B): Example

The following example shows how to configure MVPN inter-AS support in an Option B deployment. This configuration is based on the sample inter-AS topology illustrated in Figure 21.

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

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

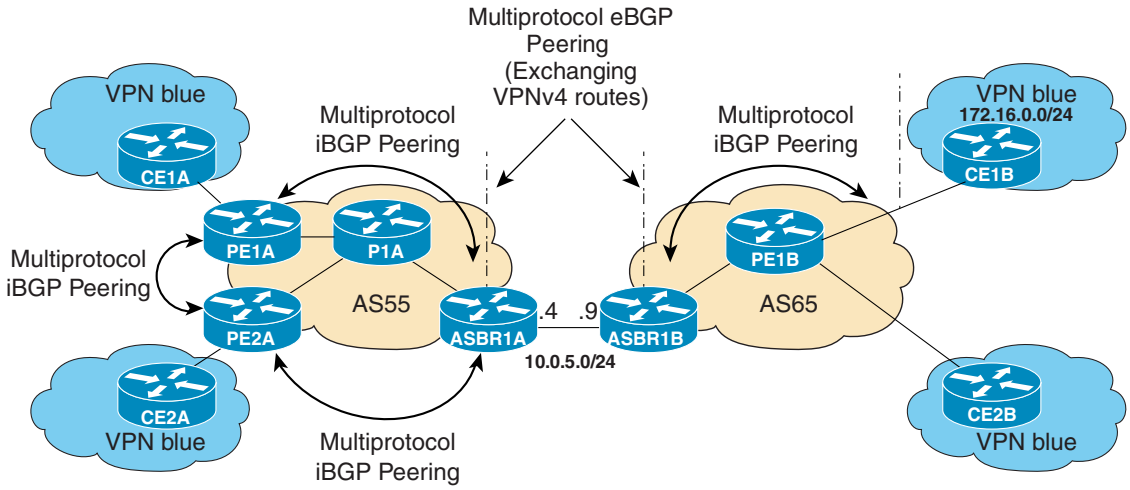


Table 2 provides information about the topology used for this particular Option B configuration example.

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

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

PE1A

```
!
ip vrf blue
 rd 55:1111
 mdt default 232.1.1.1
!
ip multicast-routing
ip multicast-routing vrf blue
ip multicast vrf blue rpf proxy rd vector
!
.
.
.
!
interface Ethernet0/0
 ip vrf forwarding blue
 ip pim sparse-mode
!
.
.
.
!
router bgp 55
 no synchronization
 bgp log-neighbor-changes
 neighbor 192.168.0.2 remote-as 55
 neighbor 192.168.0.2 update-source Loopback0
 neighbor 192.168.0.4 remote-as 55
 neighbor 192.168.0.4 update-source Loopback0
 no auto-summary
 !
 address-family ipv4 mdt
  neighbor 192.168.0.2 activate
  neighbor 192.168.0.2 next-hop-self
  neighbor 192.168.0.4 activate
  neighbor 192.168.0.4 next-hop-self
 exit-address-family
 !
 address-family vpnv4
  neighbor 192.168.0.2 activate
  neighbor 192.168.0.2 send-community extended
  neighbor 192.168.0.4 activate
  neighbor 192.168.0.4 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf blue
  redistribute connected
  redistribute static
  redistribute rip metric 50
  no auto-summary
  no synchronization
 exit-address-family
!
.
.
.
!
ip pim ssm default
!
```

PE2A

```
!
```

```
ip vrf blue
 rd 55:1111
 mdt default 232.1.1.1
!
ip multicast-routing
ip multicast-routing vrf blue
ip multicast vrf blue rpf proxy rd vector
!
.
.
.
!
interface Ethernet0/0
 ip vrf forwarding blue
 ip pim sparse-mode
!
.
.
.
!
router bgp 55
 neighbor 192.168.0.1 remote-as 55
 neighbor 192.168.0.1 update-source Loopback0
 neighbor 192.168.0.4 remote-as 55
 neighbor 192.168.0.4 update-source Loopback0
 !
 address-family ipv4 mdt
 neighbor 192.168.0.1 activate
 neighbor 192.168.0.1 next-hop-self
 neighbor 192.168.0.4 activate
 neighbor 192.168.0.4 next-hop-self
 exit-address-family
 !
 address-family vpnv4
 neighbor 192.168.0.1 activate
 neighbor 192.168.0.1 send-community extended
 neighbor 192.168.0.4 activate
 neighbor 192.168.0.4 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf blue
 redistribute connected
 redistribute static
 no synchronization
 exit-address-family
!
.
.
.
!
ip pim ssm default
!
```

ASBR1A

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

```
interface Ethernet0/0
 ip pim sparse-mode
!
.
.
.
!
router bgp 55
 bgp log-neighbor-changes
 neighbor 10.0.5.9 remote-as 65
 neighbor 192.168.0.1 remote-as 55
 neighbor 192.168.0.1 update-source Loopback0
 neighbor 192.168.0.2 remote-as 55
 neighbor 192.168.0.2 update-source Loopback0
 !
 address-family ipv4 mdt
 neighbor 10.0.5.9 activate
 neighbor 192.168.0.1 activate
 neighbor 192.168.0.1 next-hop-self
 neighbor 192.168.0.2 activate
 neighbor 192.168.0.2 next-hop-self
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.0.5.9 activate
 neighbor 10.0.5.9 send-community extended
 neighbor 192.168.0.1 activate
 neighbor 192.168.0.1 send-community extended
 neighbor 192.168.0.1 next-hop-self
 neighbor 192.168.0.2 activate
 neighbor 192.168.0.2 send-community extended
 neighbor 192.168.0.2 next-hop-self
 exit-address-family
!
.
.
.
!
ip pim ssm default
!
```

ASBR1B

```
!
ip multicast-routing
ip multicast-routing vrf blue
!
.
.
.
!
interface Ethernet0/0
 ip pim sparse-mode
!
.
.
.
!
router bgp 65
 bgp log-neighbor-changes
 neighbor 10.0.5.4 remote-as 55
 neighbor 192.168.0.8 remote-as 65
 neighbor 192.168.0.8 update-source Loopback0
 !
```

```
 address-family ipv4 mdt
 neighbor 10.0.5.4 activate
 neighbor 192.168.0.8 activate
 neighbor 192.168.0.8 next-hop-self
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.0.5.4 activate
 neighbor 10.0.5.4 send-community extended
 neighbor 192.168.0.8 activate
 neighbor 192.168.0.8 send-community extended
 neighbor 192.168.0.8 next-hop-self
 exit-address-family
!
.
.
.
!
ip pim ssm default
!
```

PE1B

```
!
ip vrf blue
 rd 55:1111
 mdt default 232.1.1.1
!
ip multicast-routing
ip multicast-routing vrf blue
ip multicast vrf blue rpf proxy rd vector
!
.
.
.
!
interface Ethernet0/0
 ip vrf forwarding blue
 ip pim sparse-mode
!
.
.
.
!
router bgp 65
 neighbor 192.168.0.9 remote-as 65
 neighbor 192.168.0.9 update-source Loopback0
 !
 address-family ipv4 mdt
 neighbor 192.168.0.9 activate
 neighbor 192.168.0.9 next-hop-self
 exit-address-family
 !
 address-family vpnv4
 neighbor 192.168.0.9 activate
 neighbor 192.168.0.9 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf blue
 redistribute connected
 redistribute static
 redistribute rip metric 50
 no synchronization
 exit-address-family
```

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

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

```
PE1A# show ip pim mdt bgp

MDT (Route Distinguisher + IPv4)              Router ID         Next Hop
  MDT group 232.1.1.1
   55:1111:192.168.0.2                        192.168.0.2       192.168.0.2
   55:1111:192.168.0.8                        192.168.0.4       192.168.0.4

PE2A# show ip pim mdt bgp

MDT (Route Distinguisher + IPv4)              Router ID         Next Hop
  MDT group 232.1.1.1
   55:1111:192.168.0.1                        192.168.0.1       192.168.0.1
   55:1111:192.168.0.8                        192.168.0.4       192.168.0.4

PE1B# show ip pim mdt bgp

MDT (Route Distinguisher + IPv4)              Router ID         Next Hop
  MDT group 232.1.1.1
   55:1111:192.168.0.1                        192.168.0.9       192.168.0.9
   55:1111:192.168.0.2                        192.168.0.9       192.168.0.9
```

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

```
PE1A# show ip mroute proxy

(192.168.0.8, 232.1.1.1)
  Proxy                    Assigner         Origin    Uptime/Expire
  55:1111/192.168.0.4      0.0.0.0          BGP MDT   00:13:07/stopped

PE2A# show ip mroute proxy

(192.168.0.8, 232.1.1.1)
  Proxy                    Assigner         Origin    Uptime/Expire
  55:1111/192.168.0.4      0.0.0.0          BGP MDT   00:14:28/stopped

PE1B# show ip mroute proxy

(192.168.0.1, 232.1.1.1)
  Proxy                    Assigner         Origin    Uptime/Expire
  55:1111/192.168.0.9      0.0.0.0          BGP MDT   00:35:19/stopped
```

```
(192.168.0.2, 232.1.1.1)
  Proxy                     Assigner         Origin    Uptime/Expire
  55:1111/192.168.0.9       0.0.0.0          BGP MDT   00:35:49/stopped
```

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

```
P1A# show ip mroute proxy

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

The following is sample output from the **show ip mroute proxy** command for ASBR1A and ASBR1B. If a router receives an RPF Vector that references a local interface (which occurs in an Option B deployment when an ASBR receives a RPF Vector owned by a local interface), the router discards the RPF Vector and performs a normal RPF lookup using information that the router learned from BGP MDT SAFI updates. In the output for all ASBR routers, under the "Proxy" field, the word "local" is displayed instead of the RPF Vector because ASBR1A and ASBR1B are using local interfaces to perform RPF lookups for PIM joins with RPF Vectors that reference one of their local interfaces. The "Assigner" field displays the RPF address that sent the PIM join to the ASBR. Because the ASBRs learn the RPF Vectors from the PIM joins (the RPF Vectors that are subsequently discarded), PIM is displayed as the origin under the "Origin" field.

```
ASBR1A# show ip mroute proxy

(192.168.0.1, 232.1.1.1)
  Proxy                     Assigner         Origin    Uptime/Expire
  55:1111/local             10.0.5.9         PIM       00:18:19/00:02:46

(192.168.0.2, 232.1.1.1)
  Proxy                     Assigner         Origin    Uptime/Expire
  55:1111/local             10.0.5.9         PIM       00:18:50/00:02:24

(192.168.0.8, 232.1.1.1)
  Proxy                     Assigner         Origin    Uptime/Expire
  55:1111/local             10.0.4.3         PIM       00:18:49/00:02:19

ASBR1B# show ip mroute proxy

(192.168.0.1, 232.1.1.1)
  Proxy                     Assigner         Origin    Uptime/Expire
  55:1111/local             10.0.7.8         PIM       00:37:39/00:02:44

(192.168.0.2, 232.1.1.1)
  Proxy                     Assigner         Origin    Uptime/Expire
  55:1111/local             10.0.7.8         PIM       00:38:10/00:02:19

(192.168.0.8, 232.1.1.1)
  Proxy                     Assigner         Origin    Uptime/Expire
  55:1111/local             10.0.5.4         PIM       00:38:09/00:02:19
```

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

PE1A

```
PE1A# show ip mroute 232.1.1.1

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

(192.168.0.8, 232.1.1.1), 00:13:11/00:02:41, flags: sTIZV
  Incoming interface: Ethernet2/0, RPF nbr 10.0.3.3, vector 192.168.0.4
  Outgoing interface list:
    MVRF blue, Forward/Sparse-Dense, 00:13:11/00:00:00

(192.168.0.2, 232.1.1.1), 00:13:12/00:02:41, flags: sTIZ
  Incoming interface: Ethernet2/0, RPF nbr 10.0.3.2
  Outgoing interface list:
    MVRF blue, Forward/Sparse-Dense, 00:13:12/00:00:00

(192.168.0.1, 232.1.1.1), 00:13:12/00:03:27, flags: sT
  Incoming interface: Loopback0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet2/0, Forward/Sparse-Dense, 00:13:11/00:02:50
```

PE2A

```
PE2A# show ip mroute 232.1.1.1

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

(192.168.0.8, 232.1.1.1), 00:17:05/00:02:46, flags: sTIZV
  Incoming interface: Ethernet2/0, RPF nbr 10.0.3.3, vector 192.168.0.4
  Outgoing interface list:
    MVRF blue, Forward/Sparse-Dense, 00:17:05/00:00:00

(192.168.0.1, 232.1.1.1), 00:17:05/00:02:46, flags: sTIZ
  Incoming interface: Ethernet2/0, RPF nbr 10.0.3.1
  Outgoing interface list:
    MVRF blue, Forward/Sparse-Dense, 00:17:05/00:00:00
```

```
(192.168.0.2, 232.1.1.1), 00:17:06/00:03:15, flags: sT
  Incoming interface: Loopback0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet2/0, Forward/Sparse-Dense, 00:17:06/00:03:08
```

P1A

```
P1A# show ip mroute 232.1.1.1

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

(192.168.0.1, 232.1.1.1), 00:17:43/00:03:08, flags: sT
  Incoming interface: Ethernet2/0, RPF nbr 10.0.3.1
  Outgoing interface list:
    Ethernet3/0, Forward/Sparse-Dense, 00:17:43/00:02:51

(192.168.0.8, 232.1.1.1), 00:18:12/00:03:15, flags: sTV
  Incoming interface: Ethernet3/0, RPF nbr 10.0.4.4, vector 192.168.0.4
  Outgoing interface list:
    Ethernet2/0, Forward/Sparse-Dense, 00:18:12/00:03:15

(192.168.0.2, 232.1.1.1), 00:18:13/00:03:18, flags: sT
  Incoming interface: Ethernet2/0, RPF nbr 10.0.3.2
  Outgoing interface list:
    Ethernet3/0, Forward/Sparse-Dense, 00:18:13/00:03:18
```

ASBR1A

```
ASBR1A# show ip mroute 232.1.1.1

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

(10.254.254.8, 232.1.1.1), 00:20:13/00:03:16, flags: sT
  Incoming interface: Ethernet6/0, RPF nbr 10.0.7.12
  Outgoing interface list:
    Ethernet5/0, Forward/Sparse-Dense, 00:20:13/00:02:46

(10.254.254.2, 232.1.1.1), 00:20:13/00:03:16, flags: sT
  Incoming interface: Ethernet5/0, RPF nbr 10.0.6.5
  Outgoing interface list:
    Ethernet6/0, Forward/Sparse-Dense, 00:20:13/00:02:39
```

ASBR1B

```
ASBR1B# show ip mroute 232.1.1.1

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

(192.168.0.1, 232.1.1.1), 00:37:43/00:03:16, flags: sTV
  Incoming interface: Ethernet4/0, RPF nbr 10.0.5.4, vector 10.0.5.4
  Outgoing interface list:
    Ethernet6/0, Forward/Sparse-Dense, 00:37:43/00:03:10

(192.168.0.8, 232.1.1.1), 00:38:14/00:03:16, flags: sT
  Incoming interface: Ethernet6/0, RPF nbr 10.0.7.8
  Outgoing interface list:
    Ethernet4/0, Forward/Sparse-Dense, 00:38:14/00:02:45

(192.168.0.2, 232.1.1.1), 00:38:14/00:03:16, flags: sTV
  Incoming interface: Ethernet4/0, RPF nbr 10.0.5.4, vector 10.0.5.4
  Outgoing interface list:
    Ethernet6/0, Forward/Sparse-Dense, 00:38:14/00:02:45
```

PE1B

```
PE1B# show ip mroute 232.1.1.1

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

(192.168.0.1, 232.1.1.1), 00:35:23/00:02:40, flags: sTIZV
  Incoming interface: Ethernet6/0, RPF nbr 10.0.7.9, vector 192.168.0.9
  Outgoing interface list:
    MVRF blue, Forward/Sparse-Dense, 00:35:23/00:00:00

(192.168.0.2, 232.1.1.1), 00:35:53/00:02:40, flags: sTIZV
  Incoming interface: Ethernet6/0, RPF nbr 10.0.7.9, vector 192.168.0.9
  Outgoing interface list:
    MVRF blue, Forward/Sparse-Dense, 00:35:53/00:00:00

(192.168.0.8, 232.1.1.1), 00:35:53/00:03:10, flags: sT
  Incoming interface: Loopback0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet6/0, Forward/Sparse-Dense, 00:35:53/00:02:35
```

Configuring the Exchange of VPNv4 Routes Between RRs Using Multihop MP-EBGP Peering Sessions (Option C): Example

The following example shows how to configure support for MVPN inter-AS option C. This configuration is based on the sample inter-AS topology illustrated in Figure 22.

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

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

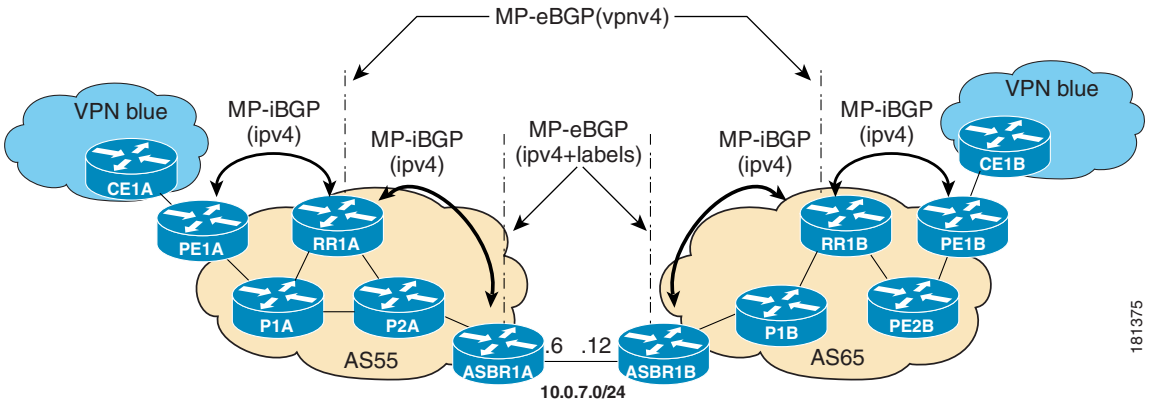


Table 3 provides information about the topology used for this inter-AS MVPN Option C configuration example.

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

PE, RR, or ASBR Router	AS Number	Loopback0 Interfaces	Default MDT (PIM-SSM)
PE1A	55	10.254.254.2/32	232.1.1.1
RR1A	55	10.252.252.4/32	232.1.1.1
ASBR1A	55	10.254.254.6/32	232.1.1.1
PE1B	65	10.254.254.8/32	232.1.1.1
RR1B	65	10.252.252.10/32	232.1.1.1
ASBR1B	65	10.254.254.12/32	232.1.1.1

PE1A

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

```
ip multicast-routing vrf blue
ip multicast rpf proxy vector
!
.
.
.
!
interface Ethernet0/0
 ip vrf forwarding blue
 ip pim sparse-mode
!
.
.
.
!
router bgp 55
 no bgp default route-target filter
 bgp log-neighbor-changes
 neighbor 10.252.252.4 remote-as 55
 neighbor 10.252.252.4 update-source Loopback0
 !
 address-family ipv4
 neighbor 10.252.252.4 activate
 neighbor 10.252.252.4 send-label
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family ipv4 mdt
 neighbor 10.252.252.4 activate
 neighbor 10.252.252.4 send-community extended
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.252.252.4 activate
 neighbor 10.252.252.4 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf blue
 redistribute connected
 redistribute static
 redistribute rip metric 50
 no synchronization
 exit-address-family
!
.
.
.
!
ip pim ssm default
!
```

RR1A

```
!
router bgp 55
 neighbor 10.252.252.10 remote-as 65
 neighbor 10.252.252.10 ebgp-multihop 255
 neighbor 10.252.252.10 update-source Loopback0
 neighbor 10.254.254.2 remote-as 55
 neighbor 10.254.254.2 update-source Loopback0
 neighbor 10.254.254.6 remote-as 55
 neighbor 10.254.254.6 update-source Loopback0
 !
```

```
 address-family ipv4
 no neighbor 10.252.252.10 activate
 neighbor 10.254.254.2 activate
 neighbor 10.254.254.2 route-reflector-client
 neighbor 10.254.254.2 send-label
 neighbor 10.254.254.6 activate
 neighbor 10.254.254.6 route-reflector-client
 neighbor 10.254.254.6 send-label
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family ipv4 mdt
 neighbor 10.252.252.10 activate
 neighbor 10.252.252.10 next-hop-unchanged
 neighbor 10.254.254.2 activate
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.252.252.10 activate
 neighbor 10.252.252.10 send-community extended
 neighbor 10.252.252.10 next-hop-unchanged
 neighbor 10.254.254.2 activate
 neighbor 10.254.254.2 send-community extended
 neighbor 10.254.254.2 route-reflector-client
 exit-address-family
!
```

ASBR1A

```
!
ip multicast-routing
ip multicast-routing vrf blue
!
.
.
.
!
interface Ethernet7/0
 ip vrf forwarding blue
 ip pim sparse-mode
!
.
.
.
!
router bgp 55
 neighbor 10.0.7.12 remote-as 65
 neighbor 10.252.252.4 remote-as 55
 neighbor 10.252.252.4 update-source Loopback0
 !
 address-family ipv4
 redistribute isis level-2 route-map inter-as
 neighbor 10.0.7.12 activate
 neighbor 10.0.7.12 route-map IN in
 neighbor 10.0.7.12 route-map OUT out
 neighbor 10.0.7.12 send-label
 neighbor 10.252.252.4 activate
 neighbor 10.252.252.4 next-hop-self
 neighbor 10.252.252.4 send-label
 no auto-summary
 no synchronization
 exit-address-family
!
```

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

ASBR1B

```
!
ip multicast-routing
ip multicast-routing vrf blue
!
.
.
.
!
interface Ethernet6/0
 ip vrf forwarding blue
 ip pim sparse-mode
!
.
.
.
!
router bgp 65
 neighbor 10.0.7.6 remote-as 55
 neighbor 10.252.252.10 remote-as 65
 neighbor 10.252.252.10 update-source Loopback0
 !
 address-family ipv4
 redistribute isis level-2 route-map inter-as
 neighbor 10.0.7.6 activate
 neighbor 10.0.7.6 route-map IN in
 neighbor 10.0.7.6 route-map OUT out
 neighbor 10.0.7.6 send-label
 neighbor 10.252.252.10 activate
 neighbor 10.252.252.10 next-hop-self
 neighbor 10.252.252.10 send-label
 no auto-summary
 no synchronization
 exit-address-family
!
.
.
.
!
ip pim ssm default
!
```

RR1B

```
!
router bgp 65
 no bgp default route-target filter
 bgp log-neighbor-changes
 neighbor 10.252.252.4 remote-as 55
 neighbor 10.252.252.4 ebgp-multihop 255
 neighbor 10.252.252.4 update-source Loopback0
 neighbor 10.254.254.8 remote-as 65
 neighbor 10.254.254.8 update-source Loopback0
 neighbor 10.254.254.12 remote-as 65
 neighbor 10.254.254.12 update-source Loopback0
 !
```

```
 address-family ipv4
 no neighbor 10.252.252.4 activate
 neighbor 10.254.254.8 activate
 neighbor 10.254.254.8 route-reflector-client
 neighbor 10.254.254.8 send-label
 neighbor 10.254.254.12 activate
 neighbor 10.254.254.12 route-reflector-client
 neighbor 10.254.254.12 send-label
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family ipv4 mdt
 neighbor 10.252.252.4 activate
 neighbor 10.252.252.4 next-hop-unchanged
 neighbor 10.254.254.8 activate
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.252.252.4 activate
 neighbor 10.252.252.4 send-community extended
 neighbor 10.252.252.4 next-hop-unchanged
 neighbor 10.254.254.8 activate
 neighbor 10.254.254.8 send-community extended
 neighbor 10.254.254.8 route-reflector-client
 exit-address-family
!
```

PE1B

```
!
ip vrf blue
 rd 55:1111
 mdt default 232.1.1.1
!
!
ip multicast-routing
ip multicast-routing vrf blue
ip multicast rpf proxy vector
!
.
.
.
!
interface Ethernet12/0
 ip vrf forwarding blue
 ip pim sparse-mode
!
.
.
.
!
router bgp 65
 no bgp default route-target filter
 bgp log-neighbor-changes
 neighbor 10.252.252.10 remote-as 65
 neighbor 10.252.252.10 update-source Loopback0
 !
 address-family ipv4
 neighbor 10.252.252.10 activate
 neighbor 10.252.252.10 send-label
 no auto-summary
 no synchronization
 exit-address-family
```

```
 !
 address-family ipv4 mdt
 neighbor 10.252.252.10 activate
 neighbor 10.252.252.10 send-community extended
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.252.252.10 activate
 neighbor 10.252.252.10 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf blue
 redistribute connected
 redistribute static
 redistribute rip metric 50
 no synchronization
 exit-address-family
!
.
.
.
!
ip pim ssm default
!
```

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

```
PE1A# show ip pim mdt bgp

MDT (Route Distinguisher + IPv4)              Router ID         Next Hop
  MDT group 232.1.1.1
   55:1111:10.254.254.8                       10.252.252.4      10.254.254.8

PE1B# show ip pim mdt bgp

MDT (Route Distinguisher + IPv4)              Router ID         Next Hop
  MDT group 232.1.1.1
   55:1111:10.254.254.2                       10.252.252.10     10.254.254.2
```

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

```
P1A# show ip mroute proxy

(10.254.254.8, 232.1.1.1)
  Proxy                     Assigner         Origin    Uptime/Expire
  10.254.254.6              10.0.2.2         PIM       00:15:37/00:02:57
P2A# show ip mroute proxy

(10.254.254.8, 232.1.1.1)
  Proxy                     Assigner         Origin    Uptime/Expire
  10.254.254.6              10.0.4.3         PIM       00:20:41/00:02:46

P1B# show ip mroute proxy

(10.254.254.2, 232.1.1.1)
  Proxy                     Assigner         Origin    Uptime/Expire
```

```
  10.254.254.12              10.0.10.9        PIM       00:29:38/00:02:16

P2B# show ip mroute proxy

(10.254.254.2, 232.1.1.1)
  Proxy                      Assigner         Origin    Uptime/Expire
  10.254.254.12              10.0.12.8        PIM       00:29:58/00:02:09
```

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

PE1A

```
PE1A# show ip mroute 232.1.1.1

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

(10.254.254.8, 232.1.1.1), 00:12:27/00:02:43, flags: sTIZv
  Incoming interface: Ethernet1/0, RPF nbr 10.0.2.3, vector 10.254.254.6
  Outgoing interface list:
    MVRF blue, Forward/Sparse-Dense, 00:12:27/00:00:00

(10.254.254.2, 232.1.1.1), 00:14:40/00:03:12, flags: sT
  Incoming interface: Loopback0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet1/0, Forward/Sparse-Dense, 00:12:27/00:03:06
```

P1A

```
P1A# show ip mroute 232.1.1.1

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

(10.254.254.2, 232.1.1.1), 00:15:40/00:03:25, flags: sT
  Incoming interface: Ethernet1/0, RPF nbr 10.0.2.2
  Outgoing interface list:
    Ethernet4/0, Forward/Sparse-Dense, 00:15:40/00:03:24

(10.254.254.8, 232.1.1.1), 00:15:40/00:03:25, flags: sTv
```

```
  Incoming interface: Ethernet4/0, RPF nbr 10.0.4.5, vector 10.254.254.6
  Outgoing interface list:
    Ethernet1/0, Forward/Sparse-Dense, 00:15:40/00:03:25
```

P2A

```
P2A# show ip mroute 232.1.1.1

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

(10.254.254.2, 232.1.1.1), 00:20:43/00:03:15, flags: sT
  Incoming interface: Ethernet4/0, RPF nbr 10.0.4.3
  Outgoing interface list:
    Ethernet5/0, Forward/Sparse-Dense, 00:20:43/00:03:15

(10.254.254.8, 232.1.1.1), 00:20:43/00:03:15, flags: sTv
  Incoming interface: Ethernet5/0, RPF nbr 10.0.6.6, vector 10.254.254.6
  Outgoing interface list:
    Ethernet4/0, Forward/Sparse-Dense, 00:20:43/00:03:14
```

ASBR1A

```
ASBR1A# show ip mroute 232.1.1.1

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

(10.254.254.8, 232.1.1.1), 00:20:13/00:03:16, flags: sT
  Incoming interface: Ethernet6/0, RPF nbr 10.0.7.12
  Outgoing interface list:
    Ethernet5/0, Forward/Sparse-Dense, 00:20:13/00:02:46

(10.254.254.2, 232.1.1.1), 00:20:13/00:03:16, flags: sT
  Incoming interface: Ethernet5/0, RPF nbr 10.0.6.5
  Outgoing interface list:
    Ethernet6/0, Forward/Sparse-Dense, 00:20:13/00:02:39
```

ASBR1B

```
ASBR1B# show ip mroute 232.1.1.1

IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
```

```
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(10.254.254.8, 232.1.1.1), 00:25:46/00:03:13, flags: sT
  Incoming interface: Ethernet7/0, RPF nbr 10.0.8.11
  Outgoing interface list:
    Ethernet6/0, Forward/Sparse-Dense, 00:25:46/00:03:04

(10.254.254.2, 232.1.1.1), 00:25:46/00:03:13, flags: sT
  Incoming interface: Ethernet6/0, RPF nbr 10.0.7.6
  Outgoing interface list:
    Ethernet7/0, Forward/Sparse-Dense, 00:25:46/00:03:07
```

P1B

```
P1B# show ip mroute 232.1.1.1

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

(10.254.254.8, 232.1.1.1), 00:29:41/00:03:17, flags: sT
  Incoming interface: Ethernet10/0, RPF nbr 10.0.10.9
  Outgoing interface list:
    Ethernet7/0, Forward/Sparse-Dense, 00:29:41/00:02:56

(10.254.254.2, 232.1.1.1), 00:29:41/00:03:17, flags: sTv
  Incoming interface: Ethernet7/0, RPF nbr 10.0.8.12, vector 10.254.254.12
  Outgoing interface list:
    Ethernet10/0, Forward/Sparse-Dense, 00:29:41/00:02:44
```

P2B

```
P2B# show ip mroute 232.1.1.1

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

```
(10.254.254.8, 232.1.1.1), 00:30:01/00:03:25, flags: sT
  Incoming interface: Ethernet11/0, RPF nbr 10.0.12.8
  Outgoing interface list:
    Ethernet10/0, Forward/Sparse-Dense, 00:30:01/00:02:30

(10.254.254.2, 232.1.1.1), 00:30:01/00:03:25, flags: sTv
  Incoming interface: Ethernet10/0, RPF nbr 10.0.10.11, vector 10.254.254.12
  Outgoing interface list:
    Ethernet11/0, Forward/Sparse-Dense, 00:30:01/00:02:36
```

PE1B

```
PE1B# show ip mroute 232.1.1.1

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

(10.254.254.2, 232.1.1.1), 00:31:22/00:02:55, flags: sTIZv
  Incoming interface: Ethernet11/0, RPF nbr 10.0.12.9, vector 10.254.254.12
  Outgoing interface list:
    MVRF blue, Forward/Sparse-Dense, 00:31:22/00:00:00

(10.254.254.8, 232.1.1.1), 00:33:35/00:03:25, flags: sT
  Incoming interface: Loopback0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet11/0, Forward/Sparse-Dense, 00:31:22/00:03:22
```

Additional References

The following sections provide references related to configuring MVPN Inter-AS Support.

Related Documents

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

Standards

Standard	Title
draft-ietf-pim-rpf-vector-03.txt	*The RPF Vector TLV*
draft-ietf-l3vpn-rfc2547bis-03.txt[1]	*BGP/MPLS IP VPNs*

1. The Internet draft standard draft-ietf-l3vpn-rfc2547bis-03.txt is generally referred to as RFC 2547bis.

MIBs

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

RFCs

RFC	Title
RFC 4364[1]	*BGP/MPLS IP Virtual Private Networks (VPN)*

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

Technical Assistance

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

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IP Multicast Command Reference* at http://www.cisco.com/en/US/docs/ios/ipmulti/command/reference/imc_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html.

- **ip multicast rpf proxy vector**
- **show ip mroute**
- **show ip pim neighbor**
- **show ip rpf**

Feature Information for Configuring Multicast VPN Inter-AS Support

Table 4 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Releases 12.2(1) or 12.0(3)S or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the see the "IP Multicast Features Roadmap."

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 4 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

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

Feature Name	Releases	Feature Information
BGP Multicast Inter-AS VPN	12.0(29)S 12.2(33)SRA 12.2(31)SB2 12.2(33)SXH 12.4(20)T	The BGP Multicast Inter-AS VPN feature introduces the IPv4 MDT SAFI in BGP. The MDT SAFI is a transitive multicast capable connector attribute that is defined as an IPv4 address family in BGP. The MDT SAFI is designed to support inter-AS VPN peering sessions. The following sections provide information about this feature: • MVPN Inter-AS Support Solution for Options B and C, page 5 • BGP Connector Attribute, page 5 • BGP MDT SAFI Updates for MVPN Inter-AS Support, page 6 • Configuring the MDT Address Family in BGP for Multicast VPN Inter-AS Support, page 24 • Displaying Information about IPv4 MDT Sessions in BGP, page 26 • Clearing IPv4 MDT Peering Sessions in BGP, page 26 • Configuring an IPv4 MDT Address-Family Session for Multicast VPN Inter-AS Support: Example, page 34 The following commands were introduced or modified by this feature: **address-family ipv4** (BGP), **clear bgp ipv4 mdt**, **show ip bgp ipv4**.

Table 4 Feature Information for Configuring Multicast VPN Inter-AS Support (continued)

Feature Name	Releases	Feature Information
Multicast VPN Inter-AS Support	12.0(30)S 12.2(33)SRA 12.2(31)SB2 12.2(33)SXH 12.4(20)T	The Multicast VPN Inter-AS support feature enables MDTs used for MVPNs to span multiple autonomous systems. Benefits include increased multicast coverage to customers that require multicast to span multiple service providers in an MPLS Layer 3 VPN service with the flexibility to support all options described in RFC 4364. Additionally, the Multicast VPN Inter-AS Support feature may be used to consolidate an existing MVPN service with another MVPN service, such as the case with a company merger or acquisition. The following sections provide information about this feature: • MVPN Inter-AS Support Overview, page 2 • Benefits of MVPN Inter-AS Support, page 3 • MVPN Inter-AS Support Implementation Requirements, page 3 • MVPN Inter-AS Support Solution for Options B and C, page 5 • Configuring a PE Router to Send BGP MDT Updates to Build the Default MDT for MVPN Inter-AS Support (Option B): Example, page 34 • Configuring a PE Router to Send BGP MDT Updates to Build the Default MDT for MVPN Inter-AS Support (Option C): Example, page 35 The following commands were introduced or modified by this feature: **ip multicast rpf proxy vector**, and **show ip mroute**, **show ip pim neighbor**, **show ip rpf**.
PIM RPF Vector	12.0(30)S 12.2(33)SRA 12.2(31)SB2 12.2(33)SXH 12.4(20)T	The PIM RPF Vector feature enables core routers to perform RPF checks on an IP address of the exit router instead of on the source router. The address on the exit router is the RPF Vector and it is inserted in PIM join messages. The following sections provide information about this feature: • PIM RPF Vector, page 7 • MVPN Inter-AS MDT Establishment for Option B, page 9 • MVPN Inter-AS MDT Establishment for Option C, page 17 The following commands were introduced or modified by this feature: **ip multicast rpf proxy vector**, **show ip mroute**, **show ip pim neighbor**.

Tunneling to Connect Non-IP Multicast Areas

This module describes how to configure a Generic Route Encapsulation (GRE) tunnel to tunnel IP multicast packets between non-IP multicast areas. The benefit is that IP multicast traffic can be sent from a source to a multicast group, over an area where IP multicast is not supported.

Module History

This module was first published on May 2, 2005, and last updated on May 2, 2005.

Finding Feature Information in This Document

Not all features may be supported in your Cisco IOS software release. Use the "Feature Information for Tunneling to Connect Non-IP Multicast Areas" to find information about feature support and configuration.

Contents

Prerequisites for Tunneling to Connect Non-IP Multicast Areas

This module assumes you understand the concepts in the "IP Multicast Technology Overview" module.

Information About Tunneling to Connect Non-IP Multicast Areas

Before connecting non-IP multicast areas, you should understand the following concepts:

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Benefits of Tunneling to Connect Non-IP Multicast Areas

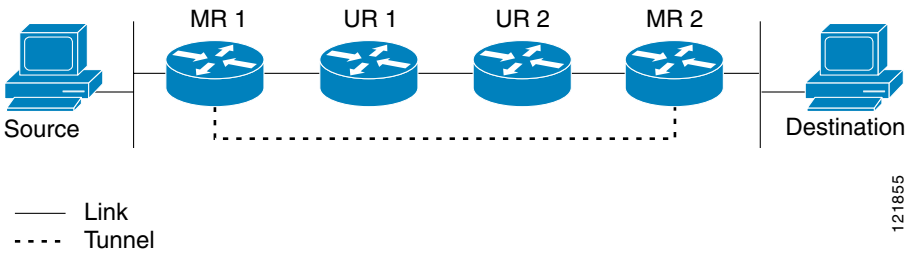
- If the path between a source and a group member (destination) does not support IP multicast, a tunnel between them can transport IP multicast packets.
- Per packet load balancing can be used. Load balancing in IP multicast is normally per (S,G). Therefore, (S1, G) can go over Link X and (S2, G) can go over Link Y, where X and Y are parallel links. If you create a tunnel between the routers, you can get per packet load balancing because the load balancing is done on the tunnel unicast packets.

IP Multicast Static Route (mroute)

IP multicast static routes (mroutes) allow you to have multicast paths diverge from the unicast paths. When using Protocol Independent Multicast (PIM), the router expects to receive packets on the same interface where it sends unicast packets back to the source. This expectation is beneficial if your multicast and unicast topologies are congruent. However, you might want unicast packets to take one path and multicast packets to take another.

The most common reason for using separate unicast and multicast paths is tunneling. When a path between a source and a destination does not support multicast routing, a solution is to configure two routers with a GRE tunnel between them. In Figure 1, each unicast router (UR) supports unicast packets only; each multicast router (MR) supports multicast packets.

Figure 1 Tunnel for Multicast Packets



In Figure 1, Source delivers multicast packets to Destination by using MR 1 and MR 2. MR 2 accepts the multicast packet only if it believes it can reach Source over the tunnel. If this situation is true, when Destination sends unicast packets to Source, MR 2 sends them over the tunnel. The check that MR2 can reach Source over the tunnel is a Reverse Path Forwarding (RPF) check, and the static mroute allows the check to be successful when the interface that the multicast packet arrives on is not the unicast path back to the source. Sending the packet over the tunnel could be slower than natively sending it through UR 2, UR 1, and MR 1.

A multicast static route allows you to use the configuration in Figure 1 by configuring a static multicast source. The system uses the configuration information instead of the unicast routing table to route the traffic. Therefore, multicast packets can use the tunnel without having unicast packets use the tunnel. Static mroutes are local to the router they are configured on and not advertised or redistributed in any way to any other router.

How to Connect Non-IP Multicast Areas

This section contains the following procedure:

Configuring a Tunnel to Connect Non-IP Multicast Areas

Configure a multicast static route if you want your multicast paths to differ from your unicast paths. For example, you might have a tunnel between two routers because the unicast path between a source and destination does not support multicast routing.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ip unnumbered** *type number*
5. **ip pim sparse-mode**
6. **tunnel source** {*ip-address* | *type number*}
7. **tunnel destination** {*hostname* | *ip-address*}
8. Repeat Steps 1 through 7 on the router at the opposite end of the tunnel, reversing the tunnel source and destination addresses.
9. **end**
10. **ip mroute** *source-address mask* **tunnel** *number* [*distance*]
11. **ip mroute** *source-address mask* **tunnel** *number* [*distance*]
12. **end**
13. **show ip mroute** [*group-address* | *group-name*] [*source-address* | *source-name*] [*interface-type interface-number*] [**summary**] [**count**] [**active** *kbps*]
14. **show ip rpf** {*source-address* | *source-name*} [**metric**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 0`	Configures a tunnel interface.
Step 4	**ip unnumbered** *type number* **Example:** `Router(config-if)# ip unnumbered ethernet 0`	Enables IP processing without assigning an IP address to the interface.
Step 5	**ip pim sparse-mode** **Example:** `Router(config-if)# ip pim sparse-mode`	Enables PIM sparse mode on the tunnel interface.
Step 6	**tunnel source** {*ip-address* | *type number*} **Example:** `Router(config-if)# tunnel source 100.1.1.1`	Configures the tunnel source.
Step 7	**tunnel destination** {*hostname* | *ip-address*} **Example:** `Router(config-if)# tunnel destination 100.1.5.3`	Configures the tunnel destination.
Step 8	Repeat Steps 1 through 7 on the router at the opposite end of the tunnel, reversing the tunnel source and destination addresses.	Router A's tunnel source address will match Router B's tunnel destination address. Router A's tunnel destination address will match Router B's tunnel source address.
Step 9	**end** **Example:** `Router(config-if)# end`	Ends the current configuration session and returns to privileged EXEC mode.
Step 10	**ip mroute** *source-address mask* **tunnel** *number* [*distance*] **Example:** `Router(config)# ip mroute 0.0.0.0 0.0.0.0 tunnel 0`	Configures a static multicast route over which to reverse path forward to the other end of the tunnel. • Because the use of the tunnel makes the multicast topology incongruent with the unicast topology, and only multicast traffic traverses the tunnel, you must configure the routers to reverse path forward correctly over the tunnel. • When a source range is specified, the mroute applies only to those sources. • In the example, the *source-address* and *mask* of 0.0.0.0 0.0.0.0 indicate any address. • The shorter distance is preferred. • The default distance is 0.

	Command or Action	Purpose
Step 11	**ip mroute** *source-address mask* **tunnel** *number* [*distance*] **Example:** `Router(config)# ip mroute 0.0.0.0 0.0.0.0 tunnel 0`	Configures a static route over which to reverse path forward from the access router to the other end of the tunnel.
Step 12	**end** **Example:** `Router(config)# end`	(Optional) Ends the current configuration session and returns to privileged EXEC mode.
Step 13	**show ip mroute** [*group-address* | *group-name*] [*source-address* | *source-name*] [*interface-type interface-number*] [**summary**] [**count**] [**active** *kbps*] **Example:** `Router# show ip mroute`	(Optional) Displays the contents of the IP multicast routing (mroute) table.
Step 14	**show ip rpf** {*source-address* | *source-name*} [**metric**] **Example:** `Router# show ip rpf 10.2.3.4`	(Optional) Displays how IP multicast routing does RPF.

Configuration Examples for Tunneling to Connect Non-IP Multicast Areas

This section provides the following configuration example:

- Tunneling to Connect Non-IP Multicast Areas: Example, page 5

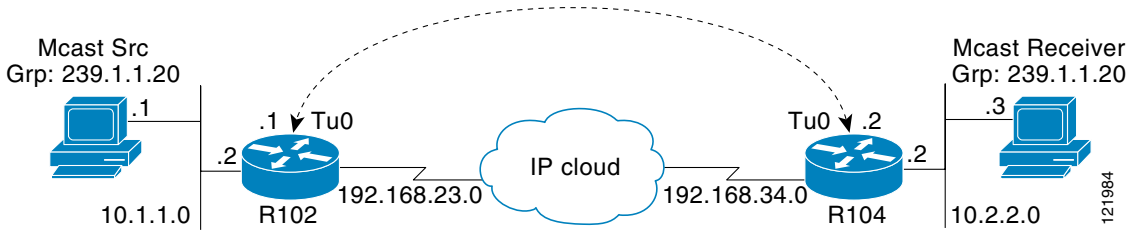
Tunneling to Connect Non-IP Multicast Areas: Example

The following example also appears online at:

http://www.cisco.com/en/US/tech/tk828/tk363/technologies_configuration_example09186a00801a5aa2.shtml

In Figure 2, the multicast source (10.1.1.1) is connected to R102 and is configured for multicast group 239.1.1.20. The multicast receiver (10.2.2.3) is connected to R104 and is configured to receive multicast packets for group 239.1.1.20. Separating R102 and R104 is an IP cloud, which is not configured for multicast routing.

Figure 2 Tunnel Connecting Non-IP Multicast Areas



A tunnel is configured between R102 to R104 sourced with their loopback interfaces. The **ip pim sparse-dense-mode** command is configured on tunnel interfaces and multicast-routing is enabled on R102 and R104. Sparse-dense mode configuration on the tunnel interfaces allows sparse-mode or dense-mode packets to be forwarded over the tunnel depending on rendezvous point (RP) configuration for the group.

Note For dense mode—With PIM dense mode configured over the tunnel, an **ip mroute 10.1.1.0 255.255.255.0 tunnel 0** command is configured on R104 to ensure a successful RPF for multicast source address 10.1.1.1. Incoming (10.1.1.1, 239.1.1.20) multicast packets over Tunnel0 (Tu0) are checked for Reverse Path Forwarding (RPF) using this mroute statement. After a successful check, the multicast packets are forwarded to outgoing interface list (OIL) interfaces.

Note For sparse mode—With PIM sparse mode configured over the tunnel, ensure that the following points are addressed:

- For a successful RPF verification of multicast traffic flowing over the shared tree (*,G) from RP, an **ip mroute rp-address nexthop** command needs to be configured for the RP address, pointing to the tunnel interface.

  Assuming R102 to be the RP (RP address 2.2.2.2) in this case, the mroute would be the **ip mroute 2.2.2.2 255.255.255.255 tunnel 0** command, which ensures a successful RPF check for traffic flowing over the shared tree.

- For a successful RPF verification of multicast (S,G) traffic flowing over the Shortest Path Tree (SPT), an **ip mroute source-address nexthop** command needs to be configured for the multicast source, pointing to the tunnel interface.

  In this case, when SPT traffic is flowing over tunnel interface an **ip mroute 10.1.1.0 255.255.255.0 tunnel 0** command is configured on R104 to ensure a successful RPF verification for incoming (10.1.1.1, 239.1.1.20) multicast packets over the Tunnel 0 interface.

R102#

```
version 12.2
hostname r102
ip subnet-zero
no ip domain-lookup
!--- It stops IP domain lookup, which improves the show command response time.
!
ip multicast-routing
!--- Enables IP multicast routing.
!
interface Loopback0
 ip address 2.2.2.2 255.255.255.255
```

```
!--- Tunnel Source interface.
!
interface Tunnel0
!--- Tunnel interface configured for PIM and carrying multicast packets to R104.
 ip address 192.168.24.1 255.255.255.252
 ip pim sparse-dense-mode
 tunnel source Loopback0
 tunnel destination 4.4.4.4
!
interface Ethernet0/0
!--- Interface connected to Source.
ip address 10.1.1.2 255.255.255.0
 ip pim sparse-dense-mode
!
interface Serial8/0
 ip address 192.168.23.1 255.255.255.252
!--- Note IP PIM sparse-dense mode is not configured on Serial interface.
!
router ospf 1
 log-adjacency-changes
 network 2.2.2.2 0.0.0.0 area 0
 network 10.1.1.0 0.0.0.255 area 0
 network 192.168.23.0 0.0.0.255 area 0
!
ip classless
ip pim bidir-enable
!
line con 0
line aux 0
line vty 0 4
 login
!
end
```

R104#

```
version 12.2
!
hostname r104
!
ip subnet-zero
no ip domain-lookup
!--- It stops IP domain lookup, which improves the show command response time.
!
ip multicast-routing
!--- Enables IP multicast routing.
!
interface Loopback0
 ip address 4.4.4.4 255.255.255.255
!--- Tunnel Source interface.
!
interface Tunnel0
 ip address 192.168.24.2 255.255.255.252
!--- Tunnel interface configured for PIM and carrying multicast packets.
ip pim sparse-dense-mode
 tunnel source Loopback0
 tunnel destination 2.2.2.2
!
interface Ethernet0/0
 ip address 10.2.2.2 255.255.255.0
 ip pim sparse-dense-mode
!
interface Serial9/0
 ip address 192.168.34.1 255.255.255.252
```

```
!--- Note IP PIM sparse-dense mode is not configured on Serial interface.
!
!
router ospf 1
 log-adjacency-changes
 network 4.4.4.4 0.0.0.0 area 0
 network 10.2.2.0 0.0.0.255 area 0
 network 192.168.34.0 0.0.0.255 area 0
!
ip classless
no ip http server
ip pim bidir-enable
ip mroute 10.1.1.0 255.255.255.0 Tunnel0
!--- This Mroute ensures a successful RPF check for packets flowing from the source.
!--- 10.1.1.1 over Shared tree in case of Dense more and SPT in case of Sparse mode.
!
ip mroute 2.2.2.2 255.255.255.255 tunnel 0
!--- This Mroute is required for RPF check when Sparse mode multicast traffic is
!--- flowing from RP (assuming R102 with 2.2.2.2 as RP)  towards receiver via tunnel
!--- before the SPT switchover.
line con 0
line aux 0
line vty 0 4
 login
!
end
```

Additional References

The following sections provide references related to tunneling to connect non-IP multicast areas.

Related Documents

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

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 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/public/support/tac/home.shtml

Feature Information for Tunneling to Connect Non-IP Multicast Areas

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Releases 12.2(1) or later appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for specific commands was introduced, see the command reference documents.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator (http://www.cisco.com/go/fn). You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click Cancel at the login dialog box and follow the instructions that appear.

Table 1 Feature Information for Tunneling to Connect Non-IP Multicast Areas

Feature Name	Releases	Feature Configuration Information
This table is intentionally left blank because no features were introduced or modified in Cisco IOS Release 12.2(1) or later. This table will be updated when feature information is added to this module.	—	—

Configuring IP Multicast over ATM Point-to-Multipoint VCs

This module describes how to configure IP multicast over ATM point-to-multipoint virtual circuits (VCs). This feature dynamically creates ATM point-to-multipoint switched virtual circuits (SVCs) to handle IP multicast traffic more efficiently. It can enhance router performance and link utilization because packets are not replicated and sent multiple times over the ATM interface.

Module History

This module was first published on May 2, 2005, and last updated on May 2, 2005.

Finding Feature Information in This Module

Not all features may be supported in your Cisco IOS software release. Use the "Feature Information for Configuring IP Multicast over ATM Point-to-Multipoint VCs" to find information about feature support and configuration.

Contents

Prerequisites for IP Multicast over ATM Point-to-Multipoint VCs

- You must have IP multicast routing and PIM sparse mode configured. This feature does not work with PIM dense mode.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

- You must have ATM configured for multipoint signaling. Refer to the "Configuring ATM" chapter in the *Cisco IOS Wide-Area Networking Configuration Guide* for more information on how to configure ATM for point-to-multipoint signaling.
- You should understand the concepts in the "IP Multicast Technology Overview" module.

Information About IP Multicast over ATM Point-to-Multipoint VCs

Before you configure IP multicast over ATM point-to-multipoint virtual circuits, you should understand the following concepts:

PIM Nonbroadcast Multiaccess

Protocol Independent Multicast (PIM) nonbroadcast multiaccess (NBMA) mode allows the software to replicate packets for each neighbor on the NBMA network. Traditionally, the software replicates multicast and broadcast packets to all broadcast configured neighbors. This action might be inefficient when not all neighbors want packets for certain multicast groups. NBMA mode enables you to reduce bandwidth on links leading into the NBMA network, and to reduce the number of CPU cycles in switches and attached neighbors.

It is appropriate to configure PIM NBMA mode on ATM, Frame Relay, Switched Multimegabit Data Service (SMDS), PRI ISDN, or X.25 networks only, especially when these media do not have native multicast available. Do not use PIM NBMA mode on multicast-capable LANs (such as Ethernet or FDDI).

You should use PIM sparse mode with this feature. Therefore, when each Join message is received from NBMA neighbors, PIM stores each neighbor IP address and interface in the outgoing interface list for the group. When a packet is destined for the group, the software replicates the packet and unicasts (data-link unicasts) it to each neighbor that has joined the group.

Consider the following two factors before enabling PIM NBMA mode:

- If the number of neighbors grows, the outgoing interface list gets large, which costs memory and replication time.
- If the network (Frame Relay, SMDS, or ATM) supports multicast natively, you should use it so that replication is performed at optimal points in the network.

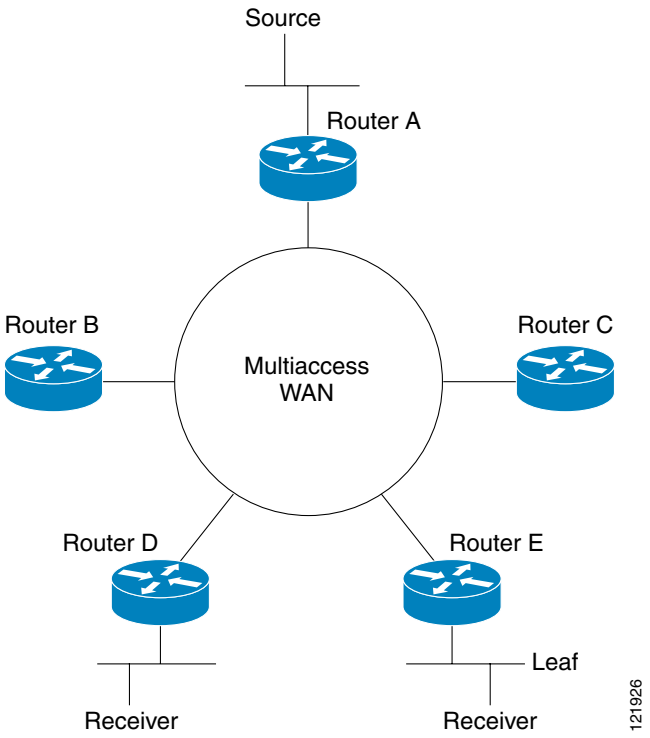
IP Multicast over ATM Point-to-Multipoint VCs

IP Multicast over ATM Point-to-Multipoint VCs is a feature that dynamically creates ATM point-to-multipoint switched virtual circuits (SVCs) to handle IP multicast traffic more efficiently.

This feature can enhance router performance and link utilization because packets are not replicated and sent multiple times over the ATM interface.

Traditionally, over NBMA networks, Cisco routers would perform a pseudobroadcast to get broadcast or multicast packets to all neighbors on a multiaccess network. For example, assume in Figure 1 that Routers A, B, C, D, and E were running the Open Shortest Path First (OSPF) protocol. Router A must deliver to Routers D and E. When Router A sends an OSPF Hello packet, the data link layer replicates the Hello packet and sends one to each neighbor (this procedure is known as pseudobroadcast), which results in four copies being sent over the link from Router A to the multiaccess WAN.

Figure 1 Environment for IP Multicast over ATM Point-to-Multipoint VCs



With the advent of IP multicast, where high-rate multicast traffic can occur, the pseudobroadcast approach does not scale. Furthermore, in the preceding example, Routers B and C would get data traffic they do not need. To handle this problem, PIM can be configured in NBMA mode using the **ip pim nbma-mode** command. PIM in NBMA mode works only for sparse mode groups. Configuring PIM in NBMA mode would allow only Routers D and E to get the traffic without distributing to Routers B and C. However, two copies are still delivered over the link from Router A to the multiaccess WAN.

If the underlying network supported multicast capability, the routers could handle this situation more efficiently. If the multiaccess WAN were an ATM network, IP multicast could use multipoint VCs.

To configure IP multicast using multipoint VCs, Routers A, B, C, D, and E in Figure 1 must run PIM sparse mode. If the Receiver directly connected to Router D joins a group and Router A is the PIM RP, the following sequence of events occurs:

1. Router D sends a PIM Join message to Router A.
2. When Router A receives the PIM join, it sets up a multipoint VC for the multicast group.
3. Later, when the Receiver directly connected to Router E joins the same group, Router E sends a PIM Join message to Router A.

4. Router A will see there is a multipoint VC already associated with the group, and will add Router E to the existing multipoint VC.
5. When the Source sends a data packet, Router A can send a single packet over its link that gets to both Router D and Router E. The replication occurs in the ATM switches at the topological diverging point from Router A to Router D and Router E.

If a host sends an IGMP report over an ATM interface to a router, the router adds the host to the multipoint VC for the group.

This feature can also be used over ATM subinterfaces.

Idling Policy for ATM VCs Created by PIM

An idling policy uses the **ip pim vc-count** command to limit the number of VCs created by PIM. When the router stays at or below the number configured, no idling policy is in effect. When the next VC to be opened will exceed the value, an idling policy is exercised. An idled VC does not mean that the multicast traffic is not forwarded; the traffic is switched to VC 0. VC 0 is the broadcast VC that is open to all neighbors listed in the map list. The name VC 0 is unique to PIM and the mrouting table.

How the Idling Policy Works

The idling policy works as follows:

- The only VCs eligible for idling are those with a current 1-second activity rate less than or equal to the value configured by the **ip pim minimum-vc-rate** interface configuration command on the ATM interface. Activity level is measured in packets per second (pps).
- The VC with the least amount of activity below the configured **ip pim minimum-vc-rate** pps rate is idled.
- If the **ip pim minimum-vc-rate** command is not configured, all VCs are eligible for idling.
- If other VCs are at the same activity level, the VC with the highest fanout (number of leaf routers on the multipoint VC) is idled.
- The activity level is rounded to three orders of magnitude (less than 10 pps, 10 to 100 pps, and 100 to 1000 pps). Therefore, a VC that has 40 pps activity and another that has 60 pps activity are considered to have the same rate, and the fanout count determines which one is idled. If the first VC has a fanout of 5 and the second has a fanout of 3, the first one is idled.
- Idling a VC means releasing the multipoint VC that is dedicated for the multicast group. The traffic of the group continues to be sent; it is moved to the static map VC. Packets will flow over a shared multipoint VC that delivers packets to all PIM neighbors.
- If all VCs have a 1-minute rate greater than the pps value, the new group (that exceeded the **ip pim vc-count** *number*) will use the shared multipoint VC.

Keeping VCs from Idling

By default, all VCs are eligible for idling. You can configure a minimum rate required to keep VCs from being idled.

How to Configure IP Multicast over ATM Point-to-Multipoint VCs

This section contains the following procedure:

Configuring IP Multicast over ATM Point-to-Multipoint VCs

Perform this task to configure IP multicast over ATM point-to-multipoint VCs. All of the steps in the task can be used in an ATM network. This feature can also be used over ATM subinterfaces. PIM NBMA mode could be used in an ATM, Frame Relay, SMDS, PRI ISDN, or X.25 network.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *number*
4. **ip pim nbma-mode**
5. **ip pim multipoint-signalling**
6. **atm multipoint-signalling**
7. **ip pim vc-count** *number*
8. **ip pim minimum-vc-rate** *pps*
9. **show ip pim vc**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface atm** *number* **Example:** `Router(config)# interface atm 0`	Configures an ATM interface.

	Command or Action	Purpose
Step 4	**ip pim nbma-mode** **Example:** `Router(config-if)# ip pim nbma-mode`	(Optional) Enables NBMA mode on a serial link.
Step 5	**ip pim multipoint-signalling** **Example:** `Router(config-if)# ip pim multipoint-signalling`	Enables IP multicast over ATM point-to-multipoint VCs. • This command enables PIM to open ATM point-to-multipoint VCs for each multicast group that a receiver joins.
Step 6	**atm multipoint-signalling** **Example:** `Router(config-if)# atm multipoint-signalling`	Enables point-to-multipoint signaling to the ATM switch. • This command is required so that static map multipoint VCs can be opened. The router uses existing static map entries that include the **broadcast** keyword to establish multipoint calls. The map list is needed because it acts like a static ARP table.
Step 7	**ip pim vc-count** *number* **Example:** `Router(config-if)# ip pim vc-count 300`	(Optional) Changes the maximum number of VCs that PIM can open. • By default, PIM can open a maximum of 200 VCs. When the router reaches this number, it deletes inactive VCs so it can open VCs for new groups that might have activity.
Step 8	**ip pim minimum-vc-rate** *pps* **Example:** `Router(config-if)# ip pim minimum-vc-rate 1500`	(Optional) Sets the minimum activity rate required to keep VCs from being idled. • By default, all VCs are eligible for idling.
Step 9	**show ip pim vc** **Example:** `Router# show ip pim vc`	(Optional) Displays ATM VC status information for multipoint VCs opened by PIM.

Configuration Examples for IP Multicast over ATM Point-to-Multipoint VCs

This section provides the following configuration example:

- IP Multicast over ATM Point-to-Multipoint VCs: Example, page 6

IP Multicast over ATM Point-to-Multipoint VCs: Example

The following example shows how to enable IP multicast over ATM point-to-multipoint VCs:

```
interface ATM2/0
ip address 171.69.214.43 255.255.255.248
 ip pim sparse-mode
 ip pim multipoint-signalling
 ip ospf network broadcast
 atm nsap-address 47.00918100000000410B0A1981.333333333333.00
```

```
 atm pvc 1 0 5 qsaal
 atm pvc 2 0 16 ilmi
 atm multipoint-signalling
 map-group mpvc
router ospf 9
 network 171.69.214.0 0.0.0.255 area 0
!
ip classless
 ip pim rp-address 171.69.10.13 98
!
map-list mpvc
 ip 171.69.214.41 atm-nsap 47.00918100000000410B0A1981.111111111111.00 broadcast
 ip 171.69.214.42 atm-nsap 47.00918100000000410B0A1981.222222222222.00 broadcast
 ip 171.69.214.43 atm-nsap 47.00918100000000410B0A1981.333333333333.00 broadcast
```

Additional References

The following sections provide references related to configuring IP multicast over ATM point-to-multipoint VCs.

Related Documents

Related Topic	Document Title
IP multicast commands	*Cisco IOS IP Multicast Command Reference*
Configuring ATM for point-to-multipoint signaling	"Configuring ATM"

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 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/public/support/tac/home.shtml

Feature Information for Configuring IP Multicast over ATM Point-to-Multipoint VCs

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Releases 12.2(1) or 12.0(3)S or later appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for specific commands was introduced, see the command reference documents.

If you are looking for information on a feature in this technology that is not documented here, see the "IP Multicast Features Roadmap".

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator (http://www.cisco.com/go/fn). You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click Cancel at the login dialog box and follow the instructions that appear.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for IP Multicast over ATM Point-to-Multipoint VCs

Feature Name	Releases	Feature Configuration Information
This table is intentionally left blank because no features were introduced or modified in Cisco IOS Release 12.2(1) or later. This table will be updated when feature information is added to this module.	—	—

Monitoring and Maintaining IP Multicast

This module describes many ways to monitor and maintain an IP multicast network, such as

- displaying which neighboring multicast routers are peering with the local router
- displaying multicast packet rates and loss information
- tracing the path from a source to a destination branch for a multicast distribution tree
- displaying the contents of the IP multicast routing table, information about interfaces configured for PIM, the PIM neighbors discovered by the router, and contents of the IP fast-switching cache
- clearing caches, tables, and databases
- monitoring the delivery of IP multicast packets and being alerted if the delivery fails to meet certain parameters (IP multicast heartbeat)
- using session description and announcement protocols and applications to assist the advertisement of multicast multimedia conferences and other multicast sessions and communicating the relevant session setup information to prospective participants (SAP listener support)
- storing IP multicast packet headers in a cache and displaying them to find out information such as who is sending IP multicast packets to what groups and any multicast forwarding loops in your network
- using managed objects to remotely monitor and configure PIM using Simple Network Management Protocol (SNMP)
- disabling fast switching of IP multicast in order to log debug messages

Module History

This module was first published on May 2, 2005, and last updated on May 2, 2005.

Finding Feature Information in This Module

Not all features may be supported in your Cisco IOS software release. Use the "Feature Information for Monitoring and Maintaining IP Multicast" section on page 19 to find information about feature support and configuration.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Monitoring and Maintaining IP Multicast

- Before performing the tasks in this module, you should be familiar with the concepts described in the "IP Multicast Technology Overview" module.
- You must also have enabled IP multicast and have Protocol Independent Multicast (PIM) configured and running on your network. Refer to the "Configuring Basic IP Multicast" module.

How to Monitor and Maintain IP Multicast

This section contains the following procedures:

- Displaying Multicast Peers, Packet Rates, and Loss Information, and Tracing a Path, page 2 (optional)
- Displaying IP Multicast System and Network Statistics, page 4 (optional)
- Clearing IP Multicast Routing Table or Caches, page 8 (optional)
- Monitoring IP Multicast Delivery Using IP Multicast Heartbeat, page 9 (optional)
- Advertising Multicast Multimedia Sessions Using SAP Listener, page 11 (optional)
- Storing IP Multicast Headers, page 13 (optional)
- Disabling Fast Switching of IP Multicast, page 15 (optional)
- Enabling PIM MIB Extensions for IP Multicast, page 16 (optional)

Displaying Multicast Peers, Packet Rates, and Loss Information, and Tracing a Path

Monitor IP multicast routing when you want to know which neighboring multicast routers are peering with the local router, what the multicast packet rates and loss information are, or when you want to trace the path from a source to a destination branch for a multicast distribution tree.

SUMMARY STEPS

1. **enable**
2. **mrinfo** [*host-name* | *host-address*] [*source-address* | *interface*]
3. **mstat** {*source-name* | *source-address*} [*destination-name* | *destination-address*] [*group-name* | *group-address*]
4. **mtrace** {*source-name* | *source-address*} [*destination-name* | *destination-address*] [*group-name* | *group-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**mrinfo** [*host-name* | *host-address*] [*source-address* | *interface*] **Example:** `Router# mrinfo`	(Optional) Queries which neighboring multicast routers are "peering" with the local router.
Step 3	**mstat** {*source-name* | *source-address*} [*destination-name* | *destination-address*] [*group-name* | *group-address*] **Example:** `Router# mstat allsource`	(Optional) Displays IP multicast packet rate and loss information.
Step 4	**mtrace** {*source-name* | *source-address*} [*destination-name* | *destination-address*] [*group-name* | *group-address*] **Example:** `Router# mtrace allsource`	(Optional) Traces the path from a source to a destination branch for a multicast distribution tree.

Examples

The following is sample output from the **mrinfo** command:

```
Router# mrinfo

192.31.7.37 (labs-allcompany) [version cisco 12.3] [flags: PMSA]:
192.31.7.37 -> 192.31.7.34 (lab-southwest) [1/0/pim]
192.31.7.37 -> 192.31.7.47 (lab-northwest) [1/0/pim]
192.31.7.37 -> 192.31.7.44 (lab-southeast) [1/0/pim]
131.119.26.10 -> 131.119.26.9 (lab-northeast) [1/32/pim]
```

The following is sample output from the **mstat** command in user EXEC mode:

```
Router> mstat labs-in-china 172.16.0.1 224.0.255.255

Type escape sequence to abort.
Mtrace from 172.16.0.0 to 172.16.0.10 via group 224.0.255.255
>From source (labs-in-china) to destination (labs-in-africa)
Waiting to accumulate statistics......
Results after 10 seconds:
Source Response Dest Packet Statistics For Only For Traffic
172.16.0.0      172.16.0.10 All Multicast Traffic From 172.16.0.0
| __/ rtt 48 ms Lost/Sent = Pct Rate To 224.0.255.255
v / hop 48 ms ---------------------- --------------------
172.16.0.1      labs-in-england
| ^ ttl 1
v | hop 31 ms 0/12 = 0% 1 pps 0/1 = --% 0 pps
172.16.0.2
```

```
172.16.0.3  infolabs.com
| ^ ttl 2
v | hop -17 ms -735/12 = --% 1 pps 0/1 = --% 0 pps
172.16.0.4
172.16.0.5      infolabs2.com
| ^ ttl 3
v | hop -21 ms -678/23 = --% 2 pps 0/1 = --% 0 pps
172.16.0.6
172.16.0.7      infolabs3.com
| ^ ttl 4
v | hop 5 ms 605/639 = 95% 63 pps 1/1 = --% 0 pps
172.16.0.8
172.16.0.9      infolabs.cisco.com
| \__ ttl 5
v \ hop 0 ms 4 0 pps 0 0 pps
172.16.0.0      172.16.0.10
Receiver Query Source
```

The following is sample output from the **mtrace** command in user EXEC mode:

```
Router> mtrace 172.16.0.0 172.16.0.10 239.254.254.254

Type escape sequence to abort.
Mtrace from 172.16.0.0 to 172.16.0.10 via group 239.254.254.254
From source (?) to destination (?)
Querying full reverse path...
0 172.16.0.10
-1 172.16.0.8 PIM thresh^ 0 0 ms
-2 172.16.0.6 PIM thresh^ 0 2 ms
-3 172.16.0.5 PIM thresh^ 0 894 ms
-4 172.16.0.3 PIM thresh^ 0 893 ms
-5 172.16.0.2 PIM thresh^ 0 894 ms
-6 172.16.0.1 PIM thresh^ 0 893 ms
```

Displaying IP Multicast System and Network Statistics

Display IP multicast system statistics to show the contents of the IP multicast routing table, information about interfaces configured for PIM, the PIM neighbors discovered by the router, contents of the IP fast-switching cache, and the contents of the circular cache header buffer.

SUMMARY STEPS

1. **enable**
2. **ping** [*group-name* | *group-address*]
3. **show ip mroute** [*group-address* | *group-name*] [*source-address* | *source-name*] [*type number*] [**summary**] [**count**] [**active** *kbps*]
4. **show ip pim interface** [*type number*] [**df** | **count**] [*rp-address*] [**detail**]
5. **show ip pim neighbor** [*type number*]
6. **show ip mcache** [*group-address* | *group-name*] [*source-address* | *source-name*]
7. **show ip mpacket** [*group-address* | *group-name*] [*source-address* | *source-name*] [**detail**]
8. **show ip pim rp** [**mapping** | **metric**] [*rp-address*]
9. **show ip rpf** {*source-address* | *source-name*} [**metric**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**ping** [*group-name* | *group-address*] **Example:** `Router# ping cbone-audio`	(Optional) Sends an ICMP echo request message to a multicast group address or group name.
Step 3	**show ip mroute** [*group-address* | *group-name*] [*source-address* | *source-name*] [*type number*] [**summary**] [**count**] [**active** *kbps*] **Example:** `Router# show ip mroute cbone-audio`	(Optional) Displays the contents of the IP multicast routing table.
Step 4	**show ip pim interface** [*type number*] [**df** | **count**] [*rp-address*] [**detail**] **Example:** `Router# show ip pim interface ethernet1/0 detail`	(Optional) Displays information about interfaces configured for PIM.
Step 5	**show ip pim neighbor** [*type number*] **Example:** `Router# show ip pim neighbor`	(Optional) Lists the PIM neighbors discovered by the router.
Step 6	**show ip mcache** [*group-address* | *group-name*] [*source-address* | *source-name*] **Example:** `Router# show ip mcache`	(Optional) Displays the contents of the IP fast-switching cache.
Step 7	**show ip mpacket** [*group-address* | *group-name*] [*source-address* | *source-name*] [**detail**] **Example:** `Router# show ip mpacket smallgroup`	(Optional) Displays the contents of the circular cache header buffer.
Step 8	**show ip pim rp** [**mapping** | **metric**] [*rp-address*] **Example:** `Router# show ip pim rp metric`	(Optional) Displays the RP routers associated with a sparse mode multicast group.
Step 9	**show ip rpf** {*source-address* | *source-name*} [**metric**] **Example:** `Router# show ip rpf 172.16.10.13`	(Optional) Displays how the router is doing RPF (that is, from the unicast routing table, DVMRP routing table, or static mroutes). Also displays the unicast routing metric.

Examples

show ip mroute

The following is sample output from the **show ip mroute** command for a router operating in sparse mode:

```
Router# show ip mroute

IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report, Z - Multicast Tunnel,
       Y - Joined MDT-data group, y - Sending to MDT-data group
Timers: Uptime/Expires
Interface state: Interface, Next-Hop, State/Mode

(*, 224.0.255.3), uptime 5:29:15, RP is 192.168.37.2, flags: SC
  Incoming interface: Tunnel0, RPF neighbor 10.3.35.1, Dvmrp
  Outgoing interface list:
    Ethernet0, Forward/Sparse, 5:29:15/0:02:57

(192.168.46.0/24, 224.0.255.3), uptime 5:29:15, expires 0:02:59, flags: C
  Incoming interface: Tunnel0, RPF neighbor 10.3.35.1
  Outgoing interface list:
    Ethernet0, Forward/Sparse, 5:29:15/0:02:57
```

show ip pim interface

The following is sample output from the **show ip pim interface** command when an interface is specified:

```
Router# show ip pim interface Ethernet1/0

Address          Interface                Ver/   Nbr    Query  DR     DR
                                          Mode   Count  Intvl  Prior
172.16.1.4       Ethernet1/0              v2/S   1      100 ms 1      172.16.1.4
```

show ip mcache

The following is sample output from the **show ip mcache** privileged EXEC command when multicast distributed switching (MDS) is in effect:

```
Router# show ip mcache

IP Multicast Fast-Switching Cache
(*, 239.2.3.4), Fddi3/0/0, Last used: mds
  Tunnel3         MAC Header: 5000602F9C150000603E473F60AAAA030000000800 (Fddi3/0/0)
  Tunnel0         MAC Header: 5000602F9C150000603E473F60AAAA030000000800 (Fddi3/0/0)
  Tunnel1         MAC Header: 5000602F9C150000603E473F60AAAA030000000800 (Fddi3/0/0)
```

show ip mpacket

The following is sample output from the **show ip mpacket** command with the *group-name* argument:

```
Router# show ip mpacket smallgroup

IP Multicast Header Cache - entry count:6, next index: 7
Key: id/ttl timestamp (name) source group

D782/117 206416.908 (company1.company.com) 192.168.228.10 224.5.6.7
7302/113 206417.908 (school.edu) 172.16.2.17 224.5.6.7
6CB2/114 206417.412 (company2.company.com) 172.16.19.40 224.5.6.7
D782/117 206417.868 (company1.company.com) 192.168.228.10 224.5.6.7
```

```
E2E9/123 206418.488 (company3.com) 239.1.8.10 224.5.6.7
1CA7/127 206418.544 (company4.company.com) 192.168.6.10 224.5.6.7
```

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

```
Router# show ip pim rp

Group:227.7.7.7, RP:10.10.0.2, v2, v1, next RP-reachable in 00:00:48
```

show ip pim rp

The following is sample output from the **show ip pim rp** command when the **mapping** keyword is specified:

```
Router# show ip pim rp mapping

PIM Group-to-RP Mappings
This system is an RP (Auto-RP)
This system is an RP-mapping agent

Group(s) 227.0.0.0/8
  RP 10.10.0.2 (?), v2v1, bidir
    Info source:10.10.0.2 (?), via Auto-RP
         Uptime:00:01:42, expires:00:00:32
Group(s) 228.0.0.0/8
  RP 10.10.0.3 (?), v2v1, bidir
    Info source:10.10.0.3 (?), via Auto-RP
         Uptime:00:01:26, expires:00:00:34
Group(s) 229.0.0.0/8
  RP 10.10.0.5 (mcast1.cisco.com), v2v1, bidir
    Info source:10.10.0.5 (mcast1.cisco.com), via Auto-RP
         Uptime:00:00:52, expires:00:00:37
Group(s) (-)230.0.0.0/8
  RP 10.10.0.5 (mcast1.cisco.com), v2v1, bidir
    Info source:10.10.0.5 (mcast1.cisco.com), via Auto-RP
         Uptime:00:00:52, expires:00:00:37
```

The following is sample output from the **show ip pim rp** command when the **metric** keyword is specified:

```
Router# show ip pim rp metric

RP Address      Metric Pref    Metric       Flags   RPF Type   Interface
10.10.0.2       0              0              L     unicast    Loopback0
10.10.0.3       90             409600         L     unicast    Ethernet3/3
10.10.0.5       90             435200         L     unicast    Ethernet3/3
```

show ip rpf

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

```
Router# show ip rpf 172.16.10.13

RPF information for host1 (172.16.10.13)
  RPF interface: BRI0
  RPF neighbor: sj1.cisco.com (172.16.121.10)
  RPF route/mask: 172.16.0.0/255.255.0.0
  RPF type: unicast
  RPF recursion count: 0
  Doing distance-preferred lookups across tables
```

The following is sample output from the **show ip rpf** command when the **metric** keyword is specified:

```
Router# show ip rpf 172.16.10.13 metric
```

```
RPF information for host1.cisco.com (172.16.10.13)
  RPF interface: BRI0
  RPF neighbor: neighbor.cisco.com (172.16.121.10)
  RPF route/mask: 172.16.0.0/255.255.0.0
  RPF type: unicast
  RPF recursion count: 0
  Doing distance-preferred lookups across tables
  Metric preference: 110
  Metric: 11
```

Clearing IP Multicast Routing Table or Caches

Clear IP multicast caches and tables to delete entries from the IP multicast routing table, the Auto-RP cache, the IGMP cache, and the caches of Catalyst switches. When these entries are cleared, the information is refreshed by being relearned, thus eliminating any incorrect entries.

SUMMARY STEPS

1. **enable**
2. **clear ip mroute** {* | *group-name* [*source-name* | *source-address*] | *group-address* [*source-name* | *source-address*]}
3. **clear ip pim auto-rp** *rp-address*
4. **clear ip mcache**
5. **clear ip igmp group** [*group-name* | *group-address* | *interface-type interface-number*]
6. **clear ip cgmp** [*interface-type interface-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ip mroute** {***** | *group-name* [*source-name* | *source-address*] | *group-address* [*source-name* | *source-address*]} **Example:** `Router# clear ip mroute 224.2.205.42 228.3.0.0`	(Optional) Deletes entries from the IP multicast routing table.
Step 3	**clear ip pim auto-rp** *rp-address* **Example:** `Router# clear ip pim auto-rp 224.5.6.7`	(Optional) Clears the Auto-RP cache.
Step 4	`clear ip mcache` **Example:** `Router # clear ip mcache`	(Optional) Clears the multicast cache.

	Command or Action	Purpose
Step 5	**clear ip igmp group** [*group-name* | *group-address* | *interface-type interface-number*] **Example:** `Router# clear ip igmp group 224.0.255.1`	(Optional) Deletes entries from the IGMP cache.
Step 6	**clear ip cgmp** [*interface-type interface-number*] **Example:** `Router# clear ip cgmp`	(Optional) Clears all group entries from the caches of Catalyst switches.

Monitoring IP Multicast Delivery Using IP Multicast Heartbeat

The IP multicast heartbeat feature provides a way to monitor the status of IP multicast delivery and be informed when the delivery fails (via Simple Network Management Protocol [SNMP] traps).

IP Multicast Heartbeat

The IP Multicast Heartbeat feature enables you to monitor the delivery of IP multicast packets and to be alerted if the delivery fails to meet certain parameters.

Although you could alternatively use MRM to monitor IP multicast, you can perform the following tasks with IP multicast heartbeat that you cannot perform with MRM:

- Generate an SNMP trap
- Monitor a production multicast stream

When IP multicast heartbeat is enabled, the router monitors IP multicast packets destined for a particular multicast group at a particular interval. If the number of packets observed is less than a configured minimum amount, the router sends an SNMP trap to a specified network management station to indicate a loss of heartbeat exception.

The **ip multicast heartbeat** command does not create a heartbeat if there is no existing multicast forwarding state for *group* in the router. This command will not create a multicast forwarding state in the router. Use the **ip igmp static-group** command on the router or on a downstream router to force forwarding of IP multicast traffic. Use the **snmp-server host ipmulticast** command to enable the sending of IP multicast traps to specific receiver hosts. Use the **debug ip mhbeat** command to debug the Multicast Heartbeat feature.

SNMP Notifications

A key feature of SNMP is the ability to generate notifications from an SNMP agent. These notifications do not require that requests be sent from the SNMP manager. Unsolicited (asynchronous) notifications can be generated as traps or inform requests. Traps are messages alerting the SNMP manager to a condition on the network. Inform requests (informs) are traps that include a request for confirmation of receipt from the SNMP manager. Notifications can indicate improper user authentication, restarts, the closing of a connection, loss of connection to a neighbor router, or other significant events.

Traps are less reliable than informs because the receiver does not send any acknowledgment when it receives a trap. The sender cannot determine if the trap was received. An SNMP manager that receives an inform request acknowledges the message with an SNMP response protocol data unit (PDU). If the

manager does not receive an inform request, it does not send a response. If the sender never receives a response, the inform request can be sent again. Thus, informs are more likely to reach their intended destination.

However, traps are often preferred because informs consume more resources in the router and in the network. Unlike a trap, which is discarded as soon as it is sent, an inform request must be held in memory until a response is received or the request times out. Also, traps are sent only once, while an inform may be retried several times. The retries increase traffic and contribute to a higher overhead on the network. Thus, traps and inform requests provide a trade-off between reliability and resources. If it is important that the SNMP manager receives every notification, use inform requests. However, if you are concerned about traffic on your network or memory in the router and you need not receive every notification, use traps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast-routing**
4. **snmp-server host** {*hostname* | *ip-address*} [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*]
5. **snmp-server enable traps ipmulticast**
6. **ip multicast heartbeat** *group-address minimum-number window-size interval*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`ip multicast-routing` **Example:** `Router(config)# ip multicast-routing`	Enables IP multicast routing.
Step 4	**snmp-server host** {*hostname* | *ip-address*} [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] **Example:** `Router(config)# snmp-server host 224.1.0.1 traps public`	Specifies the recipient of an SNMP notification operation.

	Command or Action	Purpose
Step 5	**snmp-server enable traps ipmulticast** **Example:** `Router(config)# snmp-server enable traps ipmulticast`	Enables the router to send IP multicast traps.
Step 6	**ip multicast heartbeat** *group-address minimum-number window-size interval* **Example:** `Router(config)# ip multicast heartbeat ethernet0 224.1.1.1 1 1 10`	Enables the monitoring of the IP multicast packet delivery. • The *interval* should be set to a multiple of 10 seconds on platforms that use Multicast Distributed Fast Switching (MDFS) because on those platforms, the packet counters are only updated once every 10 seconds. Other platforms may have other increments.

Examples

The following example shows how to monitor IP multicast packets forwarded through this router to group address 244.1.1.1. If no packet for this group is received in a 10-second interval, an SNMP trap will be sent to the SNMP management station with the IP address of 224.1.0.1.

```
!
ip multicast-routing
!
snmp-server host 224.1.0.1 traps public
snmp-server enable traps ipmulticast
ip multicast heartbeat ethernet0 224.1.1.1 1 1 10
```

Advertising Multicast Multimedia Sessions Using SAP Listener

Enable SAP listener support when you want to use session description and announcement protocols and applications to assist the advertisement of multicast multimedia conferences and other multicast sessions and to communicate the relevant session setup information to prospective participants.

Session Announcement Protocol (SAP)

Session Announcement Protocol (SAP) listener support is needed to use session description and announcement protocols and applications to assist the advertisement of multicast multimedia conferences and other multicast sessions and to communicate the relevant session setup information to prospective participants.

Sessions are described by the Session Description Protocol (SDP), which is defined in RFC 2327. SDP provides a formatted, textual description of session properties (for example, contact information, session lifetime, and the media) being used in the session (for example, audio, video, and whiteboard) with their specific attributes such as time-to-live (TTL) scope, group address, and User Datagram Protocol (UDP) port number.

Many multimedia applications rely on SDP for session descriptions. However, they may use different methods to disseminate these session descriptions. For example, IP/TV relies on the web to disseminate session descriptions to participants. In this example, participants must know of a web server that provides the session information.

MBONE applications (for example, vic, vat, and wb) and other applications rely on multicast session information sent throughout the network. In these cases, SAP is used to transport the SDP session announcements. SAP Version 2 uses the well-known session directory multicast group 224.2.127.254 to disseminate SDP session descriptions for global scope sessions and group 239.255.255.255 for administrative scope sessions.

Note The Session Directory (SDR) application is commonly used to send and receive SDP/SAP session announcements.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sap cache-timeout** *minutes*
4. **interface** *type number*
5. **ip sap listen**
6. **end**
7. **clear ip sap** [*group-address* | **"***session-name***"**]
8. **show ip sap** [*group-address* | **"***session-name***"** | **detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip sap cache-timeout** *minutes* **Example:** `Router(config)# ip sap cache-timeout 600`	(Optional) Limits how long a SAP cache entry stays active in the cache. • By default, SAP cache entries are deleted 24 hours after they are received from the network.
Step 4	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Selects an interface that is connected to hosts on which IGMPv3 can be enabled.
Step 5	**ip sap listen** **Example:** `Router(config-if)# ip sap listen`	Enables the Cisco IOS software to listen to session directory announcements.

	Command or Action	Purpose
Step 6	`end` **Example:** `Router(config-if)# end`	Ends the session and returns to EXEC mode.
Step 7	`clear ip sap [`*group-address* `|` **"***session-name***"**`]` **Example:** `Router# clear ip sap "Sample Session"`	Deletes a SAP cache entry or the entire SAP cache.
Step 8	`show ip sap [`*group-address* `|` **"***session-name***"** `|` **detail**`]` **Example:** `Router# show ip sap 224.2.197.250 detail`	(Optional) Displays the SAP cache.

Examples

The following example enables a router to listen to session directory announcements and changes the SAP cache timeout to 30 minutes.

```
ip multicast routing
ip sap cache-timeout 30
interface loopback 0
 ip address 10.0.0.51 255.255.255.0
 ip pim sparse-dense mode
 ip sap listen
```

The following is sample output from the **show ip sap** command for a session using multicast group 224.2.197.250:

```
Router# show ip sap 224.2.197.250

SAP Cache - 198 entries
Session Name: Session1
   Description: This broadcast is brought to you courtesy of Name1.
   Group: 0.0.0.0, ttl: 0, Contiguous allocation: 1
   Lifetime: from 10:00:00 PDT Jul 4 1999 until 10:00:00 PDT Aug 1 1999
   Uptime: 4d05h, Last Heard: 00:01:40
   Announcement source: 128.102.84.134
   Created by: sample 3136541828 3139561476 IN IP4 128.102.84.134
   Phone number: Sample Digital Video Lab (555) 555-5555
   Email: email1 <name@email.com>
   URL: http://url.com/
   Media: audio 20890 RTP/AVP 0
     Media group: 224.2.197.250, ttl: 127
     Attribute: ptime:40
   Media: video 62806 RTP/AVP 31
     Media group: 224.2.190.243, ttl: 127
```

Storing IP Multicast Headers

You can store IP multicast packet headers in a cache and then display them to determine any of the following information:

- Who is sending IP multicast packets to what groups

- Interpacket delay
- Duplicate IP multicast packets (if any)
- Multicast forwarding loops in your network (if any)
- Scope of the group
- UDP port numbers
- Packet length

Perform this task if you need any of the information listed above.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast cache-headers** [**rtp**]
4. **exit**
5. **show ip mpacket** [*group-address* | *group-name*] [*source-address* | *source-name*] [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast cache-headers** [**rtp**] **Example:** `Router(config)# ip multicast cache-headers`	Allocates a circular buffer to store IP multicast packet headers that the router receives.
Step 4	`exit` **Example:** `Router(config)# exit`	Returns to privilege EXEC mode.
Step 5	**show ip mpacket** [*group-address* | *group-name*] [*source-address* | *source-name*] [**detail**] **Example:** `Router# show ip mpacket smallgroup`	(Optional) Displays the contents of the circular cache-header buffer.

Examples

The following is sample output from the **show ip mpacket** command for the group named “smallgroup.”

```
Router# show ip mpacket smallgroup

IP Multicast Header Cache - entry count:6, next index: 7
Key: id/ttl timestamp (name) source group

D782/117 206416.908 (company1.company.com) 192.168.228.10 224.5.6.7
7302/113 206417.908 (school.edu) 172.16.2.17 224.5.6.7
6CB2/114 206417.412 (company2.company.com) 172.16.19.40 224.5.6.7
D782/117 206417.868 (company1.company.com) 192.168.228.10 224.5.6.7
E2E9/123 206418.488 (company3.com) 239.1.8.10 224.5.6.7
1CA7/127 206418.544 (company4.company.com) 192.168.6.10 224.5.6.7
```

Disabling Fast Switching of IP Multicast

Disable fast switching if you want to log debug messages, because when fast switching is enabled, debug messages are not logged.

You might also want to disable fast switching, which places the router in process switching, if packets are not reaching their destinations. If fast switching is disabled and packets are reaching their destinations, then switching may be the cause.

Fast Switching of IP Multicast

Fast switching of IP multicast packets is enabled by default on all interfaces (including generic routing encapsulation [GRE] and DVMRP tunnels), with one exception: It is disabled and not supported over X.25 encapsulated interfaces. The following are properties of fast switching:

- If fast switching is disabled on an *incoming* interface for a multicast routing table entry, the packet is sent at process level for all interfaces in the outgoing interface list.
- If fast switching is disabled on an *outgoing* interface for a multicast routing table entry, the packet is process-level switched for that interface, but may be fast switched for other interfaces in the outgoing interface list.
- When fast switching is enabled, debug messages are not logged.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no ip pim mroute-cache**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Specifies an interface.
Step 4	**no ip mroute-cache** **Example:** `Router(config-if)# no ip mroute-cache`	Disables fast switching of IP multicast.

Enabling PIM MIB Extensions for IP Multicast

PIM MIB extensions for IP multicast introduce support in Cisco IOS software for the CISCO-PIM-MIB, which is an extension of RFC 2934 and an enhancement to the former Cisco implementation of the PIM MIB.

PIM MIB Extensions

Protocol Independent Multicast (PIM) is an IP multicast routing protocol used for routing multicast data packets to multicast groups. RFC 2934 defines the PIM for IPv4 MIB, which describes managed objects that enable users to remotely monitor and configure PIM using Simple Network Management Protocol (SNMP).

PIM MIB extensions introduce the following new classes of PIM notifications:

- neighbor-change—This notification results from the following conditions:
  - When a router's PIM interface is disabled or enabled (using the **ip pim** command in interface configuration mode)
  - When a router's PIM neighbor adjacency expires or is established (defined in RFC 2934)
- rp-mapping-change—This notification results from a change in the rendezvous point (RP) mapping information due to either Auto-RP or bootstrap router (BSR) messages.
- invalid-pim-message—This notification results from the following conditions:
  - When an invalid (*, G) Join or Prune message is received by the device (for example, when a router receives a Join or Prune message for which the RP specified in the packet is not the RP for the multicast group)

– When an invalid PIM register message is received by the device (for example, when a router receives a register message from a multicast group for which it is not the RP)

Benefits of PIM MIB Extensions

PIM MIB extensions have the following benefits:

- Allow users to identify changes in the multicast topology of their network by detecting changes in the RP mapping.
- Provide traps to monitor the PIM protocol on PIM-enabled interfaces.
- Help users identify routing issues when multicast neighbor adjacencies expire or are established on a multicast interface.
- Enable users to monitor RP configuration errors (for example, errors due to flapping in dynamic RP allocation protocols like Auto-RP).

Restrictions for PIM MIB Extensions

The following MIB tables are not supported in Cisco IOS software:

- pimIpMRouteTable
- pimIpMRouteNextHopTable
- The pimInterfaceVersion object was removed from RFC 2934 and, therefore, is no longer supported in Cisco IOS software.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server enable traps pim** [**neighbor-change** | **rp-mapping-change** | **invalid-pim-message**]
4. **snmp-server host** *host-address* [**traps** | **informs**] *community-string* **pim**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**snmp-server enable traps pim** [**neighbor-change** | **rp-mapping-change** | **invalid-pim-message**] **Example:** `Router(config)# snmp-server enable traps pim neighbor-change`	(Optional) Enables a router to send PIM notifications. The keywords are as follows: • **neighbor-change**—Enables notifications indicating when a router's PIM interface is disabled or enabled, or when a router's PIM neighbor adjacency expires or is established. • **rp-mapping-change**—Enables notifications indicating a change in RP mapping information due to either Auto-RP or BSR messages. • **invalid-pim-message**—Enables notifications for monitoring invalid PIM protocol operations (for example, when a router receives a join or prune message for which the RP specified in the packet is not the RP for the multicast group or when a router receives a register message from a multicast group for which it is not the RP).
Step 4	**snmp-server host** *host-address* [**traps** | **informs**] *community-string* **pim** **Example:** `Router(config)# snmp-server host 10.10.10.10 traps public pim`	Specifies the recipient of a PIM SNMP notification operation.

Configuration Examples for Monitoring and Maintaining IP Multicast

This section provides the following configuration example:

• Generating Notifications That PIM Is Enabled: Example, page 18

Generating Notifications That PIM Is Enabled: Example

The following example shows how to configure a router to generate notifications indicating that a PIM interface of the router has been enabled. The first line configures PIM traps to be sent as SNMP v2c traps to the host with IP address 10.0.0.1. The second line configures the router to send the neighbor-change class of notification to the host.

```
snmp-server host 10.0.0.1 traps version 2c public pim
snmp-server enable traps pim neighbor-change
interface ethernet0/0
 ip pim sparse-dense-mode
```

Additional References

The following sections provide references related to monitoring and maintaining IP multicast.

Related Documents

Related Topic	Document Title
IP multicast SNMP notifications	"Configuring SNMP Support" module
IP multicast commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS IP Multicast Command Reference*

MIBs

MIBs	MIBs Link
• CISCO-IPMROUTE-MIB • MSDP-MIB • IGMP-STD-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2934	*Protocol Independent Multicast for IPv4 MIB*

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 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/public/support/tac/home.shtml

Feature Information for Monitoring and Maintaining IP Multicast

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Releases 12.2(1) or 12.0(3)S or later appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for specific commands was introduced, see the command reference documents.

If you are looking for information on a feature in this technology that is not documented here, see the "Configuring IP Multicast Roadmap".

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator (http://www.cisco.com/go/fn). You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click Cancel at the login dialog box and follow the instructions that appear.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Monitoring and Maintaining IP Multicast

Feature Names	Releases	Feature Configuration Information
PIM MIB Extensions	12.2(4)T	Protocol Independent Multicast (PIM) is an IP multicast routing protocol used for routing multicast data packets to multicast groups. RFC 2934 defines the PIM for IPv4 MIB, which describes managed objects that enable users to remotely monitor and configure PIM using Simple Network Management Protocol (SNMP). The following sections provide information about this feature: "Enabling PIM MIB Extensions for IP Multicast" section on page 16
PIM MIB Extensions	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Load Splitting IP Multicast Traffic over ECMP

First Published: May 2, 2005
Last Updated: February 27, 2007

This module describes how to load split IP multicast traffic over Equal Cost Multipath (ECMP). Multicast traffic from different sources or from different sources and groups are load split across equal-cost paths to take advantage of multiple paths through the network.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Load Splitting IP Multicast Traffic over ECMP" section on page 21.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Load Splitting IP Multicast Traffic over ECMP

This module assumes you have met the following prerequisites:

- You understand the concepts in the "IP Multicast Technology Overview" module.
- You have IP multicast configured in your network. See the "Configuring Basic IP Multicast" module.

Information About Load Splitting IP Multicast Traffic over ECMP

Before you load split IP multicast traffic, you should understand the following concepts:

Load Splitting Versus Load Balancing

Load splitting and load balancing are not the same. Load splitting provides a means to randomly distribute (*, G) and (S, G) traffic streams across multiple equal-cost reverse path forwarding (RPF) paths, which does not necessarily result in a balanced IP multicast traffic load on those equal-cost RPF paths. By randomly distributing (*, G) and (S, G) traffic streams, the methods used for load splitting IP multicast traffic attempt to distribute an equal amount of traffic flows on each of the available RPF paths not by counting the flows, but, rather, by making a pseudorandom decision. These methods are collectively referred to as ECMP multicast load splitting methods. ECMP multicast load splitting methods, thus, result in better load-sharing in networks where there are many traffic streams that utilize approximately the same amount of bandwidth.

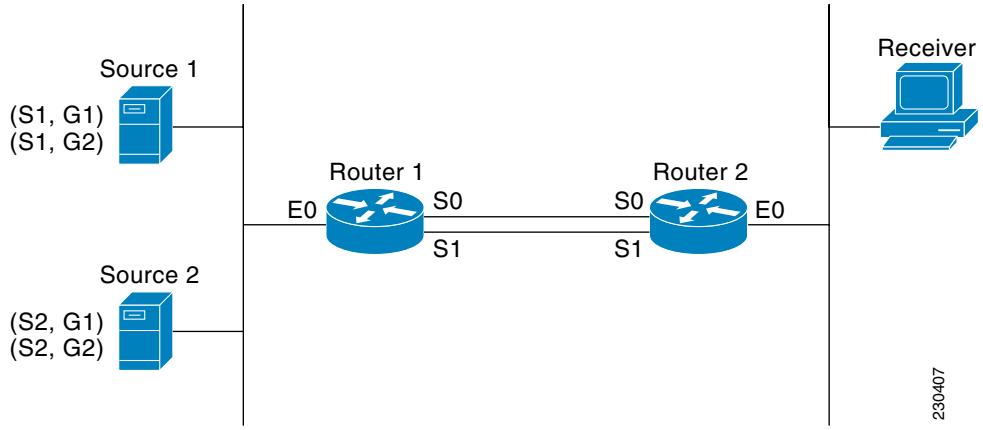
If there are just a few (S, G) or (*, G) states flowing across a set of equal-cost links, the chance that they are well balanced is quite low. To overcome this limitation, precalculated source addresses—for (S, G) states—or rendezvous point (RP) addresses—for (*, G) states—can be used to achieve a reasonable form of load balancing. This limitation applies equally to the per-flow load splitting in Cisco Express Forwarding (CEF) or with EtherChannels: As long as there are only a few flows, those methods of load splitting will not result in good load distribution without some form of manual engineering.

Default Behavior for IP Multicast When Multiple Equal-Cost Paths Exist

By default, for Protocol Independent Multicast sparse mode (PIM-SM), Source Specific Multicast (PIM-SSM), bidirectional PIM (bidir-PIM), and PIM dense mode (PIM-DM) groups, if multiple equal-cost paths are available, Reverse Path Forwarding (RPF) for IPv4 multicast traffic is based on the PIM neighbor with the highest IP address. This method is referred to as *the highest PIM neighbor behavior*. This behavior is in accordance with RFC 2362 for PIM-SM, but also applies to PIM-SSM, PIM-DM, and bidir-PIM.

Figure 1 illustrates a sample topology that is used in this section to explain the default behavior for IP multicast when multiple equal-cost paths exist.

Figure 1 Default Behavior for IP Multicast When Multiple Equal-Cost Paths Exist



In Figure 1, two sources, S1 and S2, are sending traffic to IPv4 multicast groups, G1 and G2. Either PIM-SM, PIM-SSM, or PIM-DM can be used in this topology. If PIM-SM is used, assume that the default of 0 for the **ip pim spt-threshold** command is being used on Router 2, that an Interior Gateway Protocol (IGP) is being run, and that the output of the **show ip route** command for S1 and for S2 (when entered on Router 2) displays serial interface 0 and serial interface 1 on Router 1 as equal-cost next-hop PIM neighbors of Router 2.

Without further configuration, IPv4 multicast traffic in the topology illustrated in Figure 1 would always flow across one serial interface (either serial interface 0 or serial interface 1), depending on which interface has the higher IP address. For example, suppose that the IP addresses configured on serial interface 0 and serial interface 1 on Router 1 are 10.1.1.1 and 10.1.2.1, respectively. Given that scenario, in the case of PIM-SM and PIM-SSM, Router 2 would always send PIM join messages towards 10.1.2.1 and would always receive IPv4 multicast traffic on serial interface 1 for all sources and groups shown in Figure 1. In the case of PIM-DM, Router 2 would always receive IP multicast traffic on serial Interface 1, only that in this case, PIM join messages are not used in PIM-DM; instead Router 2 would prune the IP multicast traffic across serial interface 0 and would receive it through serial interface 1 because that interface has the higher IP address on Router 1.

IPv4 RPF lookups are performed by intermediate multicast router to determine the RPF interface and RPF neighbor for IPv4 (*,G) and (S, G) multicast routes (trees). An RPF lookup consists of RPF route-selection and route-path-selection. RPF route-selection operates solely on the IP unicast address to identify the root of the multicast tree. For (*, G) routes (PIM-SM and Bidir-PIM), the root of the multicast tree is the RP address for the group G; for (S, G) trees (PIM-SM, PIM-SSM and PIM-DM), the root of the multicast tree is the source S. RPF route-selection finds the best route towards the RP or source in the routing information base (RIB), and, if configured (or available), the Distance Vector

Multicast Routing Protocol (DVMRP) routing table, the Multiprotocol Border Gateway Protocol (MBGP) routing table or configured static mroutes. If the resulting route has only one available path, then the RPF lookup is complete, and the next-hop router and interface of the route become the RPF neighbor and RPF interface of this multicast tree. If the route has more than one path available, then route-path-selection is used to determine which path to choose.

For IP multicast, the following route-path-selection methods are available:

Note All methods but the default method of route-path-selection available in IP multicast enable some form of ECMP multicast load splitting.

- Highest PIM neighbor—This is the default method; thus, no configuration is required. If multiple equal-cost paths are available, RPF for IPv4 multicast traffic is based on the PIM neighbor with the highest IP address; as a result, without configuration, ECMP multicast load splitting is disabled by default.
- ECMP multicast load splitting method based on source address—You can configure ECMP multicast load splitting using the **ip multicast multipath** command. Entering this form of the **ip multicast multipath** command enables ECMP multicast load splitting based on source address using the S-hash algorithm. For more information, see the "ECMP Multicast Load Splitting Based on Source Address Using the S-Hash Algorithm" section.
- ECMP multicast load splitting method based on source and group address—In Cisco IOS Release 12.2(33)SRB and subsequent 12.2SR releases, you can configure ECMP multicast load splitting using the **ip multicast multipath** command with the **s-g-hash** and **basic** keywords. Entering this form of the **ip multicast multipath** command enables ECMP multicast load splitting based on source and group address using the basic S-G-hash algorithm. For more information, see the "ECMP Multicast Load Splitting Based on Source and Group Address Using the Basic S-G-Hash Algorithm" section.
- ECMP multicast load splitting method based on source, group, and next-hop address—In Cisco IOS Release 12.2(33)SRB and subsequent 12.2SR releases, you can configure ECMP multicast load splitting using the **ip multicast multipath** command with the **s-g-hash** and **next-hop-based** keywords. Entering that form of the command splits IP multicast traffic based on source, group, and next-hop address using the next-hop-based S-G-hash algorithm. For more information, see the "ECMP Multicast Load Splitting Based on Source, Group, and Next-Hop Address Using the Next-Hop-Based S-G-Hash Algorithm" section.

The default behavior (the highest PIM neighbor behavior) does not result in any form of ECMP load-splitting in IP multicast, but instead selects the PIM neighbor that has the highest IP address among the next-hop PIM neighbors for the available paths. A next hop is considered to be a PIM neighbor when it displays in the output of the **show ip pim neighbor** command, which is the case when PIM hello messages have been received from it and have not timed out. If none of the available next hops are PIM neighbors, then simply the next hop with the highest IP address is chosen.

Methods to Load Split IP Multicast Traffic

In general, the following methods are available to load split IP multicast traffic:

- You can enable ECMP multicast load splitting based on source address, based on source and group address, or based on source, group, and next-hop address. After the equal-cost paths are recognized, ECMP multicast load splitting operates on a per (S, G) basis, rather than a per packet basis as in unicast traffic.

- Alternative methods to load split IP multicast are to consolidate two or more equal-cost paths into a generic routing encapsulation (GRE) tunnel and allow the unicast routing protocol to perform the load splitting, or to load split across bundle interfaces, such as Fast or Gigabit EtherChannel interfaces, Multilink PPP (MLPPP) link bundles, or Multilink Frame Relay (FR.16) link bundles.

Overview of ECMP Multicast Load Splitting

By default, ECMP multicast load splitting of IPv4 multicast traffic is disabled. ECMP multicast load splitting can be enabled using the **ip multicast multipath** command.

ECMP Multicast Load Splitting Based on Source Address Using the S-Hash Algorithm

The **ip multicast multipath** command is used to enable ECMP multicast load splitting traffic based on source address using the S-hash algorithm. When the **ip multicast multipath** command is configured, the RPF interface for each (*, G) or (S, G) state will be selected among the available equal-cost paths, depending on the RPF address to which the state resolves. For an (S, G) state, the RPF address is the source address of the state; for a (*, G) state, the RPF address is the address of the RP associated with the group address of the state.

When ECMP multicast load splitting based on source address is configured, multicast traffic for different states can be received across more than just one of the equal-cost interfaces. The method applied by IPv4 multicast is quite similar in principle to the default per-flow load splitting in IPv4 CEF or the load splitting used with Fast and Gigabit EtherChannels. This method of ECMP multicast load splitting, however, is subject to polarization.

Note For more information about ECMP multicast load splitting and polarization, see the "Polarization As a By-Product of Using the S-Hash and Basic S-G-Hash Algorithms for ECMP Multicast Load Splitting" section.

ECMP Multicast Load Splitting Based on Source and Group Address Using the Basic S-G-Hash Algorithm

In Cisco IOS Release 12.2(33)SRB and subsequent 12.2SR releases, the **ip multicast multipath** command is used with the **s-g-hash** and **basic** keywords to enable ECMP multicast load splitting based on source and group address. The **basic** keyword enables a simple hash, referred to as the basic S-G-hash algorithm, which is based on source and group address. The basic S-G-hash algorithm is predictable because no randomization is used in coming up with the hash value. The S-G-hash mechanism, however, is subject to polarization because for a given source and group, the same hash is always picked irrespective of the router this hash is being calculated on.

Note For more information about ECMP multicast load splitting and polarization, see the "Polarization As a By-Product of Using the S-Hash and Basic S-G-Hash Algorithms for ECMP Multicast Load Splitting" section.

Note The basic S-G-hash algorithm ignores bidir-PIM groups.

Predictability As a By-Product of Using the S-Hash and Basic S-G-Hash Algorithms for ECMP Multicast Load Splitting

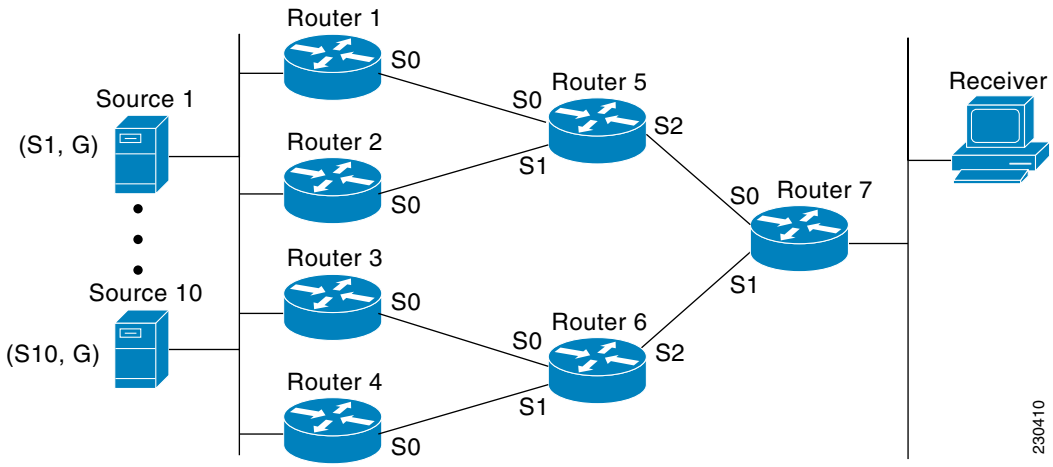
The method used by ECMP multicast load splitting in IPv4 multicast allows for consistent load splitting in a network where the same number of equal-cost paths are present in multiple places in a topology. If an RP address or source addresses are calculated once to have flows split across *N* paths, then they will be split across those *N* paths in the same way in all places in the topology. Consistent load splitting, thus, allows for predictability, which, in turns, enables load splitting of IPv4 multicast traffic to be manually engineered.

Polarization As a By-Product of Using the S-Hash and Basic S-G-Hash Algorithms for ECMP Multicast Load Splitting

The hash mechanism used in IPv4 multicast to load split multicast traffic by source address or by source and group address is subject to a problem usually referred to as *polarization*. A by-product of ECMP multicast load splitting based on source address or on source and group address, polarization is a problem that prevents routers in some topologies from effectively utilizing all available paths for load splitting.

Figure 2 illustrates a sample topology that is used in this section to explain the problem of polarization when configuring ECMP multicast load splitting based on source address or on source and group address.

Figure 2 Polarization Topology



In the topology illustrated in Figure 2, notice that Router 7 has two equal-cost paths towards the sources, S1 to S10, through Router 5 and Router 6. For this topology, suppose that ECMP multicast load splitting is enabled with the **ip multicast multipath** command on all routers in the topology. In that scenario, Router 7 would apply equal-cost load splitting to the 10 (S, G) states. The problem of polarization in this scenario would affect Router 7 because that router would end up choosing serial interface 0 on Router 5 for sources S1 to S5 and serial interface 1 on Router 6 for sources S6 to S10. The problem of polarization, furthermore, would also affect Router 5 and Router 6 in this topology. Router 5 has two equal-cost paths for S1 to S5 through serial interface 0 on Router 1 and serial interface 1 on Router 2. Because Router 5 would apply the same hash algorithm to select which of the two paths to use, it would end up using just one of these two upstream paths for sources S1 to S5; that is, either all the traffic would flow across

Router 1 and Router 5 *or* across Router 2 and Router 5. It would be impossible in this topology to utilize Router 1 and Router 5 *and* Router 2 and Router 5 for load splitting. Likewise, the polarization problem would apply to Router 3 and Router 6 *and* Router 4 and Router 6; that is, it would be impossible in this topology to utilize both Router 3 and Router 6 *and* Router 4 and Router 6 for load splitting.

ECMP Multicast Load Splitting Based on Source, Group, and Next-Hop Address Using the Next-Hop-Based S-G-Hash Algorithm

In Cisco IOS 12.2(33)SRB and subsequent 12.2SR releases, the **ip multicast multipath** command is used with the **s-g-hash** and **next-hop-based** keywords to enable ECMP multicast load splitting based on source, group, and next-hop address. The **next-hop-based** keyword enables a more complex hash, the next-hop-based S-G-hash algorithm, which is based on source, group, and next-hop address. The next-hop-based S-G-hash algorithm is predictable because no randomization is used in calculating the hash value. Unlike the S-hash and basic S-G-hash algorithms, the hash mechanism used by the next-hop-based S-G-hash algorithm is not subject to polarization.

Note The next-hop-based S-G-hash algorithm in IPv4 multicast is the same algorithm used in IPv6 ECMP multicast load splitting, which, in turn, utilizes the same hash function used for PIM-SM bootstrap router (BSR).

The next-hop-based hash mechanism does not produce polarization and also maintains better RPF stability when paths fail. These benefits come at the cost that the source or RP IP addresses cannot be used to reliably predict and engineer the outcome of load splitting when the next-hop-based S-G-hash algorithm is used. Because many customer networks have implemented equal-cost multipath topologies, the manual engineering of load splitting, thus, is not a requirement in many cases. Rather, it is more of a requirement that the default behavior of IP multicast be similar to IP unicast; that is, it is expected that IP multicast use multiple equal-cost paths on a best-effort basis. Load splitting for IPv4 multicast, therefore, could not be enabled by default because of the anomaly of polarization.

Note Load splitting for Cisco IOS CEF unicast also uses a method that does not exhibit polarization and likewise cannot be used to predict the results of load splitting or engineer the outcome of load splitting.

The next-hop-based hash function avoids polarization because it introduces the actual next-hop IP address of PIM neighbors into the calculation, so the hash results are different for each router, and in effect, there is no problem of polarization. In addition to avoiding polarization, this hash mechanism also increases stability of the RPF paths chosen in the face of path failures. Consider a router with four equal-cost paths and a large number of states that are load split across these paths. Suppose that one of these paths fails, leaving only three available paths. With the hash mechanism used by the polarizing hash mechanisms (the hash mechanism used by the S-hash and basic S-G-hash algorithms), the RPF paths of all states would likely reconverge and thus change between those three paths, especially those paths that were already using one of those three paths. These states, therefore, may unnecessarily change their RPF interface and next-hop neighbor. This problem exists simply because the chosen path is determined by taking the total number of paths available into consideration by the algorithm, so once a path changes, the RPF selection for all states is subject to change too. For the next-hop-based hash mechanism, only the states that were using the changed path for RPF would need to reconverge onto one of the three remaining paths. The states that were already using one of those paths would not change. If the fourth path came back up, the states that initially used it would immediately reconverge back to that path without affecting the other states.

Note The next-hop-based S-G-hash algorithm ignores bidir-PIM groups.

Effect of ECMP Multicast Load Splitting on PIM Neighbor Query and Hello Messages for RPF Path Selection

When the **ip multicast multipath** command is *not* enabled, and there are multiple equal-cost paths towards an RP or a source, IPv4 multicast will first elect the highest IP address PIM neighbor. A PIM neighbor is a router from which PIM hello (or PIMv1 query) messages are received. For example, consider a router that has two equal-cost paths learned by an IGP or configured through two static routes. The next hops of these two paths are 10.1.1.1 and 10.1.2.1. If both of these next-hop routers send PIM hello messages, then 10.1.2.1 would be selected as the highest IP address PIM neighbor. If only 10.1.1.1 sends PIM hello messages, then 10.1.1.1 would be selected. If neither of these routers sends PIM hello messages, then 10.1.2.1 would be selected. This deferrence to PIM hello messages allows the construction of certain types of dynamic failover scenarios with only static multicast routes (mroutes); it is otherwise not very useful.

Note For more information about configuring static mroutes, see the "Configuring Multiple Static Mroutes in Cisco IOS" configuration note on the Cisco IOS IP multicast FTP site, which is available at the following FTP path: ftp://ftpeng.cisco.com/ipmulticast.

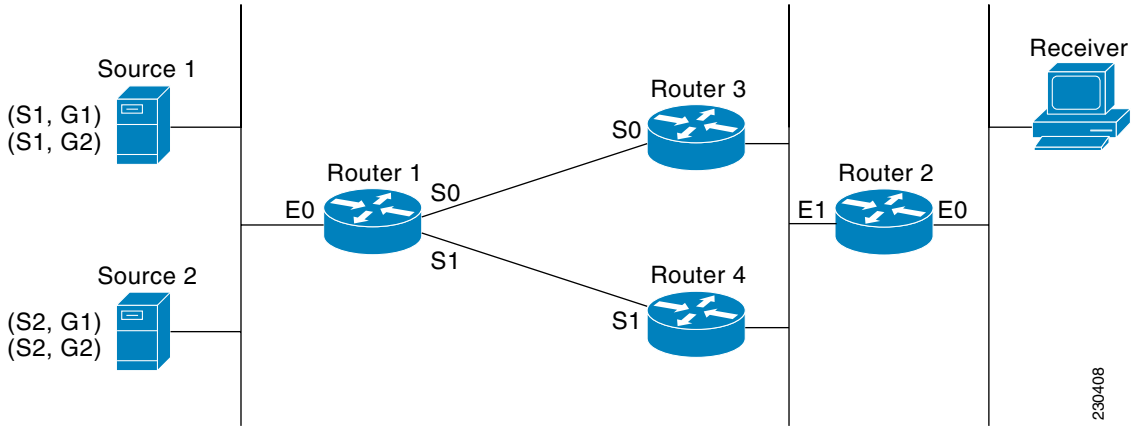
When the **ip multicast multipath** command *is* enabled, the presence of PIM hello message from neighbors is not considered; that is, the chosen RPF neighbor does not depend on whether or not PIM hello messages are received from that neighbor—it only depends on the presence or absence of an equal-cost route entry.

Effect of ECMP Multicast Loading Splitting on Assert Processing in PIM-DM and DF Election in Bidir-PIM

The **ip multicast multipath** command only changes the RPF selection on the downstream router; it does not have an effect on designated forwarder (DF) election in bidir-PIM or the assert processing on upstream routers in PIM-DM.

Figure 3 illustrates a sample topology that is used in this section to explain the effect of ECMP multicast load splitting on assert processing in PIM-DM and DF election in bidir-PIM.

Figure 3 ECMP Multicast Load Splitting and Assert Processing in PIM-DM and DF Election in Bidir-PIM



In Figure 3, Router 2 has two equal-cost paths to S1 and S2 and the RP addresses on Router 1. Both paths are across Ethernet interface 1: one path towards Router 3 and one path towards Router 4. For PIM-SM and PIM-SSM (*, G) and (S, G) RPF selection, there is no difference in the behavior of Router 2 in this topology versus Router 2 in the topology illustrated in Figure 1. There is, however, a difference when using PIM-DM or bidir-PIM.

If PIM-DM is used in the topology illustrated in Figure 3, Router 3 and Router 4 would start flooding traffic for the states onto Ethernet interface 1 and would use the PIM assert process to elect one router among them to forward the traffic and to avoid traffic duplication. As both Router 3 and Router 4 would have the same route cost, the router with the higher IP address on Ethernet interface 1 would always win the assert process. As a result, if PIM-DM is used in this topology, traffic would *not* be load split across Router 3 and Router 4.

If bidir-PIM is used in the topology illustrated in Figure 3, a process called DF election would take place between Router 2, Router 3, and Router 4 on Ethernet interface 1. The process of DF election would elect one router for each RP to forward traffic across Ethernet interface 1 for any groups using that particular RP, based on the router with the highest IP address configured for that interface. Even if multiple RPs are used (for example one for G1 and another one for G2), the DF election for those RPs would always be won by the router that has the higher IP address configured on Ethernet interface 1 (either Router 3 or Router 4 in this topology). The election rules used for DF election are virtually the same as the election rules used for the PIM assert process, only the protocol mechanisms to negotiate them are more refined for DF election (in order to return the results more expediently). As a result, when bidir-PIM is used in this topology, load splitting would always occur across Ethernet interface 1.

The reason that ECMP multicast load splitting does influence the RPF selection but *not* the assert process in PIM-DM or DF election in bidir-PIM is because both the assert process and DF election are cooperative processes that need to be implemented consistently between participating routers. Changing them would require some form of protocol change that would also need to be agreed upon by the participating routers. RPF selection is a purely router local policy and, thus, can be enabled or disabled without protocol changes individually on each router.

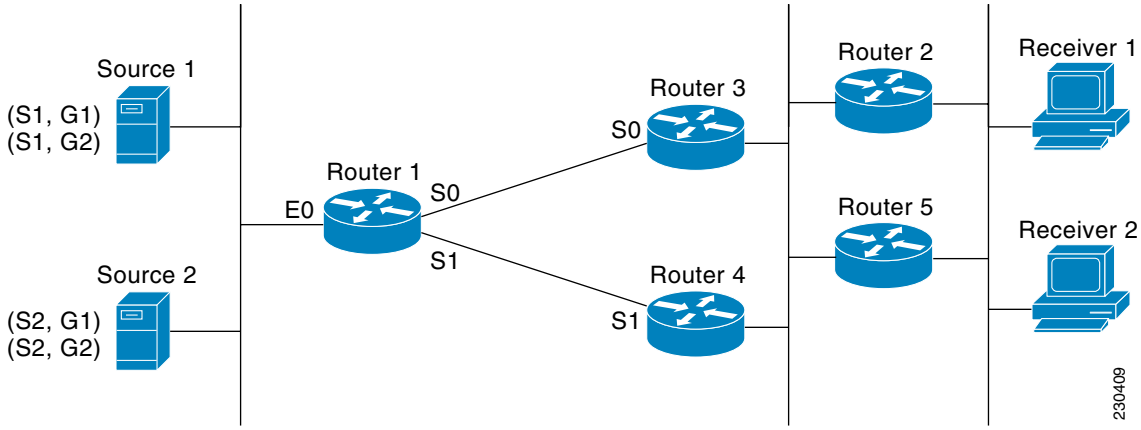
For PIM-DM and bidir-PIM, configuring ECMP multicast load splitting with the **ip multicast multipath** command is only effective in topologies where the equal-cost paths are not upstream PIM neighbors on the same LAN, but rather neighbors on different LANs or point-to-point links.

Effect of ECMP Multicast Load Splitting on the PIM Assert Process in PIM-SM and PIM-SSM

There are also cases where ECMP multicast load splitting with the **ip multicast multipath** command can become ineffective due to the PIM assert process taking over, even when using PIM-SM with (*, G) or (S, G) forwarding or PIM-SSM with (S, G) forwarding.

Figure 4 illustrates a sample topology that is used in this section to explain the effect of ECMP multicast load splitting on the PIM assert process in PIM-SM and PIM-SSM.

Figure 4 ECMP Multicast Load Splitting and the PIM Assert Process in PIM-SM and PIM-SSM



In the topology illustrated in Figure 4, if both Router 2 and Router 5 are Cisco routers and are consistently configured for ECMP multicast load splitting with the **ip multicast multipath** command, then load splitting would continue to work as expected; that is, both routers would have Router 3 and Router 4 as equal-cost next hops and would sort the list of equal-cost paths in the same way (by IP address). When applying the multipath hash function, for each (S, G) or (*, G) state, they would choose the same RPF neighbor (either Router 3 or Router 4) and send their PIM joins to this neighbor.

If Router 5 and Router 2 are inconsistently configured with the **ip multicast multipath** command, or if Router 5 is a third-party router, then Router 2 and Router 5 may choose different RPF neighbors for some (*, G) or (S, G) states. For example Router 2 could choose Router 3 for a particular (S, G) state or Router 5 could choose Router 4 for a particular (S, G) state. In this scenario, Router 3 and Router 4 would both start to forward traffic for that state onto Ethernet interface 1, see each other's forwarded traffic, and—to avoid traffic duplication—start the assert process. As a result, for that (S, G) state, the router with the higher IP address for Ethernet interface 1 would forward the traffic. However, both Router 2 and Router 5 would be tracking the winner of the assert election and would send their PIM joins for that state to this assert winner, even if this assert winner is not the same router as the one that they calculated in their RPF selection. For PIM-SM and PIM-SSM, therefore, the operation of ECMP multicast load splitting can only be guaranteed when all downstream routers on a LAN are consistently configured Cisco routers.

ECMP Multicast Load Splitting and Reconvergence When Unicast Routing Changes

When unicast routing changes, all IP multicast routing states reconverge immediately based on the available unicast routing information. Specifically, if one path goes down, the remaining paths reconverge immediately, and if the path comes up again, multicast forwarding will subsequently reconverge to the same RPF paths that were used before the path failed. Reconvergence occurs whether the **ip multicast multipath** command is configured or not.

Use of BGP with ECMP Multicast Load Splitting

ECMP multicast load splitting works with RPF information learned through BGP in the same way as with RPF information learned from other protocols: It chooses one path out of the multiple paths installed by the protocol. The main difference with BGP is that it only installs a single path, by default. For example, when a BGP speaker learns two identical external BGP (eBGP) paths for a prefix, it will choose the path with the lowest router ID as the best path. The best path is then installed in the IP routing table. If BGP multipath support is enabled and the eBGP paths are learned from the same neighboring AS, instead of picking the single best path, BGP installs multiple paths in the IP routing table. By default, BGP will install only one path to the IP routing table.

To leverage ECMP multicast load splitting for BGP learned prefixes, you must enable BGP multipath using the **maximum-paths** command. Once configured, when BGP installs the remote next-hop information, RPF lookups will recurse to find the best next hop towards that BGP next hop (as in unicast). If for example there is only a single BGP path for a given prefix, but there are two IGP paths to reach that BGP next hop, then multicast RPF will correctly load split between the two different IGP paths.

Note For more information about BGP multipath, see the "iBGP Multipath Load Sharing" and "BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN" chapters in the "BGP" part of the *Cisco IOS IP Routing Configuration Guide*, Release 12.4.

Use of ECMP Multicast Load Splitting with Static Mroutes

If it is not possible to use an IGP to install equal cost routes for certain sources or RPs, static routes can be configured using the **ip route** command to specify the equal-cost paths for load splitting. You cannot use static mroutes (configured with the **ip mroute** command) to configure equal-cost paths because Cisco IOS software does not support the configuration of one static mroute per prefix. There are some workarounds for this limitation using recursive route lookups, but the workarounds cannot be applied to equal-cost multipath routing.

Note For more information about configuring static mroutes, see the "Configuring Multiple Static Mroutes in Cisco IOS" configuration note on the Cisco IOS IP multicast FTP site, which is available at the following FTP path: ftp://ftpeng.cisco.com/ipmulticast.

If you only want to specify static mroutes for equal-cost multipaths, in IPv4 multicast you can specify static mroutes using the **ip mroute** command; those static mroutes, however, would only apply to multicast. If you want to specify that the equal-cost multipaths apply to both unicast and multicast

routing, you can configure static routes using the **ip route** command. In Cisco IOS IPv6 multicast, there is no such restriction; that is, equal-cost multipath mroutes can be configured for static IPv6 mroutes that apply to only unicast routing, only multicast routing, or both.

Note For more information about configuring IPv6 static mroutes, see the "Implementing IPv6 Multicast" chapter in the *Cisco IOS IPv6 Configuration Guide*, Release 12.4.

Alternative Methods of Load Splitting IP Multicast Traffic

Load splitting of IP multicast traffic can also be achieved by consolidating multiple parallel links into a single tunnel over which the multicast traffic is then routed. This method of load splitting is more complex to configure than ECMP multicast load splitting using the **ip multicast multipath** command. One such case where configuring load splitting across equal-cost paths using GRE links can be beneficial is the case where the total number of (S, G) or (*, G) states is so small and the bandwidth carried by each state so variable that even the manual engineering of the source or RP addresses cannot guarantee the appropriate load splitting of the traffic.

Note With the availability of ECMP multicast load splitting, tunnels typically only need to be used if per-packet load sharing is required.

IP multicast traffic can also be used to load split across bundle interfaces, such as Fast or Gigabit EtherChannel interfaces, MLPPP link bundles or Multilink Frame Relay (FRF.16) bundles. GRE or other type of tunnels can also constitute such forms of Layer 2 link bundles. Before using such an Layer 2 mechanism, it is necessary to understand how unicast and multicast traffic is load split.

For more information about load splitting IP multicast traffic using a GRE tunnel, see the "Configuring IP Multicast" chapter in the "IP Multicast" part of the *Cisco IOS IP Configuration Guide*, Release 12.2.

Note Before load splitting IP multicast traffic across equal-cost paths over a tunnel, you need to configure CEF per-packet load balancing or else the GRE packets will not be load balanced per packet. For information about configuring CEF per-packet load balancing, see the "Configuring a Load-Balancing Scheme for CEF Traffic" module.

For more information about Cisco IOS software support of MLPPP link bundles, Fast or Gigabit EtherChannels, and Multilink Frame Relay (FRF.16) bundles, perform a search on the Cisco Support site based on your hardware platform.

How to Load Split IP Multicast Traffic over ECMP

This section contains the following procedure:

- Enabling ECMP Multicast Load Splitting, page 13 (required)

Enabling ECMP Multicast Load Splitting

Perform one of the following tasks to load split IP multicast traffic across multiple equal-cost paths, depending on whether you want to enable ECMP multicast load splitting based on source address, source and group address, or on source, group, and next-hop address:

- Enabling ECMP Multicast Load Splitting Based on Source Address Using the S-Hash Algorithm, page 14
- Enabling ECMP Multicast Load Splitting Based on Source and Group Address Using the Basic S-G-Hash Algorithm, page 16
- Enabling ECMP Multicast Load Splitting Based on Source, Group, and Next-Hop Address Using the Next-Hop-Based S-G-Hash Algorithm, page 17

ECMP Multicast Load Splitting

If two or more equal-cost paths from a source are available, unicast traffic will be load split across those paths. However, by default, multicast traffic is not load split across multiple equal-cost paths. In general, multicast traffic flows down from the RPF neighbor. According to PIM specifications, this neighbor must have the highest IP address if more than one neighbor has the same metric.

Configuring load splitting with the **ip multicast multipath** command causes the system to load split multicast traffic across multiple equal-cost paths based on source address using the S-hash algorithm. When the **ip multicast multipath** command is configured and multiple equal-cost paths exist, the path in which multicast traffic will travel is selected based on the source IP address. Multicast traffic from different sources will be load split across the different equal-cost paths. Load splitting will not occur across equal-cost paths for multicast traffic from the same source sent to different multicast groups.

Note The **ip multicast multipath** command load splits the traffic and does not *load balance* the traffic. Traffic from a source will use only one path, even if the traffic far outweighs traffic from other sources.

If the **ip multicast multipath** command is configured with the **s-g-hash** keyword and multiple equal-cost paths exist, load splitting will occur across equal-cost paths based on source and group address or on source, group, and next-hop address. If you specify the optional **s-g-hash** keyword for load splitting IP multicast traffic, you must select the algorithm used to calculate the equal-cost paths by specifying one of the following keywords:

- **basic**—Enables a simple hash based on source and group address. The basic S-G-hash algorithm is predictable because no randomization is used in calculating the hash value. The basic S-G-hash algorithm, however, is subject to polarization because for a given source and group, the same hash is always picked irrespective of the router the hash is being calculated on.
- **next-hop-based**—Enables a more complex hash based on source, group, and next-hop address. The next-hop-based S-G-hash algorithm is predictable because no randomization is used in coming up with the hash value. Unlike the S-hash and basic S-G-hash algorithms, the next-hop-based hash mechanism is not subject to polarization.

Prerequisites

- Be sure to enable the **ip multicast multipath** command on the router that is supposed to be the *receiver* for traffic from more than one incoming interfaces, which is opposite of unicast routing. From the perspective of unicast, multicast is active on the *sending* router connecting to more than one outgoing interfaces.
- When enabling ECMP multicast load splitting based on source address, make sure you have an adequate number of sources (that is, at least more than two sources). Because ECMP multicast load splitting is statistically based on source address, if you only have two sources, the two sources may end up using the same link, which, of course, negates ECMP load splitting capabilities.
- When using PIM-SM with shortest path tree (SPT) forwarding, ensure that the T-bit is set for the forwarding of all (S, G) states.
- Before performing this task ensure that there are multiple paths for sources. Use the **show ip route** command with the IP address of the source for the *ip-address* argument to validate that there are multiple paths available to the source or the IP address of the RP to validate that there are multiple paths available to the RP. If you do not see multiple paths in the output of the **show ip route** command, then you will not be able to configure ECMP multicast load splitting using the **ip multicast multipath** command.
- Prior to configuring ECMP multicast load splitting, it is best practice to use the **show ip rpf** command to validate whether sources can take advantage of IP multicast multipath capabilities.
- BGP does not install multiple equal-cost paths by default. Use the **maximum-paths** command to configure multipath (for example in BGP). For more information, see the "Use of BGP with ECMP Multicast Load Splitting" section.

Restrictions

The **ip multicast multipath** command does not support configurations in which the same PIM neighbor IP address is reachable through multiple equal-cost paths. This situation typically occurs if unnumbered interfaces are used. Use different IP addresses for all interfaces when configuring the **ip multicast multipath** command.

Enabling ECMP Multicast Load Splitting Based on Source Address Using the S-Hash Algorithm

Perform this task to enable ECMP multicast load splitting of multicast traffic based on source address (using the S-hash algorithm) to take advantage of multiple paths through the network. The S-hash algorithm is predictable because no randomization is used in calculating the hash value. The S-hash algorithm, however, is subject to polarization because for a given source, the same hash is always picked irrespective of the router the hash is being calculated on.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast multipath**
4. Repeat Steps 1 through 3 on all the routers in a redundant topology.
5. **end**

6. **show ip rpf** *source-address* [*group-address*]
7. **show ip route** *ip-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast multipath** **Example:** `Router(config)# ip multicast multipath`	Enables ECMP multicast load splitting based on source address using the S-hash algorithm. • Because this command changes the way an RPF neighbor is selected, it must be configured consistently on all routers in a redundant topology to avoid looping. **Note** Be sure to enable the **ip multicast multipath** command on the router that is supposed to be the *receiver* for traffic from more than one incoming interfaces, which is opposite to unicast routing. From the perspective of unicast, multicast is active on the *sending* router connecting to more than one outgoing interfaces.
Step 4	Repeat Steps 1 through 3 on all the routers in a redundant topology.	—
Step 5	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 6	**show ip rpf** *source-address* [*group-address*] **Example:** `Router# show ip rpf 10.1.1.2`	Displays information about how IP multicast routing does RPF. • Use this command to verify RPF selection so as to ensure that IP multicast traffic is being properly load split.
Step 7	**show ip route** *ip-address* **Example:** `Router# show ip route 10.1.1.2`	Displays the current state of the IP routing table. • Use this command to verify that there multiple paths available to a source or RP for ECMP multicast load splitting. • For the *ip-address* argument, enter the IP address of a source to validate that there are multiple paths available to the source (for shortest path trees) or the IP address of an RP to validate that there are multiple paths available to the RP (for shared trees).

Enabling ECMP Multicast Load Splitting Based on Source and Group Address Using the Basic S-G-Hash Algorithm

In Cisco IOS Release 12.2(33)SRB and subsequent 12.2SR releases, perform this task to enable ECMP multicast load splitting of multicast traffic based on source and group address (using the basic S-G-hash algorithm) to take advantage of multiple paths through the network. The basic S-G-hash algorithm is predictable because no randomization is used in calculating the hash value.The basic S-G-hash algorithm, however, is subject to polarization because for a given source and group, the same hash is always picked irrespective of the router the hash is being calculated on.

The basic S-G-hash algorithm provides more flexible support for ECMP multicast load splitting than the the S-hash algorithm. Using the basic S-G-hash algorithm for load splitting, in particular, enables multicast traffic from devices that send many streams to groups or that broadcast many channels, such as IPTV servers or MPEG video servers, to be more effectively load split across equal-cost paths.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast multipath s-g-hash basic**
4. Repeat Steps 1 through 3 on all the routers in a redundant topology.
5. **end**
6. **show ip rpf** *source-address* [*group-address*]
7. **show ip route** *ip-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast multipath s-g-hash basic** **Example:** `Router(config)# ip multicast multipath s-g-hash basic`	Enables ECMP multicast load splitting based on source and group address using the basic S-G-hash algorithm. • Because this command changes the way an RPF neighbor is selected, it must be configured consistently on all routers in a redundant topology to avoid looping. **Note** Be sure to enable the **ip multicast multipath** command on the router that is supposed to be the *receiver* for traffic from more than one incoming interfaces, which is opposite to unicast routing. From the perspective of unicast, multicast is active on the *sending* router connecting to more than one outgoing interfaces.

	Command or Action	Purpose
Step 4	Repeat Steps 1 through 3 on all the routers in a redundant topology.	—
Step 5	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 6	**show ip rpf** *source-address* [*group-address*] **Example:** `Router# show ip rpf 10.1.1.2`	Displays information about how IP multicast routing does RPF. • Use this command to verify RPF selection so as to ensure that IP multicast traffic is being properly load split.
Step 7	**show ip route** *ip-address* **Example:** `Router# show ip route 10.1.1.2`	Displays the current state of the IP routing table. • Use this command to verify that there multiple paths available to a source or RP for ECMP multicast load splitting. • For the *ip-address* argument, enter the IP address of a source to validate that there are multiple paths available to the source (for shortest path trees) or the IP address of an RP to validate that there are multiple paths available to the RP (for shared trees).

Enabling ECMP Multicast Load Splitting Based on Source, Group, and Next-Hop Address Using the Next-Hop-Based S-G-Hash Algorithm

In Cisco IOS Release 12.2(33)SRB and subsequent 12.2SR releases, perform this task to enable ECMP multicast load splitting of multicast traffic based on source, group, and next-hop address (using the next-hop-based S-G-hash algorithm) to take advantage of multiple paths through the network. The next-hop-based S-G-hash algorithm is predictable because no randomization is used in calculating the hash value. Unlike the S-hash and basic S-G-hash algorithms, the hash mechanism used by the next-hop-based S-G-hash algorithm is not subject to polarization.

The next-hop-based S-G-hash algorithm provides more flexible support for ECMP multicast load splitting than S-hash algorithm and eliminates the polarization problem. Using the next-hop-based S-G-hash algorithm for ECMP multicast load splitting enables multicast traffic from devices that send many streams to groups or that broadcast many channels, such as IPTV servers or MPEG video servers, to be more effectively load split across equal-cost paths.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast multipath s-g-hash next-hop-based**
4. Repeat Steps 1 through 3 on all the routers in a redundant topology.
5. **end**
6. **show ip rpf** *source-address* [*group-address*]
7. **show ip route** *ip-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast multipath s-g-hash next-hop-based** **Example:** `Router(config)# ip multicast multipath s-g-hash next-hop-based`	Enables ECMP multicast load splitting based on source, group, and next-hop-address using the next-hop-based S-G-hash algorithm. • Because this command changes the way an RPF neighbor is selected, it must be configured consistently on all routers in a redundant topology to avoid looping. **Note** Be sure to enable the **ip multicast multipath** command on the router that is supposed to be the *receiver* for traffic from more than one incoming interfaces, which is opposite to unicast routing. From the perspective of unicast, multicast is active on the *sending* router connecting to more than one outgoing interfaces.
Step 4	Repeat Steps 1 through 3 on all the routers in a redundant topology.	—
Step 5	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 6	**show ip rpf** *source-address* [*group-address*] **Example:** `Router# show ip rpf 10.1.1.2`	Displays information about how IP multicast routing does RPF. • Use this command to verify RPF selection so as to ensure that IP multicast traffic is being properly load split.
Step 7	**show ip route** *ip-address* **Example:** `Router# show ip route 10.1.1.2`	Displays the current state of the IP routing table. • Use this command to verify that there multiple paths available to a source or RP for ECMP multicast load splitting. • For the *ip-address* argument, enter the IP address of a source to validate that there are multiple paths available to the source (for shortest path trees) or the IP address of an RP to validate that there are multiple paths available to the RP (for shared trees).

Configuration Examples for Load Splitting IP Multicast Traffic over ECMP

This section provides the following configuration examples:

Enabling ECMP Multicast Load Splitting Based on Source Address Using the S-Hash Algorithm: Example

The following example shows how to enable ECMP multicast load splitting on a router based on source address using the S-hash algorithm:

```
ip multicast multipath
```

Enabling ECMP Multicast Load Splitting Based on Source and Group Address Using the Basic S-G-Hash Algorithm: Example

The following example shows how to enable ECMP multicast load splitting on a router based on source and group address using the basic S-G-hash algorithm:

```
ip multicast multipath s-g-hash basic
```

Enabling ECMP Multicast Load Splitting Based on Source, Group, and Next-Hop Address Using the Next-Hop-Based S-G-Hash Algorithm: Example

The following example shows how to enable ECMP multicast load splitting on a router based on source, group, and next-hop address using the next-hop-based S-G-hash algorithm:

```
ip multicast multipath s-g-hash next-hop-based
```

Additional References

The following sections provide references related to load splitting IP multicast traffic over ECMP.

Related Documents

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

Standards

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

MIBs

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

RFCs

RFC	Title
RFC 2362	*Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register on Cisco.com.	http://www.cisco.com/techsupport

Feature Information for Load Splitting IP Multicast Traffic over ECMP

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or Cisco IOS Releases 12.2(1) or 12.0(3)S or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the "IP Multicast Features Roadmap."

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Load Splitting IP Multicast Traffic over ECMP

Feature Name	Releases	Feature Information
IP Multicast Load Splitting—Equal Cost Multipath (ECMP) Using S, G and Next Hop	12.2(33)SRB	The IP Multicast Load Splitting—Equal Cost Multipath (ECMP) Using S, G and Next Hop feature introduces more flexible support for ECMP multicast load splitting by adding support for load splitting based on source and group address and on source, group, and next-hop address. This feature enables multicast traffic from devices that send many streams to groups or that broadcast many channels, such as IPTV servers or MPEG video servers, to be more effectively load split across equal-cost paths. Prior to the introduction of this feature, the Cisco IOS software only supported ECMP multicast load splitting based on source address, which restricted multicast traffic sent by a single source to multiple groups from being load split across equal-cost paths. In 12.2(33)SRB, this feature was introduced on the Cisco 7600 series router. The following sections provide information about this feature: • ECMP Multicast Load Splitting Based on Source and Group Address Using the Basic S-G-Hash Algorithm, page 5 • ECMP Multicast Load Splitting Based on Source, Group, and Next-Hop Address Using the Next-Hop-Based S-G-Hash Algorithm, page 7 The following commands were modified by this feature: **ip multicast multipath**, **show ip rpf**.

Configuring an Intermediate IP Multicast Helper Between Broadcast-Only Networks

First Published: February 11, 2008
Last Updated: February 11, 2008

When a multicast-capable internetwork is between two subnets with broadcast-only-capable hosts, you can convert broadcast traffic to IP multicast traffic at the first hop router and convert it back to broadcast traffic at the last hop router to deliver the packets to the broadcast clients. You can thus take advantage of the multicast capability of an intermediate IP multicast helper. Configuring an intermediate IP multicast helper allows the transport of broadcast packets across an IP multicast-enabled network, thereby preventing unnecessary replication at the intermediate routers.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Configuring an Intermediate IP Multicast Helper Between Broadcast-Only Networks" section on page 10.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Configuring an Intermediate IP Multicast Helper Between Broadcast-Only Networks

- You understand the concepts documented in the "IP Multicast Technology Overview" module.
- You have IP multicast configured in your network environment and your IP multicast network is between broadcast-only networks. See the "Configuring Basic IP Multicast" module for more information about configuring IP multicast.

Information About Configuring an Intermediate IP Multicast Helper Between Broadcast-Only Networks

To configure an intermediate IP multicast helper between broadcast-only networks, you should be familiar with the following concept:

Intermediate IP Multicast Helper Capability

An intermediate IP multicast helper allows the transport of broadcast packets across an IP multicast-enabled network, thereby preventing unnecessary replication at the intermediate routers.

When configuring an intermediate IP multicast helper between broadcast-only networks, you must configure the first hop router to convert broadcast traffic to IP multicast traffic and the last hop router to convert IP multicast traffic back to broadcast traffic.

How to Configure an Intermediate IP Multicast Helper Between Broadcast-Only Networks

This section contains the following procedures:

Configuring the First Hop Router to Convert Broadcast Traffic to IP Multicast Traffic

Perform this task to convert broadcast traffic to IP multicast traffic on the first hop router. The first hop router is on the border between the broadcast-only network and IP multicast network.

Prerequisites

- This task assumes that you have an IP multicast network configured between two broadcast-only networks. For more information about configuring IP multicast, see the "Configuring Basic IP Multicast" module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {**deny** | **permit**} **udp** {**any** | [**host**] *source-address source-wildcard*} [*operator* [*port*]] {**any** | [**host**] *destination-address destination-wildcard*} [*operator* [*port*]]
4. **interface** *type number*
5. **ip multicast helper-map broadcast** *group-address access-list*
6. **exit**
7. **ip forward-protocol udp** [*port*]
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* {**deny** | **permit**} **udp** {**any** | [**host**] *source-address source-wildcard*} [*operator* [*port*]] {**any** | [**host**] *destination-address destination-wildcard*} [*operator* [*port*]] **Example:** `Router# access-list 105 permit udp host 126.1.22.199 host 126.1.22.255 eq 4000`	Creates an extended IP access list to control which UDP broadcast packets are translated. **Note** For more information about creating IP access lists, see the "Creating an IP Access List and Applying It to an Interface" module in the *Cisco IOS Security Configuration Guide*.

	Command or Action	Purpose
Step 4	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Enters interface configuration mode. • For the *type* and *number* arguments, select an incoming interface on the first hop router that is receiving broadcast-only traffic
Step 5	**ip multicast helper-map broadcast** *group-address access-list* **Example:** `Router(config-if)# ip multicast helper-map broadcast 239.254.2.5 105`	Allows IP multicast routing in a multicast-capable internetwork between two broadcast-only internetworks. • In the configuration on the first hop router, the **ip multicast helper-map** command is used with the **broadcast** keyword and *group-address* argument to specify the traffic to be converted from broadcast to multicast. The multicast group address specified for the *group-address* argument is the address to which the converted traffic will be directed. • For the *access-list* argument, specify the name or number of the access list created in Step 3 of this task. **Note** The form of the **ip multicast helper-map** command used in the configuration of the first hop router is different from the form of the command used in the configuration of the last hop router. See the "Configuring an Intermediate IP Multicast Helper Between Broadcast-Only Networks: Example"section for an example of this task.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 7	**ip forward-protocol udp** [*port*] **Example:** `Router(config)# ip forward-protocol udp 4000`	Configures the forwarding of UDP broadcast messages destined for the specified port.
Step 8	**end** **Example:** `Router(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

Configuring the Last Hop Router to Convert the IP Multicast Traffic Back to Broadcast Traffic

Perform this task to convert the IP multicast traffic back to broadcast traffic on the last hop router. The last hop router is on the border between the intermediate IP multicast network and broadcast-only network.

Prerequisites

- This task assumes that you have an IP multicast network configured between two broadcast-only networks. For more information about configuring IP multicast, see the "Configuring Basic IP Multicast" module in the *Cisco IOS IP Multicast Configuration Guide*.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {**deny** | **permit**} **udp** {**any** | [**host**] *source-address source-wildcard*} [*operator* [*port*]] {**any** | [**host**] *destination-address destination-wildcard*} [*operator* [*port*]]
4. **interface** *type number*
5. **ip multicast helper-map** *group-address broadcast-address access-list*
6. **exit**
7. **interface** *type number*
8. **ip directed-broadcast**
9. **exit**
10. **ip forward-protocol udp** [*port*]
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**access-list** *access-list-number* {**deny** | **permit**} **udp** {**any** | [**host**] *source-address source-wildcard*} [*operator* [*port*]] {**any** | [**host**] *destination-address destination-wildcard*} [*operator* [*port*]] **Example:** `Router# access-list 105 permit udp host 126.1.22.199 host 126.1.22.255 eq 4000`	Creates an extended IP access list to control which UDP broadcast packets are translated. **Note** For more information about creating IP access lists, see the "Creating an IP Access List and Applying It to an Interface" module in the *Cisco IOS Security Configuration Guide*.
Step 4	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Enters interface configuration mode. • For the *type* and *number* arguments, select an incoming interface on the last hop router that is receiving IP multicast traffic.
Step 5	**ip multicast helper-map** *group-address broadcast-address access-list* **Example:** `Router(config-if)# ip multicast helper-map 239.254.2.5 126.1.28.255 105`	Allows IP multicast routing in a multicast-capable internetwork between two broadcast-only internetworks. • In the configuration on the last hop router, the **ip multicast helper-map** command is used with the *group-address* and *broadcast-address* arguments to specify the traffic to be converted from IP multicast to broadcast. The multicast group address specified for the *group-address* argument is the address of the traffic to be converted from IP multicast to broadcast. The broadcast address specified for the *broadcast-address* argument is the address to which the broadcast traffic will be sent. • For the *access-list* argument, specify the name or number of the access list created in Step 3 of this task. **Note** The form of the **ip multicast helper-map** command used in the configuration of the first hop router is different from the form of the command used in the configuration of the last hop router. See the "Configuring an Intermediate IP Multicast Helper Between Broadcast-Only Networks: Example"section for an example of this task.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration and returns to global configuration mode.
Step 7	**interface** *type number* **Example:** `Router(config)# interface ethernet 2`	Enters interface configuration mode. • For the *type* and *number* arguments, select an outgoing interface on the last hop router that is facing the destination broadcast-only subnet.
Step 8	**ip directed-broadcast** **Example:** `Router(config-if)# ip directed-broadcast`	Enables the translation of a directed broadcast to physical broadcasts.

	Command or Action	Purpose
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration and returns to global configuration mode.
Step 10	**ip forward-protocol udp** [*port*] **Example:** `Router(config)# ip forward-protocol udp 4000`	Configures the forwarding of UDP broadcast messages destined for the specified port.
Step 11	**end** **Example:** `Router(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

Configuration Examples for an Intermediate IP Multicast Helper Between Broadcast-Only Networks

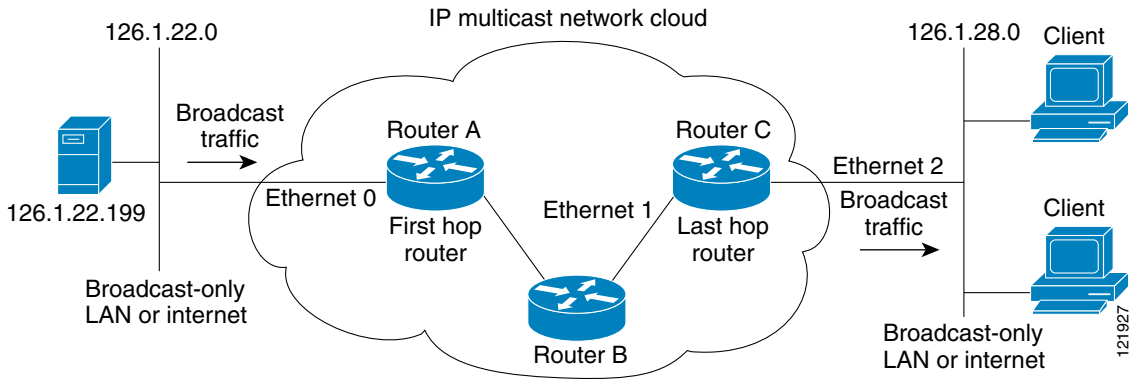
This section provides the following configuration example:

- "Configuring an Intermediate IP Multicast Helper Between Broadcast-Only Networks: Example" section on page 7

Configuring an Intermediate IP Multicast Helper Between Broadcast-Only Networks: Example

This example shows how to configure an intermediate IP multicast helper between broadcast-only networks. The topology used for this example is illustrated in Figure 1.

Figure 1 IP Multicast Helper Example Topology



In this example, a server on the LAN connected to Ethernet interface 0 of Router A is sending a UDP broadcast traffic with a source address of 126.1.22.199 and a destination address of 126.1.22.255:4000. The configuration on the first hop router converts the broadcast traffic arriving at incoming Ethernet interface 0 destined for UDP port 4000 to IP multicast traffic. The access list permits traffic being sent from the server at 126.1.22.199 being sent to 126.1.22.255:4000. The traffic is sent to group address 239.254.2.5. The **ip forward-protocol** command specifies the forwarding of broadcast messages destined for UDP port 4000.

Note This example primarily displays the configuration related to configuring an intermediate IP multicast helper. Protocol Independent Multicast-Sparse Mode (PIM-SM) is the multicast protocol used in this example. PIM-SM requires the use of a rendezvous point (RP). For more information about configuring RPs, see the "Configuring Basic IP Multicast" module in the Cisco IOS IP Multicast Configuration Guide.

The configuration on the last hop router converts the IP multicast traffic at incoming Ethernet interface 1 back to broadcast at outgoing Ethernet interface 2. Again, not all multicast traffic emerging from the multicast network should be converted from multicast to broadcast, only the traffic destined for 126.1.22.255:4000.

The configurations for Router A and Router C are as follows:

Router A—First Hop Router Configuration

```
interface ethernet 0
 ip address 126.1.22.1 255.255.255.0
 ip pim sparse-mode
 ip multicast helper-map broadcast 239.254.2.5 105
access-list 105 permit udp host 126.1.22.199 host 126.1.22.255 eq 4000
ip forward-protocol udp 4000
```

Router C—Last Hop Router Configuration

```
interface ethernet 1
 ip address 126.1.26.1 255.255.255.0
 ip pim sparse-mode
 ip multicast helper-map 239.254.2.5 126.1.28.255 105
!
interface ethernet 2
 ip address 126.1.28.1 255.255.255.0
 ip directed-broadcast
access-list 105 permit udp host 126.1.22.199 any eq 4000
ip forward-protocol udp 4000
```

Additional References

The following sections provide references related to configuring an intermediate IP multicast helper between broadcast-only networks.

Related Documents

Related Topic	Document Title
Basic IP multicast concepts, configuration tasks, and examples	"Configuring Basic IP Multicast" module
Overview of the IP multicast technology area	"IP Multicast TechnologyOverview" module
IP multicast commands: complete command syntax, command mode, command history, command defaults, usage guidelines, and examples	*Cisco IOS IP Multicast Command Reference*

Standards

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

MIBs

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

RFCs

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

Technical Assistance

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

Feature Information for Configuring an Intermediate IP Multicast Helper Between Broadcast-Only Networks

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Releases 12.2(1) or 12.0(3)S or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the "IP Multicast Features Roadmap."

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Configuring an IP Multicast Helper Between Broadcast-Only Networks

Feature Name	Releases	Feature Information
This table is intentionally left blank because no features were introduced or modified in this module since Cisco IOS Release 12.2(1). This table will be updated when feature information is added to this module.	—	—

Constraining IP Multicast in a Switched Ethernet Network

This module describes how to configure routers to use the Cisco Group Management Protocol (CGMP) in switched Ethernet networks to control multicast traffic to Layer 2 switch ports and the Router-Port Group Management Protocol (RGMP) to constrain IP multicast traffic on router-only network segments.

The default behavior for a Layer 2 switch is to forward all multicast traffic to every port that belongs to the destination LAN on the switch. This behavior reduces the efficiency of the switch, whose purpose is to limit traffic to the ports that need to receive the data. This behavior requires a constraining mechanism to reduce unnecessary multicast traffic, which improves switch performance.

Module History

This module was first published on May 2, 2005, and last updated on May 2, 2005.

Finding Feature Information in This Document

Not all features may be supported in your Cisco IOS software release. Use the "Feature Information for Constraining IP Multicast in a Switched Ethernet Network" to find information about feature support and configuration.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Constraining IP Multicast in a Switched Ethernet Network

Before using the tasks in this module, you should be familiar with the concepts described in the "IP Multicast Technology Overview" module.

Information About IP Multicast in a Switched Ethernet Network

Before you perform the tasks in this module, you should understand the following concepts:

IP Multicast Traffic and Layer 2 Switches

The default behavior for a Layer 2 switch is to forward all multicast traffic to every port that belongs to the destination LAN on the switch. This behavior reduces the efficiency of the switch, whose purpose is to limit traffic to the ports that need to receive the data. This behavior requires a constraining mechanism to reduce unnecessary multicast traffic, which improves switch performance.

Three methods that efficiently constrain IP multicast in a Layer 2 switching environment are described in the following sections:

Note CGMP and IGMP snooping are used on subnets that include end users or receiver clients. RGMP is used on routed segments that contain only routers, such as in a collapsed backbone.

RGMP and CGMP cannot interoperate. However, Internet Group Management Protocol (IGMP) can interoperate with CGMP and RGMP snooping.

CGMP on Catalyst Switches for IP Multicast

CGMP is a Cisco-developed protocol used on routers connected to Catalyst switches to perform tasks similar to those performed by IGMP. CGMP is necessary for those Catalyst switches that do not distinguish between IP multicast data packets and IGMP report messages, both of which are addressed to the same group address at the MAC level. The switch can distinguish IGMP packets, but would need to use software on the switch, greatly impacting its performance.

You must configure CGMP on the multicast routers and the Layer 2 switches. The result is that, with CGMP, IP multicast traffic is delivered only to those Catalyst switch ports that are attached to interested receivers. All other ports that have not explicitly requested the traffic will not receive it unless these ports are connected to a multicast router. Multicast router ports must receive every IP multicast data packet.

Using CGMP, when a host joins a multicast group, it multicasts an unsolicited IGMP membership report message to the target group. The IGMP report is passed through the switch to the router for normal IGMP processing. The router (which must have CGMP enabled on this interface) receives the IGMP report and processes it as it normally would, but also creates a CGMP Join message and sends it to the switch. The Join message includes the MAC address of the end station and the MAC address of the group it has joined.

The switch receives this CGMP Join message and then adds the port to its content-addressable memory (CAM) table for that multicast group. All subsequent traffic directed to this multicast group is then forwarded out the port for that host.

The Layer 2 switches are designed so that several destination MAC addresses could be assigned to a single physical port. This design allows switches to be connected in a hierarchy and also allows many multicast destination addresses to be forwarded out a single port.

The router port also is added to the entry for the multicast group. Multicast routers must listen to all mulitcast traffic for every group because IGMP control messages are also sent as multicast traffic. The rest of the multicast traffic is forwarded using the CAM table with the new entries created by CGMP.

IGMP Snooping

IGMP snooping is an IP multicast constraining mechanism that runs on a Layer 2 LAN switch. IGMP snooping requires the LAN switch to examine, or “snoop,” some Layer 3 information (IGMP Join/Leave messages) in the IGMP packets sent between the hosts and the router. When the switch receives the IGMP host report from a host for a particular multicast group, the switch adds the port number of the host to the associated multicast table entry. When the switch hears the IGMP Leave group message from a host, the switch removes the table entry of the host.

Because IGMP control messages are sent as multicast packets, they are indistinguishable from multicast data at Layer 2. A switch running IGMP snooping must examine every multicast data packet to determine if it contains any pertinent IGMP control information. IGMP snooping implemented on a low-end switch with a slow CPU could have a severe performance impact when data is sent at high rates. The solution is to implement IGMP snooping on high-end switches with special application-specific integrated circuits (ASICs) that can perform the IGMP checks in hardware. CGMP is a better option for low-end switches without special hardware.

Router-Port Group Management Protocol (RGMP)

CGMP and IGMP snooping are IP multicast constraining mechanisms designed to work on routed network segments that have active receivers. They both depend on IGMP control messages that are sent between the hosts and the routers to determine which switch ports are connected to interested receivers.

Switched Ethernet backbone network segments typically consist of several routers connected to a switch without any hosts on that segment. Because routers do not generate IGMP host reports, CGMP and IGMP snooping will not be able to constrain the multicast traffic, which will be flooded to every port on the VLAN. Routers instead generate Protocol Independent Multicast (PIM) messages to Join and Prune multicast traffic flows at a Layer 3 level.

Router-Port Group Management Protocol (RGMP) is an IP multicast constraining mechanism for router-only network segments. RGMP must be enabled on the routers and on the Layer 2 switches. A multicast router indicates that it is interested in receiving a data flow by sending an RGMP Join message for a particular group. The switch then adds the appropriate port to its forwarding table for that multicast group—similar to the way it handles a CGMP Join message. IP multicast data flows will be forwarded only to the interested router ports. When the router no longer is interested in that data flow, it sends an RGMP Leave message and the switch removes the forwarding entry.

If there are any routers that are not RGMP-enabled, they will continue to receive all multicast data.

How to Constrain Multicast in a Switched Ethernet Network

This section describes the following tasks:

- Configuring Switches for IP Multicast, page 4
- Configuring IGMP Snooping, page 4
- Enabling CGMP on a Router, page 4 (optional)
- Configuring IP Multicast in a Layer 2 Switched Ethernet Network, page 5 (optional)

Configuring Switches for IP Multicast

If you have switching in your multicast network, consult the documentation for the switch you are working with for information about how to configure IP multicast.

Configuring IGMP Snooping

No configuration is required on the router. Consult the documentation for the switch you are working with to determine how to enable IGMP snooping and follow the provided instructions.

Enabling CGMP on a Router

CGMP is a protocol used on routers connected to Catalyst switches to perform tasks similar to those performed by IGMP. CGMP is necessary because the Catalyst switch cannot distinguish between IP multicast data packets and IGMP report messages, which are both at the MAC level and are addressed to the same group address.

Restrictions

- CGMP should be enabled only on 802 or ATM media, or LAN emulation (LANE) over ATM.
- CGMP should be enabled only on routers connected to Catalyst switches.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*

4. **ip cgmp** [**proxy** | **router-only**]
5. **end**
6. **clear ip cgmp** [*interface-type interface-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Selects an interface that is connected to hosts on which IGMPv3 can be enabled.
Step 4	**ip cgmp** [**proxy**	**router-only**] **Example:** `Router(config-if)# ip cgmp proxy`	Enables CGMP on an interface of a router connected to a Cisco Catalyst 5000 family switch. • The **proxy** keyword enables the CGMP proxy function. When enabled, any router that is not CGMP-capable will be advertised by the proxy router. The proxy router advertises the existence of other non-CGMP-capable routers by sending a CGMP Join message with the MAC address of the non-CBMP-capable router and group address of 0000.0000.0000.
Step 5	`end` **Example:** `Router(config-if)# end`	Ends the current configuration session and returns to EXEC mode.
Step 6	**clear ip cgmp** [*interface-type interface-number*] **Example:** `Router# clear ip cgmp`	(Optional) Clears all group entries from the caches of Catalyst switches.

Configuring IP Multicast in a Layer 2 Switched Ethernet Network

Perform this task to configure IP multicast in a Layer 2 Switched Ethernet network using RGMP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **interface** *type number*
4. ip rgmp
5. end
6. debug ip rgmp
7. **show ip igmp interface**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Selects an interface that is connected to hosts.
Step 4	`ip rgmp` **Example:** `Router(config-if)# ip rgmp`	Enables RGMP on Ethernet, Fast Ethernet, and Gigabit Ethernet interfaces.
Step 5	`end` **Example:** `Router(config-if)# end`	Ends the current configuration session and returns to EXEC mode.
Step 6	`debug ip rgmp` **Example:** `Router# debug ip rgmp`	(Optional) Logs debug messages sent by an RGMP-enabled router.
Step 7	`show ip igmp interface` **Example:** `Router# show ip igmp interface`	(Optional) Displays multicast-related information about an interface.

Configuration Examples for Constraining IP Multicast in a Switched Ethernet Network

This section provides the following configuration examples:

CGMP Configuration: Example

The following example is for a basic network environment where multicast source(s) and multicast receivers are in the same VLAN. The desired behavior is that the switch will constrain the multicast forwarding to those ports that request the multicast stream.

A 4908G-L3 router is connected to the Catalyst 4003 on port 3/1 in VLAN 50. The following configuration is applied on the GigabitEthernet1 interface. Note that there is no **ip multicast-routing** command configured because the router is not routing multicast traffic across its interfaces.

```
interface GigabitEthernet1
 ip address 192.168.50.11 255.255.255.0
 ip pim dense-mode
 ip cgmp
```

RGMP Configuration: Example

The following example shows how to configure RGMP on a router:

```
ip multicast-routing
ip pim sparse-mode
interface ethernet 0
 ip rgmp
```

Additional References

The following sections provide references related to constraining IP multicast in a switched Ethernet network.

Related Documents

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

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 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/public/support/tac/home.shtml

Feature Information for Constraining IP Multicast in a Switched Ethernet Network

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Releases 12.2(1) or later appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for specific commands was introduced, see the command reference documents.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator (http://www.cisco.com/go/fn). You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click Cancel at the login dialog box and follow the instructions that appear.

Table 1 Feature Information for Constraining IP Multicast in a Switched Ethernet Network

Feature Name	Releases	Feature Configuration Information
Cisco IOS	—	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.
CGMP - Cisco Group Management Protocol	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Configuring IP Multicast over Unidirectional Links

IP multicast requires bidirectional communication, yet some networks include broadcast satellite links, which are unidirectional. Unidirectional link routing (UDLR) provides three mechanisms for a router to emulate a bidirectional link to enable the routing of unicast and multicast packets over a physical unidirectional interface, such as a broadcast satellite link. The mechanisms are a UDLR tunnel, Internet Group Management Protocol (IGMP) UDLR, and IGMP proxy. This document describes a UDLR tunnel and IGMP UDLR. IGMP proxy is described in the "Customizing IGMP" module. The three mechanisms may be used independently or in combination.

Module History

This module was first published on May 2, 2005, and last updated on May 2, 2005.

Finding Feature Information in This Document

Use the "Feature Information for Configuring IP Multicast over Unidirectional Links" section to find information about feature support and configuration.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for UDLR

This module assumes you have met the following prerequisites:

- You understand the concepts in the "IP Multicast Technology Overview" module.
- You have IP multicast configured in your network. Refer to the "Configuring Basic IP Multicast" module.

Information About UDLR

Before you configure unidirectional link routing, you should understand the following concepts:

UDLR Overview

Both unicast and multicast routing protocols forward data on interfaces from which they have received routing control information. This model requires a bidirectional link. However, some network links are unidirectional. For networks that are unidirectional (such as broadcast satellite links), a method of communication that allows for control information to operate in a unidirectional environment is necessary. (Note that IGMP is not a routing protocol.)

Specifically, in unicast routing, when a router receives an update message on an interface for a prefix, it forwards data for destinations that match that prefix out that same interface. This is the case in distance vector routing protocols. Similarly, in multicast routing, when a router receives a Join message for a multicast group on an interface, it forwards copies of data destined for that group out that same interface. Based on these principles, unicast and multicast routing protocols cannot be supported over UDLs without the use of UDLR. UDLR is designed to enable the operation of routing protocols over UDLs without changing the routing protocols themselves.

UDLR enables a router to emulate the behavior of a bidirectional link for IP operations over UDLs. UDLR has three complementary mechanisms for bidirectional link emulation, which are described in the following sections:

- UDLR Tunnel—A mechanism for routing unicast and multicast traffic.
- Internet Group Management Protocol (IGMP) UDLR—Mechanism for routing multicast traffic. This method scales well for many broadcast satellite links.
- IGMP Proxy—Mechanism for routing multicast traffic.

You can use each mechanism independently or in conjunction with the others. IGMP proxy is described in the "Customizing IGMP" module.

UDLR Tunnel

The UDLR tunnel mechanism enables IP and its associated unicast and multicast routing protocols to treat the unidirectional link (UDL) as being logically bidirectional. A packet that is destined on a receive-only interface is picked up by the UDLR tunnel mechanism and sent to an upstream router using

a generic routing encapsulation (GRE) tunnel. The control traffic flows in the opposite direction of the user data flow. When the upstream router receives this packet, the UDLR tunnel mechanism makes it appear that the packet was received on a send-only interface on the UDL.

The purpose of the unidirectional GRE tunnel is to move control packets from a downstream node to an upstream node. The one-way tunnel is mapped to a one-way interface (that goes in the opposite direction). Mapping is performed at the link layer, so the one-way interface appears bidirectional. When the upstream node receives packets over the tunnel, it must make the upper-layer protocols act as if the packets were received on the send-capable UDL.

A UDLR tunnel supports the following functionality:

- Address Resolution Protocol (ARP) and Next Hop Resolution Protocol (NHRP) over a UDL
- Emulation of bidirectional links for all IP traffic (as opposed to only control-only broadcast/multicast traffic)
- Support for IP GRE multipoint at a receive-only tunnel

Note A UDL router can have many routing peers (for example, routers interconnected via a broadcast satellite link). As with bidirectional links, the number of peer routers a router has must be kept relatively small to limit the volume of routing updates that must be processed. For multicast operation, we recommend using the IGMP UDLR mechanism when interconnecting more than 20 routers.

IGMP UDLR

In addition to a UDLR tunnel, another mechanism that enables support of multicast routing protocols over UDLs is using IP multicast routing with IGMP, which accommodates UDLR. This mechanism scales well for many broadcast satellite links.

With IGMP UDLR, an upstream router sends periodic queries for members on the UDL. The queries include a unicast address of the router that is not the unicast address of the unidirectional interface. The downstream routers forward IGMP reports received from directly connected members (on interfaces configured to helper forward IGMP reports) to the upstream router. The upstream router adds the unidirectional interface to the (*, G) outgoing interface list, thereby enabling multicast packets to be forwarded down the UDL.

In a large enterprise network, it is not possible to be able to receive IP multicast traffic via satellite and forward the traffic throughout the network. This limitation exists because receiving hosts must be directly connected to the downstream router. However, you can use the IGMP proxy mechanism to overcome this limitation. Refer to the “Customizing IGMP” module for more information on this mechanism.

How to Route IP Multicast over Unidirectional Links

This section includes the following procedures. You can do either or both in your network. If you want to configure IGMP Proxy, refer to the “Customizing IGMP” module.

- Configuring a UDLR Tunnel, page 4 (optional)
- Configuring IGMP UDLR, page 6 (optional)

Configuring a UDLR Tunnel

To configure a UDLR tunnel, perform the task in this section. The tunnel mode defaults to GRE. You need not assign an IP address to the tunnel (you need not use the **ip address** or **ip unnumbered** commands). You must configure the tunnel endpoint addresses.

You must configure both the upstream and downstream routers to meet the following conditions:

- On the upstream router, where the UDL can only send, you must configure the tunnel to receive. When packets are received over the tunnel, the upper-layer protocols treat the packet as though it is received over the unidirectional, send-only interface.
- On the downstream router, where the UDL can only receive, you must configure the tunnel to send. When packets are sent by upper-layer protocols over the interface, they will be redirected and sent over this GRE tunnel.

Prerequisite

Before configuring UDLR tunnel, ensure that all routers on the UDL have the same subnet address. If all routers on the UDL cannot have the same subnet address, the upstream router must be configured with secondary addresses to match all the subnets that the downstream routers are attached to.

SUMMARY STEPS

On the Upstream Router:

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **interface tunnel** *number*
5. **tunnel udlr receive-only** *type number*
6. **tunnel source** {*ip-address* |*type number*}
7. **tunnel destination** {*hostname* | *ip-address*}
8. Move to the downstream router.

On the Downstream Router:

9. **enable**
10. **configure terminal**
11. **interface** *type number*
12. **interface tunnel** *number*
13. **tunnel udlr send-only** *type number*
14. **tunnel source** {*ip-address* |*type number*}
15. **tunnel destination** {*hostname* | *ip-address*}
16. **tunnel udlr address-resolution**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted. • Do this step on the upstream router.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Configures the unidirectional send-only interface.
Step 4	**interface tunnel** *number* **Example:** `Router(config-if)# interface tunnel 0`	Configures the receive-only tunnel interface.
Step 5	**tunnel udlr receive-only** *type number* **Example:** `Router(config-if)# tunnel udlr receive-only ethernet 0`	Configures the UDLR tunnel. • Use the same *type* and *number* values as the unidirectional send-only interface *type* and *number* values specified with the **interface** *type number* command in Step 3.
Step 6	**tunnel source** {ip-address | *type number*} **Example:** `Router(config-if)# tunnel source 10.3.4.5`	Configures the tunnel source.
Step 7	**tunnel destination** {*hostname* | ip-address} **Example:** `Router(config-if)# tunnel destination 11.8.2.3`	Configures the tunnel destination.
Step 8	Move to the downstream router.	—
Step 9	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 10	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 11	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Configures the unidirectional receive-only interface.
Step 12	**interface tunnel** *number* **Example:** `Router(config-if)# interface tunnel 0`	Configures the send-only tunnel interface.
Step 13	**tunnel udlr send-only** *type number* **Example:** `Router(config-if)# tunnel udlr send-only ethernet 0`	Configures the UDLR tunnel. • Use the same *type* and *number* values as the unidirectional receive-only interface *type* and *number* values specified with the **interface** *type number* command in Step 3.
Step 14	**tunnel source** {ip-address | *type number*} **Example:** `Router(config-if)# tunnel source 11.8.2.3`	Configures the tunnel source.
Step 15	**tunnel destination** {*hostname* | ip-address} **Example:** `Router(config-if)# tunnel destination 10.3.4.5`	Configures the tunnel destination.
Step 16	**tunnel udlr address-resolution** **Example:** `Router(config-if)# tunnel udlr address-resolution`	Enables the forwarding of ARP and NHRP.

Configuring IGMP UDLR

To configure an IGMP UDL, you must configure both the upstream and downstream routers. You need not specify whether the direction is sending or receiving; IGMP learns the direction by the nature of the physical connection.

When the downstream router receives an IGMP report from a host, the router sends the report to the IGMP querier associated with the UDL interface identified in the **ip igmp helper-address** command.

Distance for the Default RPF Interface Determines Which Path Is Used

By default, the distance for the default reverse path forwarding (RPF) interface is 15. Any explicit sources learned by routing protocols will take preference if their distance is less than the distance configured by the **ip multicast default-rpf-distance** command. Use this command on downstream routers if you want some sources to use RPF to reach the UDLR link and others to use the terrestrial paths.

- If you want IGMP to prefer the UDL, set the distance to be less than the distances of the unicast routing protocols.

- If you want IGMP to prefer the non-UDL, set the distance to be greater than the distances of the unicast routing protocols.

Prerequisites

Before configuring IGMP UDLR, ensure that the following conditions exist:

- All routers on the UDL have the same subnet address. If all routers on the UDL cannot have the same subnet address, the upstream router must be configured with secondary addresses to match all the subnets that the downstream routers are attached to.
- Multicast receivers are directly connected to the downstream routers.

SUMMARY STEPS

On the Upstream Router:

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip igmp unidirectional-link**
5. Move to the downstream router.

On the Downstream Router:

6. **enable**
7. **configure terminal**
8. **ip multicast default-rpf-distance** *distance*
9. **interface** *type number*
10. **ip igmp unidirectional-link**
11. **ip igmp helper-address udl** *type number*
12. **exit**
13. **show ip igmp udlr** [*group-name* | *group-address* | *type number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted. • Begin on the upstream router.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Configures the interface.
Step 4	**ip igmp unidirectional-link** **Example:** `Router(config-if)# ip igmp unidirectional-link`	Configures IGMP on the interface to be unidirectional.
Step 5	Move to the downstream router.	—
Step 6	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted. • Begin on the upstream router.
Step 7	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 8	**ip multicast default-rpf-distance** *distance* **Example:** `Router# ip multicast default-rpf-distance 10`	(Optional) Sets the distance for the default RPF interface.
Step 9	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Configures the interface.
Step 10	**ip igmp unidirectional-link** **Example:** `Router(config-if)# ip igmp unidirectional-link`	Configures IGMP on the interface to be unidirectional.
Step 11	**ip igmp helper-address udl** *type number* **Example:** `Router(config-if)# ip igmp helper-address udl ethernet 0`	Configures the interface to be an IGMP helper. • Use this command on every downstream router, on every interface to specify the *type* and *number* values that identify the UDL interface.
Step 12	**exit** **Example:** `Router(config-if)# exit`	Exits configuration mode and returns to EXEC mode.
Step 13	**show ip igmp udlr** [*group-name* | *group-address* | *type number*] **Example:** `Router(config)# show ip igmp udlr`	(Optional) Displays UDLR information for directly connected multicast groups on interfaces that have a UDL helper address configured.

Configuration Examples for UDLR

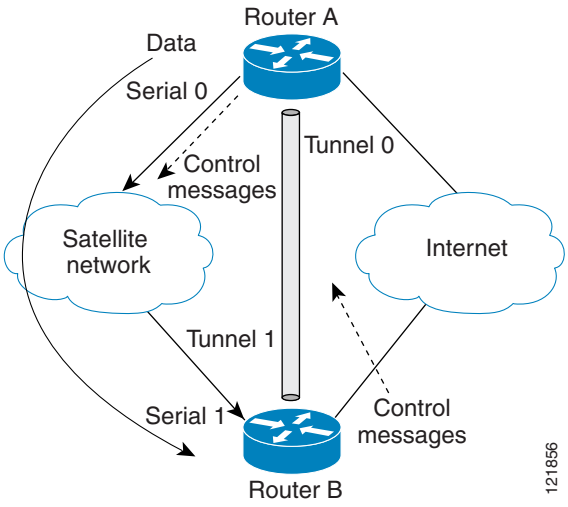
This section includes the following examples:

UDLR Tunnel: Example

The following example shows how to configure a UDLR tunnel. In the example, Router A (the upstream router) is configured with Open Shortest Path First (OSPF) and PIM. Serial interface 0 has send-only capability. Therefore, the UDLR tunnel is configured as receive only, and points to serial 0.

Router B (the downstream router) is configured with OSPF and PIM. Serial interface 1 has receive-only capability. Therefore, the UDLR tunnel is configured as send-only, and points to serial 1. The forwarding of ARP and NHRP is enabled. Figure 1 illustrates the example.

Figure 1 UDLR Tunnel Example



Router A Configuration

```
ip multicast-routing
!
! Serial0 has send-only capability
!
interface serial 0
 encapsulation hdlc
 ip address 10.1.0.1 255.255.0.0
 ip pim sparse-dense-mode
!
! Configure tunnel as receive-only UDLR tunnel.
!
interface tunnel 0
 tunnel source 11.0.0.1
 tunnel destination 11.0.0.2
```

```
 tunnel udlr receive-only serial 0
!
! Configure OSPF.
!
router ospf
 network 10.0.0.0 0.255.255.255 area 0
```

Router B Configuration

```
ip multicast-routing
!
! Serial1 has receive-only capability
!
interface serial 1
 encapsulation hdlc
 ip address 10.1.0.2 255.255.0.0
 ip pim sparse-dense-mode
!
! Configure tunnel as send-only UDLR tunnel.
!
interface tunnel 0
 tunnel source 11.0.0.2
 tunnel destination 11.0.0.1
 tunnel udlr send-only serial 1
 tunnel udlr address-resolution
!
! Configure OSPF.
!
router ospf
 network 10.0.0.0 0.255.255.255 area 0
```

IGMP UDLR: Example

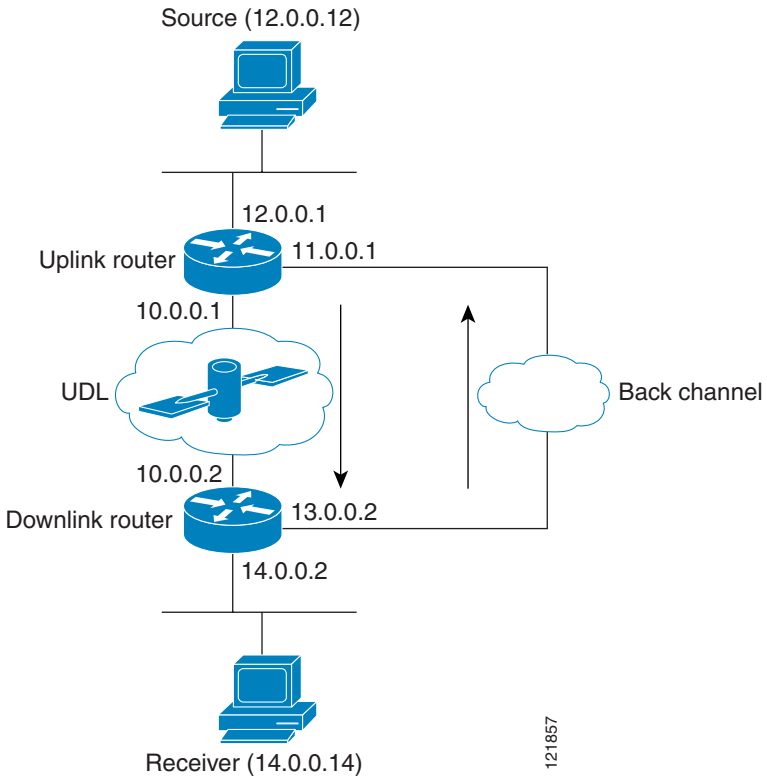
The following example shows how to configure IGMP UDLR. In this example, uplink-rtr is the local upstream router and downlink-rtr is the downstream router. Figure 2 illustrates the example.

Both routers are also connected to each other by a back channel connection. Both routers have two IP addresses: one on the UDL and one on the interface that leads to the back channel. The back channel is any return route and can have any number of routers.

Note Configuring PIM on the back channel interfaces on the uplink router and downlink router is optional.

All routers on a UDL must have the same subnet address. If all routers on a UDL cannot have the same subnet address, the upstream router must be configured with secondary addresses to match all the subnets that the downstream routers are attached to.

Figure 2 IGMP Unidirectional Link Routing Example



Uplink Router (uplink-rtr) Configuration

```
ip multicast-routing
!
! Interface that source is attached to
!
interface ethernet 0
 description Typical IP multicast enabled interface
 ip address 12.0.0.1 255.0.0.0
 ip pim sparse-dense-mode
!
! Back channel
!
interface ethernet 1
 description Back channel which has connectivity to downlink-rtr
 ip address 11.0.0.1 255.0.0.0
 ip pim sparse-dense-mode
!
! Unidirectional link
!
interface serial 0
 description Unidirectional to downlink-rtr
 ip address 10.0.0.1 255.0.0.0
 ip pim sparse-dense-mode
 ip igmp unidirectional-link
 no keepalive
```

Downlink Router (downlink-rtr) Configuration

```
ip multicast-routing
```

```
!
! Interface that receiver is attached to, configure for IGMP reports to be
! helpered for the unidirectional interface.
!
interface ethernet 0
 description Typical IP multicast-enabled interface
 ip address 14.0.0.2 255.0.0.0
 ip pim sparse-dense-mode
 ip igmp helper-address udl serial 0
!
! Back channel
!
interface ethernet 1
 description Back channel that has connectivity to downlink-rtr
 ip address 13.0.0.2 255.0.0.0
 ip pim sparse-dense-mode
!
! Unidirectional link
!
interface serial 0
 description Unidirectional to uplink-rtr
 ip address 10.0.0.2 255.0.0.0
 ip pim sparse-dense-mode
 ip igmp unidirectional-link
 no keepalive
```

Integrated UDLR Tunnel, IGMP UDLR, and IGMP Proxy: Example

The following example shows how to configure UDLR tunnels, IGMP UDLR, and IGMP proxy on both the upstream and downstream routers sharing a UDL.

Upstream Configuration

```
ip multicast-routing
!
interface Tunnel0
  ip address 9.1.89.97 255.255.255.252
  no ip directed-broadcast
  tunnel source 9.1.89.97
  tunnel mode gre multipoint
  tunnel key 5
  tunnel udlr receive-only Ethernet2/3
!
interface Ethernet2/0
  no ip address
  shutdown
!
! user network
interface Ethernet2/1
  ip address 9.1.89.1 255.255.255.240
  no ip directed-broadcast
  ip pim dense-mode
  ip cgmp
  fair-queue 64 256 128
  no cdp enable
  ip rsvp bandwidth 1000 100
!
interface Ethernet2/2
  ip address 9.1.95.1 255.255.255.240
  no ip directed-broadcast
!
```

```
! physical send-only interface
interface Ethernet2/3
  ip address 9.1.92.100 255.255.255.240
  no ip directed-broadcast
  ip pim dense-mode
  ip nhrp network-id 5
  ip nhrp server-only
  ip igmp unidirectional-link
  fair-queue 64 256 31
  ip rsvp bandwidth 1000 100
!
router ospf 1
  network 9.1.92.96 0.0.0.15 area 1
!
ip classless
ip route 9.1.90.0 255.255.255.0 9.1.92.99
```

Downstream Configuration

```
ip multicast-routing
!
interface Loopback0
  ip address 9.1.90.161 255.255.255.252
  ip pim sparse-mode
  ip igmp helper-address udl Ethernet2/3
  ip igmp proxy-service
!
interface Tunnel0
  ip address 9.1.90.97 255.255.255.252
  ip access-group 120 out
  no ip directed-broadcast
  no ip mroute-cache
  tunnel source 9.1.90.97
  tunnel destination 9.1.89.97
  tunnel key 5
  tunnel udlr send-only Ethernet2/3
  tunnel udlr address-resolution
!
interface Ethernet2/0
  no ip address
  no ip directed-broadcast
  shutdown
  no cdp enable
!
! user network
interface Ethernet2/1
  ip address 9.1.90.1 255.255.255.240
  no ip directed-broadcast
  ip pim sparse-mode
  ip igmp mroute-proxy Loopback0
  no cdp enable
!
! Backchannel
interface Ethernet2/2
  ip address 9.1.95.3 255.255.255.240
  no ip directed-broadcast
  no cdp enable
!
! physical receive-only interface
interface Ethernet2/3
  ip address 9.1.92.99 255.255.255.240
  no ip directed-broadcast
  ip pim sparse-mode
  ip igmp unidirectional-link
```

```
  no keepalive
  no cdp enable
!
router ospf 1
  network 9.1.90.0 0.0.0.255 area 1
  network 9.1.92.96 0.0.0.15 area 1
!
ip classless
ip route 0.0.0.0 0.0.0.0 9.1.95.1
! set rpf to be the physical receive-only interface
ip mroute 0.0.0.0 0.0.0.0 9.1.92.96
ip pim rp-address 9.1.90.1
!
! permit ospf, ping and rsvp, deny others
access-list 120 permit icmp any any
access-list 120 permit 46 any any
access-list 120 permit ospf any any
```

Additional References

The following sections provide references related to UDLR.

Related Documents

Related Topic	Document Title
IP multicast commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Multicast Command Reference*
Tunnel interfaces	"Implementing Tunnels"module
IGMP and IGMP Proxy	"Customizing IGMP"module

MIBs

MIB	MIBs Link
• None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 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/public/support/tac/home.shtml

Feature Information for Configuring IP Multicast over Unidirectional Links

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Releases 12.2(1) or later appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for specific commands was introduced, see the command reference documents.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator (http://www.cisco.com/go/fn). You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click Cancel at the login dialog box and follow the instructions that appear.

Table 1 Feature Information for Configuring IP Multicast over Unidirectional Links

Feature Name	Releases	Feature Configuration Information
This table is intentionally left blank because no features were introduced or modified in Cisco IOS Release 12.2(1) or later. This table will be updated when feature information is added to this module.	—	—

Using the Multicast Routing Monitor

The Multicast Routing Monitor (MRM) is a management diagnostic tool that provides network fault detection and isolation in a large multicast routing infrastructure. It is designed to notify a network administrator of multicast routing problems in a test environment.

Module History

This module was first published on May 2, 2005, and last updated on May 2, 2005.

Finding Feature Information in This Document

Not all features may be supported in your Cisco IOS software release. Use the "Feature Information for Using the Multicast Routing Monitor" to find information about feature support and configuration.

Contents

Restrictions for Using the Multicast Routing Monitor

You must make sure the underlying multicast forwarding network being tested has no access lists or boundaries that deny the MRM data and control traffic. Specifically, consider the following factors:

- MRM test data are User Datagram Protocol (UDP) and Real-Time Transport Protocol (RTP) packets addressed to the configured multicast group address.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

- MRM control traffic between the Test Sender, Test Receiver, and Manager is addressed to the 224.0.1.111 multicast group, which all three components join. The 224.0.1.111 group is an IANA-registered group.
- Take into account the unicast IP addresses of sources and receivers when considering what could prevent control traffic flowing.

Information About the Multicast Routing Monitor

Before using MRM, you should understand the following concepts:

- Multicast Routing Monitor Operation, page 2
- Benefits of Multicast Routing Monitor, page 2

Multicast Routing Monitor Operation

MRM has three components that play different roles: the Manager, the Test Sender, and the Test Receiver. To test a multicast environment using test packets, perhaps before an upcoming multicast event, you need all three components.

You create a test based on various test parameters, name the test, and start the test. The test runs in the background and the command prompt returns.

If the Test Receiver detects an error (such as packet loss or duplicate packets), it sends an error report to the router configured as the Manager. The Manager immediately displays the error report. (The **show ip mrm status-report** command also displays error reports, if any.) You then troubleshoot your multicast environment as normal, perhaps using the **mtrace** command from the source to the Test Receiver. If the **show ip mrm status-report** command displays no error reports, the Test Receiver is receiving test packets without loss or duplicates from the Test Sender.

The Cisco implementation of MRM supports Internet Draft of Multicast Routing Monitor (MRM), Internet Engineering Task Force (IETF), March 1999. The IETF originally conceived MRM to use both test packets and real data. The Cisco implementation does not use real data due to technical issues and the fact that the IETF draft did not progress.

Benefits of Multicast Routing Monitor

The benefits of the MRM are as follows:

- MRM allows network personnel to generate test flows without having to use host devices.
- MRM can verify a multicast environment prior to an event. You need not wait for real multicast traffic to fail in order to find out that a problem exists. You can test the multicast routing environment before a planned event.
- MRM provides easy diagnostics. The error information is easy for the user to understand.
- MRM is scalable. This diagnostic tool works well for many users.

How to Use the Multicast Routing Monitor

This section contains the following procedures:

Configuring a Test Receiver

Perform this task to configure a Test Receiver on a router or host.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip mrm test-receiver**
5. **ip mrm accept-manager** ***access-list***

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Specifies an interface, and enters interface configuration mode.
Step 4	**ip mrm test-receiver** **Example:** `Router(config-if)# ip mrm test-receiver`	Configures the interface to operate as a Test Receiver.
Step 5	**ip mrm accept-manager** *access-list* **Example:** `Router(config-if)# ip mrm accept-manager supervisor`	(Optional) Specifies that the Test Receiver can accept status report requests only from Managers specified by the access list. • The access list is required and can be named or numbered. • This example uses an access list named "supervisor." The access list is presumed to be already configured.

Configuring a Test Sender

Perform this task to configure a Test Sender on a different router or host from where you configured the Test Receiver.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip mrm test-sender**
5. **ip mrm accept-manager** [*access-list*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

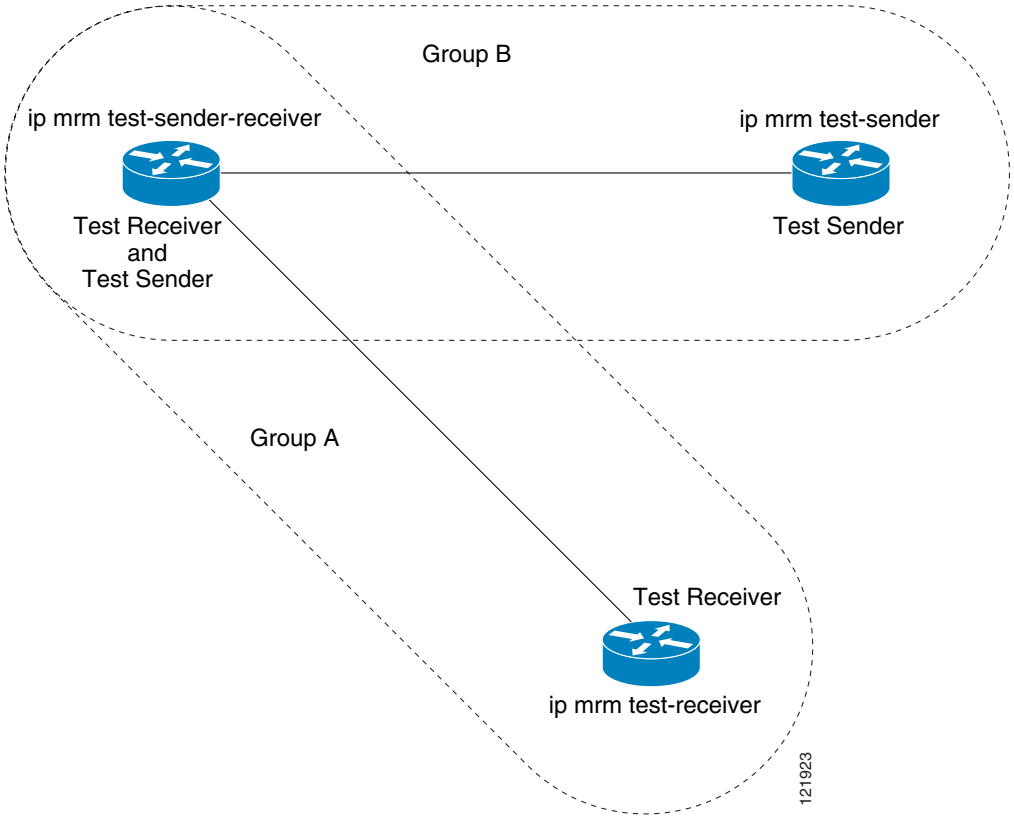
	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Specifies an interface, and enters interface configuration mode.
Step 4	**ip mrm test-sender** **Example:** `Router(config-if)# ip mrm test-sender`	Configures the interface to operate as a Test Sender.
Step 5	**ip mrm accept-manager** [*access-list*] **Example:** `Router(config-if)# ip mrm accept-manager supervisor`	(Optional) Specifies that the Test Sender can accept status report requests only from Managers specified by the access list. • This example uses an access list named "supervisor." The access list is presumed to be already configured.

Monitoring Multiple Groups

If you have more than one multicast group to monitor, you could configure an interface that is a Test Sender for one group and a Test Receiver for another group.

Figure 1 illustrates an environment where the router on the left is the Test Sender for Group A and the Test Receiver for Group B.

Figure 1 Test Sender and Test Receiver for Different Groups on One Router



SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip mrm test-sender-receiver**
5. **ip mrm accept-manager** *access-list* [**test-sender** | **test-receiver**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Specifies an interface, and enters interface configuration mode.
Step 4	**ip mrm test-sender-receiver** **Example:** `Router(config-if)# ip mrm test-sender-receiver`	Configures the interface to operate as a Test Sender for one group and Test Receiver for another group.
Step 5	**ip mrm accept-manager** *access-list* [**test-sender** | **test-receiver**] **Example:** `Router(config-if)# ip mrm accept-manager supervisor test-sender`	(Optional) Specifies that the Test Sender or Test Receiver can accept status report requests only from Managers specified by the access list. • By default, the command applies to both the Test Sender and Test Receiver. Because this device is both, you might need to specify that the restriction applies to only the Test Sender or only the Test Receiver using the **test-sender** keyword or **test-receiver** keyword, respectively.

Configuring a Manager

Perform this task to configure a router as a Manager in order for MRM to function.

Note A host cannot be a Manager.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip mrm manager** *test-name*
4. **manager** *type number* **group** *ip-address*
5. **beacon** [**interval** *seconds*] [**holdtime** *seconds*] [**ttl** *ttl-value*]
6. **udp-port** [**test-packet** *port-number*] [**status-report** *port-number*]
7. **senders** *access-list* [**packet-delay** *milliseconds*] [**rtp** | **udp**] [**target-only** | **all-multicasts** | **all-test-senders**]
8. **receivers** *access-list* **sender-list** *access-list* [*packet-delay*]
9. **receivers** *access-list* [**window** *seconds*] [**report-delay** *seconds*] [**loss** *percentage*] [**no-join**] [**monitor** | **poll**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip mrm manager** *test-name* **Example:** `Router(config)# ip mrm manager test1`	Specifies the name of an MRM test to be created or modified, and enters MRM manager configuration mode. • The test name is used to start, stop, and monitor a test. • From MRM manager configuration mode, you specify the parameters of the test.
Step 4	**manager** ***type number*** **group** ***ip-address*** **Example:** `Router(config-mrm-manager)# manager ethernet 0 group 239.1.1.1`	Specifies which interface on the router is the Manager, and specifies the multicast group address the Test Receiver will listen to.
Step 5	**beacon** [**interval** ***seconds***] [**holdtime** ***seconds***] [**ttl** ***ttl-value***] **Example:** `Router(config-mrm-manager)# beacon interval 60`	(Optional) Changes the frequency, duration, or scope of beacon messages that the Manager sends to the Test Sender and Test Receiver. • By default, beacon messages are sent at an interval of 60 seconds. • By default, the duration of a test period is 86400 seconds (1 day). • By default, the TTL is 32 hops.

	Command or Action	Purpose
Step 6	**udp-port [test-packet** ***port-number*****] [status-report** ***port-number*****]** **Example:** `Router(config-mrm-manager)# udp-port test-packet 20202`	(Optional) Changes the UDP port numbers to which the Test Sender sends test packets or the Test Receiver sends status reports. • Use the optional **test-packet** keyword and *port-number* argument to change the UDP port to which the Test Sender sends test packets. The port number must be even if the packets are Real-Time Transport Protocol (RTP)-encapsulated. The range is from 16384 to 65535. • By default, the Test Sender uses UDP port number 16834 to send test packets. • Use the optional **status-report** keyword and *port-number* argument to change the UDP port to which the Test Receiver sends status reports. The port number must be odd if the packets are RTP Control Protocol (RTCP)-encapsulated. The range is from 16834 to 65535. • By default, the Test Receiver uses UDP port number 65535 to send status reports.
Step 7	**senders** access-list [**packet-delay** *milliseconds*] [**rtp** | **udp**] [**target-only** | **all-multicasts** | **all-test-senders**] **Example:** `Router(config-mrm-manager)# senders 1 packet-delay 30 udp all-test-senders`	Establishes Test Senders for MRM tests. • Use the optional **packet-delay** keyword and *milliseconds* argument to specify the delay between test packets (in milliseconds). The range is from 50 to 10000. The default is 200 milliseconds, which results in 5 packets per second. • Use the optional **rtp** keyword or **udp** keyword to specify the encapsulation of test packets, either Real-Time Transport Protocol (RTP) encapsulated or User Datagram Protocol (UDP) encapsulated. By default, test packets are RTP-encapsulated. • Use the optional **target-only** keyword to specify that test packets are sent out on the targeted interface only (that is, the interface with the IP address that is specified in the Test Sender request target field). By default, test packets are sent out on all interfaces that are enabled with IP multicast. • Use the optional **all-multicasts** keyword to specify that the test packets are sent out on all interfaces that are enabled with IP multicast. This is the default method for sending test packets. • Use the optional **all-test-senders** keyword to specify that test packets are sent out on all interfaces that have test-sender mode enabled. By default, test packets are sent out on all interfaces that are enabled with IP multicast.

	Command or Action	Purpose
Step 8	**receivers** *access-list* **sender-list** *access-list* [*packet-delay*] **Example:** `Router(config-mrm-manager)# receivers 1 sender-list 3`	Establishes Test Receivers for MRM. **Note** Although the Cisco IOS CLI parser accepts the command entered without the **sender-list** *access-list* keyword-argument pair, this keyword-argument pair is not optional. For an MRM test to work, you must specify the sources that the Test Receiver should monitor using the **sender-list** keyword and *access-list* argument. • Use the **sender-list** keyword and *access-list* to specify the sources that the Test Receiver should monitor. If the named or numbered access list matches any access list specified in the **senders** command, the associated **packet-delay** *milliseconds* keyword and argument of that **senders** command are used in the MRM test. Otherwise, the **receivers** command requires that a delay be specified for the *packet-delay* argument. • Use the optional *packet-delay* argument to specify the delay between test packets (in milliseconds). The range is from 50 to 10000. If the **sender-list** access list matches any access list specified in a **senders** command, the associated **packet-delay** *milliseconds* keyword and argument of that **senders** command are used in this command. Otherwise, the **receivers** command requires that a delay be specified for the *packet-delay* argument.

	Command or Action	Purpose
Step 9	**receivers** *access-list* [**window** *seconds*] [**report-delay** *seconds*] [**loss** *percentage*] [**no-join**] [**monitor** | **poll**] **Example:** `Router(config-mrm-manager)# receivers 1 window 7 report-delay 30`	(Optional) Modifies the parameters of Test Receivers. • Use the optional **window** keyword and *seconds* argument to specify the duration (in seconds) of a test period. This is a sliding window of time in which the packet count is collected, so that the loss percentage can be calculated. The range is from 1 to 10. The default is 5 seconds. • Use the optional **report-delay** keyword and *seconds* argument to specify the delay (in seconds) between status reports. The delay prevents multiple Test Receivers from sending status reports to the Manager at the same time for the same failure. This value is relevant only if there are multiple Test Receivers. The range is from 1 to 60. The default is 1 second. • Use the optional **loss** keyword and *percentage* argument to specify the threshold percentage of packet loss required before a status report is triggered. The range is from 0 to 100. The default is 0 percent, which means that a status report is sent for any packet loss. • Use the optional **no-join** keyword to specify that the Test Receiver does not join the monitored group. The default is that the Test Receiver joins the monitored group. • Use either the optional **monitor** or **poll** keyword to specify whether the Test Receiver monitors the test group or polls for receiver statistics. The **monitor** keyword means the Test Receiver reports only if the test criteria are met. The **poll** keyword means the Test Receiver sends status reports regularly, whether test criteria are met or not. The default is the behavior set with the **monitor** keyword.

Conducting an MRM Test and Viewing Results

From the router playing the Manager role you can start and stop the MRM test. To start and subsequently stop your MRM test, perform this task.

When the test begins, the Manager sends a unicast control packet to the Test Sender and Test Receiver, and then the Manager starts sending beacons. The Test Sender and Test Receiver send acknowledgments to the Manager and begin sending or receiving test packets. If an error occurs, the Test Receiver sends an error report to the Manager, which immediately displays the report.

SUMMARY STEPS

1. **enable**
2. **clear ip mrm status-report** [*ip-address*]
3. **show ip mrm interface** [*type number*]
4. **show ip mrm manager** [*test-name*]
5. **mrm** *test-name* **start**
6. **mrm** *test-name* **stop**
7. **show ip mrm status-report** [*ip-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ip mrm status-report** [*ip-address*] **Example:** `Router# clear ip mrm status-report 172.16.0.0`	(Optional) Clears the MRM status report cache.
Step 3	**show ip mrm interface** [*type number*] **Example:** `Router# show ip mrm interface Ethernet 1`	(Optional) Displays MRM information related to interfaces. • Use this command before starting an MRM test to verify the interfaces are participating in MRM, in which roles, and whether the interfaces are up or down.
Step 4	**show ip mrm manager** [*test-name*] **Example:** `Router# show ip mrm manager test1`	(Optional) Displays information about MRM tests. • Use this command before starting an MRM test to verify MRM status information and the parameters configured for an MRM test.
Step 5	**mrm** *test-name* **start** **Example:** `Router# mrm test1 start`	Starts the MRM test.

	Command or Action	Purpose
Step 6	**mrm** *test-name* **stop** **Example:** `Router# mrm test1 stop`	Stops the MRM test.
Step 7	**show ip mrm status-report** [*ip-address*] **Example:** `Router# show ip mrm status-report`	(Optional) Displays the status reports in the MRM status report cache.

Configuration Examples for MRM

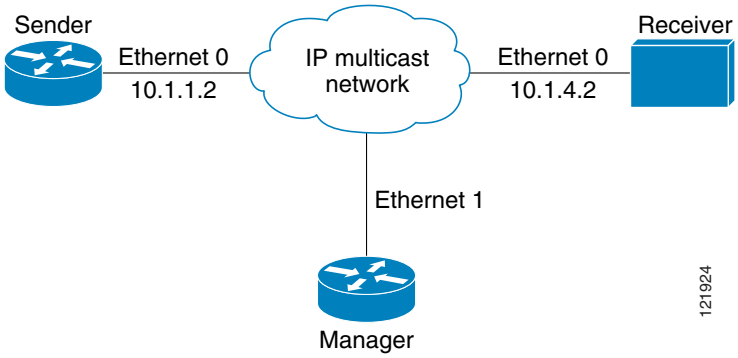
This section provides the following configuration example:

- MRM Configuration: Example, page 13

MRM Configuration: Example

Figure 2 illustrates a Test Sender, a Test Receiver, and a Manager in an MRM environment. The partial configurations for the three devices follow the figure.

Figure 2 Multicast Routing Monitor Example



Test Sender Configuration

```
interface Ethernet 0
 ip mrm test-sender
```

Test Receiver Configuration

```
interface Ethernet 0
 ip mrm test-receiver
```

Manager Configuration

```
ip mrm manager test1
manager Ethernet 1 group 239.1.1.1
senders 1
receivers 2 sender-list 1
```

```
!
access-list 1 permit 10.1.1.2
access-list 2 permit 10.1.4.2
```

Additional References

The following sections provide references related to the using the MRM.

Related Documents

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

Standards

Standard	Title
draft-ietf-mboned-mrm-use-00.txt	*Justification and Use of the Multicast Routing Monitor (MRM) Protocol*

MIBs

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

RFCs

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

Technical Assistance

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

Feature Information for Using the Multicast Routing Monitor

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Releases 12.2(1) or later appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for specific commands was introduced, see the command reference documents.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator (http://www.cisco.com/go/fn). You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click Cancel at the login dialog box and follow the instructions that appear.

Table 1 Feature Information for Using the Multicast Routing Monitor

Feature Name	Releases	Feature Configuration Information
This table is intentionally left blank because no features were introduced or modified in Cisco IOS Release 12.2(1) or later. This table will be updated when feature information is added to this module.	—	—

Configuring PGM Host and Router Assist

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Note Support for the PGM Host feature has been removed. Use of this feature is not recommended.

This module describes the PGM Host and Router Assist feature. PGM Host and Router Assist enables Cisco routers to support multicast applications that operate at the PGM transport layer and the PGM network layer, respectively.

The PGM Reliable Transport Protocol itself is implemented on the hosts of the customer. For information on PGM Reliable Transport Protocol, refer to the Internet Engineering Task Force (IETF) protocol specification draft named *PGM Reliable Transport Protocol Specification*.

For a complete description of the PGM Router Assist commands in this module, see the *Cisco IOS IP Multicast Command Reference*. To locate documentation of other commands that appear in this module, use the command reference master index, or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release.

PGM Overview

Pragmatic General Multicast (PGM) is a reliable multicast transport protocol for multicast applications that require reliable, ordered, duplicate-free multicast data delivery from multiple sources to multiple receivers. PGM guarantees that a receiver in a multicast group either receives all data packets from transmissions and retransmissions, or can detect unrecoverable data packet loss. PGM is intended as a

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

solution for multicast applications with basic reliability requirements. PGM has two main parts: a host element (also referred to as the transport layer of the PGM protocol) and a network element (also referred to as the network layer of the PGM protocol).

The transport layer of the PGM protocol has two main parts: a source part and a receiver part. The transport layer defines how multicast applications send and receive reliable, ordered, duplicate-free multicast data from multiple sources to multiple receivers. PGM Host is the Cisco implementation of the transport layer of the PGM protocol.

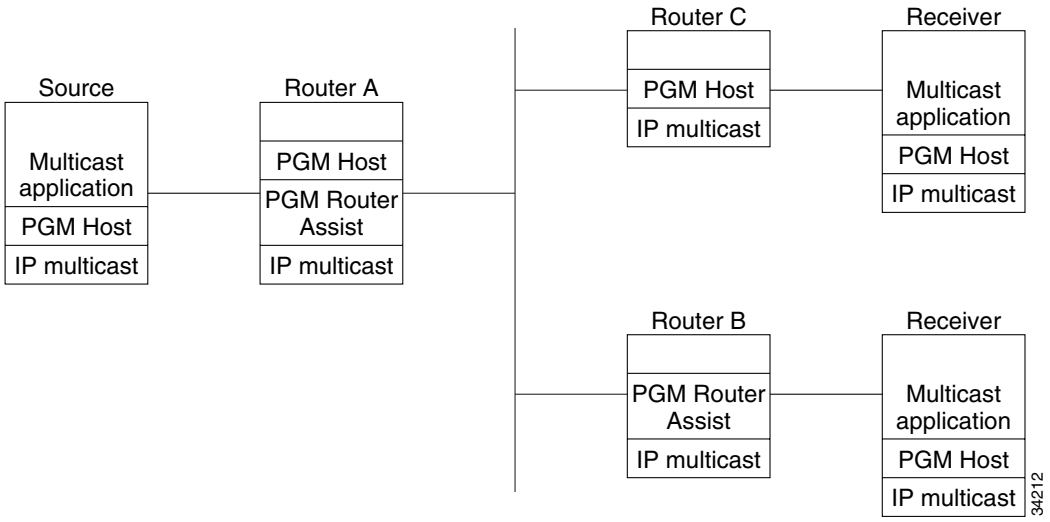
The network layer of the PGM protocol defines how intermediate network devices (such as routers and switches) handle PGM transport data as the data flows through a network. PGM Router Assist is the Cisco implementation of the network layer of the PGM protocol.

Note PGM contains an element that assists routers and switches in handling PGM transport data as it flows through a network. Unlike the Router Assist element, the Host element does not have a current practical application.

PGM is network-layer independent; PGM Host and Router Assist in the Cisco IOS software support PGM over IP. Both PGM Host and Router Assist use a unique transport session identifier (TSI) that identifies each individual PGM session.

Figure 81 shows a simple network topology using the PGM Host and Router Assist feature.

Figure 81 Network Topology Using PGM Host and Router Assist



When the router is functioning as a network element (PGM Router Assist is configured) and PGM Host is configured (Router A in Figure 81), the router can process received PGM packets as a virtual PGM Host, originate PGM packets and serve as its own first hop PGM network element, and forward received PGM packets.

When the router is functioning as a network element and PGM Host is not configured (Router B in Figure 81), the router forwards received PGM packets as specified by PGM Router Assist parameters.

When the router is not functioning as a network element and PGM Host is configured (Router C in Figure 81), the router can receive and forward PGM packets on any router interface simultaneously as specified by PGM Host feature parameters. Although this configuration is supported, it is not recommended in a PGM network because PGM Host works optimally on routers that have PGM Router Assist configured.

PGM Host Configuration Task List

Note Support for the PGM Host feature has been removed. Use of this feature is not recommended.

To configure PGM Host, perform the tasks described in the following sections. The tasks in the first section are required; the tasks in the remaining section are optional.

- Enabling PGM Host (Required)
- Verifying PGM Host Configuration (Optional)

See the end of this module for the section "PGM Host and Router Assist Configuration Examples."

Prerequisites

Before you configure PGM Host, ensure that the following tasks are performed:

- PGM Reliable Transport Protocol is configured on hosts connected to your network.
- PGM Router Assist is configured on intermediate routers and switches connected to your network.
- IP multicast routing is configured on all devices connected to your network that will be processing IP multicast traffic, including the router on which you are configuring PGM Host.
- Protocol Independent Multicast (PIM) or another IP multicast routing protocol is configured on each PGM interface in your network that will send and receive IP multicast packets.
- A PGM multicast virtual host interface (vif) is configured on the router (if you do not plan to source PGM packets through a physical interface installed on the router). The vif enables the router to send and receive IP multicast packets on several different interfaces at once, as dictated by the multicast routing tables on the router.

Enabling PGM Host

Note Support for the PGM Host feature has been removed. Use of this feature is not recommended.

When enabling PGM Host on your router, you must source PGM packets through a vif or out a physical interface installed in the router.

Sourcing PGM packets through a vif enables the router to send and receive PGM packets through any router interface. The vif also serves as the interface to the multicast applications that reside at the PGM network layer.

Sourcing IP multicast traffic out a specific physical or logical interface type (for example, an Ethernet, serial, or loopback interface) configures the router to send PGM packets out that interface only and to receive packets on any router interface.

Enabling PGM Host with a Virtual Host Interface

To enable PGM Host globally on the router and to configure the router to source PGM packets through a vif, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip pgm host`	Enables PGM Host (both the source and receiver parts of the PGM network layer) globally on the router and configures the router to source PGM packets through a vif. **Note** You must configure a vif by using the **interface vif** *number* global configuration command on the router before enabling PGM Host on the router; otherwise, the router will not know to use the vif to source PGM packets and PGM Host will not be enabled on the router.

See the "PGM Host with a Virtual Interface Example" section later in this module for an example of enabling PGM Host with a virtual interface.

Enabling PGM Host with a Physical Interface

To enable PGM Host globally on the router and to configure the router to source PGM packets through a physical interface, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip pgm host`	Enables PGM Host (both the source and receiver part of the PGM network layer) globally on the router.
Step 2	`Router(config)# ip pgm host source-interface` *type number*	Configures the router to source PGM packets through a physical (or logical) interface.

See the "PGM Host with a Physical Interface Example" section later in this module for an example of enabling PGM Host with a physical interface.

Verifying PGM Host Configuration

Note Support for the PGM Host feature has been removed. Use of this feature is not recommended.

To verify that PGM Host is configured correctly on your router, use the following **show** commands in EXEC mode:

- Use the **show ip pgm host sessions** command to display information about current open PGM transport sessions:

```
Router> show ip pgm host sessions

Idx  GSI           Source Port  Type     State   Dest Port  Mcast Address
1    000000000000  0            receiver listen  48059      224.3.3.3
2    9CD72EF099FA  1025         source   conn    48059      224.1.1.1
```

  Specifying a traffic session number or a multicast IP address with the **show ip pgm host sessions** command displays information specific to that PGM transport session:

```
Router> show ip pgm host sessions 2

Idx  GSI           Source Port  Type     State   Dest Port  Mcast Address
2    9CD72EF099FA  1025         source   conn    48059      224.1.1.1

    stream-type (apdu), ttl (255)

    spm-ambient-ivl (6000), txw-adv-secs (6000)
    txw-adv-timeout-max (3600000), txw-rte (16384), txw-secs (30000)
    ncf-max (infinite), spm-rpt-ivl (3000), ihb-min (1000)
    ihb-max (10000), join (0), tpdu-size (16384)
    txw-adv-method (time), tx-buffer-mgmt (return)

    ODATA packets sent                      0
          bytes sent                        0
    RDATA packets sent                      0
          bytes sent                        0
    Total bytes sent                        0
    ADPUs sent                              0
    APDU  transmit memory errors            0
    SPM   packets sent                      6
    NCF   packets sent                      0
    NAK   packets received                  0
          packets received in error         0
    General bad packets                     0
    TX window lead                          0
    TX window trail                         0
```

- Use the **show ip pgm host traffic** command to display traffic statistics at the PGM transport layer:

```
Router> show ip pgm host traffic

General Statistics :

    Sessions in                             0
             out                            0
    Bytes    in                             0
             out                            0

Source Statistics :

    ODATA packets sent                      0
          bytes sent                        0
    RDATA packets sent                      0
          bytes sent                        0
    Total bytes sent                        0
    ADPUs sent                              0
    APDU  transmit memory errors            0
    SPM   packets sent                      0
    NCF   packets sent                      0
    NAK   packets received                  0
          packets received in error         0

Receiver Statistics :

    ODATA packets received                  0
          packets received in error         0
          valid bytes received              0
    RDATA packets received                  0
          packets received in error         0
          valid bytes received              0
    Total valid bytes received              0
    Total bytes received in error           0
    ADPUs received                          0
```

```
SPM   packets received                  0
      packets received in error         0
NCF   packets received                  0
      packets received in error         0
NAK   packets received                  0
      packets received in error         0
      packets sent                      0
Undeliverable packets                   0
General bad packets                     0
Bad checksum packets                    0
```

PGM Router Assist Configuration Task List

To configure PGM Router Assist, perform the required task described in the following section:

- Enabling PGM Router Assist (Required)

Prerequisites

Before you enable PGM Router Assist, ensure that the following tasks are completed:

- PGM Reliable Transport Protocol is configured on hosts connected to your network.
- IP multicast is configured on the router upon which you will enable PGM Router Assist.
- PIM is configured on each PGM interface.

Enabling PGM Router Assist

When enabling PGM Router Assist on your router, you must set up your router to forward PGM packets through a vif or out a physical interface installed in the router.

Setting up your router to forward PGM packets through a vif enables the router to forward PGM packets through any router interface. The vif also serves as the interface to the multicast applications that reside at the PGM network layer.

Setting up your router to forward PGM packets out a specific physical or logical interface type (for example, an Ethernet, serial, or loopback interface) configures the router to forward PGM packets out that interface only.

Enabling PGM Router Assist with a Virtual Host Interface

To enable PGM Router Assist on a vif, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip pgm router`	Enables the router to assist PGM on this interface. **Note** You must configure a vif by using the **interface vif** *number* global configuration command on the router before enabling PGM Assist on the router; otherwise, PGM Assist will not be enabled on the router.

See the "PGM Router Assist with a Virtual Interface Example" section later in this module for an example of enabling PGM Router Assist with a virtual interface.

Enabling PGM Router Assist with a Physical Interface

To enable PGM Router Assist on the router and to configure the router to forward PGM packets through a physical interface, use the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)# ip pgm router`	Enables the router to assist PGM on this interface.

See the "PGM Router Assist with a Physical Interface Example" section later in this module for an example of enabling PGM Router Assist with a physical interface.

Monitoring and Maintaining PGM Host and Router Assist

This section provides information on monitoring and maintaining the PGM Host and Router Assist feature.

Monitoring and Maintaining PGM Host

Note Support for the PGM Host feature has been removed. Use of this feature is not recommended.

To reset PGM Host connections, use the following command in privileged EXEC mode:

Command	Purpose
`Router# clear ip pgm host {defaults | traffic}`	Resets PGM Host connections to their default values and clears traffic statistics.

To enable PGM Host debugging, use the following command in privileged EXEC mode:

Command	Purpose
`Router# debug ip pgm host`	Displays debug messages for PGM Host.

To display PGM Host information, use the following commands in user EXEC mode, as needed:

Command	Purpose
`Router> show ip pgm host defaults`	Displays the default values for PGM Host traffic.
`Router> show ip pgm host sessions [session-number | group-address]`	Displays open PGM Host traffic sessions.
`Router> show ip pgm host traffic`	Displays PGM Host traffic statistics.

Monitoring and Maintaining PGM Router Assist

To clear PGM traffic statistics, use the following command in privileged EXEC mode:

Command	Purpose
`Router# clear ip pgm router [[traffic [type number]] | [rtx-state [group-address]]]`	Clears the PGM traffic statistics. Use the **rtx-state** keyword to clear PGM retransmit state.

To display PGM information, use the following command in privileged EXEC mode:

Command	Purpose
`Router# show ip pgm router [[interface [type number]] | [state [group-address]] | [traffic [type number]]] [verbose]`	Displays information about PGM traffic statistics and TSI state. The TSI is the transport-layer identifier for the source of a PGM session. Confirms that PGM Router Assist is configured, although there might not be any active traffic. Use the **state** or **traffic** keywords to learn whether an interface is actively using PGM.

PGM Host and Router Assist Configuration Examples

Note Support for the PGM Host feature has been removed. Use of this feature is not recommended.

This section provides the following configuration examples:

- PGM Host with a Virtual Interface Example
- PGM Host with a Physical Interface Example
- PGM Router Assist with a Virtual Interface Example
- PGM Router Assist with a Physical Interface Example

Note For clarity, extraneous information has been omitted from the examples in the following sections.

PGM Host with a Virtual Interface Example

Note Support for the PGM Host feature has been removed. Use of this feature is not recommended.

The following example shows PGM Host (both the source and receiver part of the PGM network layer) enabled globally on the router and PGM packets sourced through virtual host interface 1 (vif1). PGM packets can be sent and received on the vif and on the two physical interfaces (ethernet1 and ethernet2) simultaneously.

```
ip multicast-routing
ip routing
ip pgm host
```

```
interface vif1
ip address 10.0.0.1 255.255.255.0
ip pim dense-mode
no ip directed-broadcast
no ip mroute-cache

interface ethernet1
ip address 10.1.0.1 255.255.255.0
ip pim dense-mode
no ip directed-broadcast
no ip mroute-cache
media-type 10BaseT

interface ethernet2
ip address 10.2.0.1 255.255.255.0
ip pim dense-mode
no ip directed-broadcast
no ip mroute-cache
media-type 10BaseT
```

PGM Host with a Physical Interface Example

Note Support for the PGM Host feature has been removed. Use of this feature is not recommended.

The following example shows PGM Host (both the source and receiver part of the PGM network layer) enabled globally on the router and PGM packets sourced out of physical Ethernet interface 1. PGM packets can be received on physical Ethernet interfaces 1 and 2 simultaneously.

```
ip multicast-routing
ip routing
ip pgm host
ip pgm host source-interface ethernet1
ip pgm host source-interface ethernet2

interface ethernet1
ip address 10.1.0.1 255.255.255.0
ip pim dense-mode
no ip directed-broadcast
no ip mroute-cache
media-type 10BaseT

interface ethernet2
ip address 10.2.0.1 255.255.255.0
ip pim dense-mode
no ip directed-broadcast
no ip mroute-cache
media-type 10BaseT
```

PGM Router Assist with a Virtual Interface Example

The following example shows PGM Router Assist (the PGM network layer) enabled on the router and the router set up to forward PGM packets on virtual host interface 1 (vif1). PGM packets can be received on interfaces vif1, ethernet1, and ethernet2 simultaneously.

```
ip multicast-routing
ip routing
```

```
interface vif1
ip address 10.0.0.1 255.255.255.0
ip pim dense-mode
ip pgm router
no ip directed-broadcast
no ip mroute-cache

interface ethernet1
ip address 10.1.0.1 255.255.255.0
ip pim dense-mode
ip pgm router
no ip directed-broadcast
no ip mroute-cache
media-type 10BaseT

interface ethernet2
ip address 10.2.0.1 255.255.255.0
ip pim dense-mode
ip pgm router
no ip directed-broadcast
no ip mroute-cache
```
media-type 10BaseT

PGM Router Assist with a Physical Interface Example

The following example shows PGM Router Assist (the PGM network layer) enabled on the router and the router set up to forward PGM packets out of physical Ethernet interfaces 1 and 2. PGM packets can be received on physical Ethernet interfaces 1 and 2 simultaneously.

```
ip multicast-routing
ip routing

interface ethernet1
ip address 10.1.0.1 255.255.255.0
ip pim dense-mode
ip pgm router
no ip directed-broadcast
no ip mroute-cache
media-type 10BaseT

interface ethernet2
ip address 10.2.0.1 255.255.255.0
ip pim dense-mode
ip pgm router
no ip directed-broadcast
no ip mroute-cache
media-type 10BaseT
```

Configuring Router-Port Group Management Protocol

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

This chapter describes the Router-Port Group Management Protocol (RGMP). RGMP is a Cisco protocol that restricts IP multicast traffic in switched networks. RGMP is a Layer 2 protocol that enables a router to communicate to a switch (or a networking device that is functioning as a Layer 2 switch) the multicast group for which the router would like to receive or forward traffic. RGMP restricts multicast traffic at the ports of RGMP-enabled switches that lead to interfaces of RGMP-enabled routers.

For a complete description of the RGMP commands in this chapter, refer to the*Cisco IOS IP Multicast Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index, or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release.

IP Multicast Routing Overview

The Cisco IOS software supports the following protocols to implement IP multicast routing:

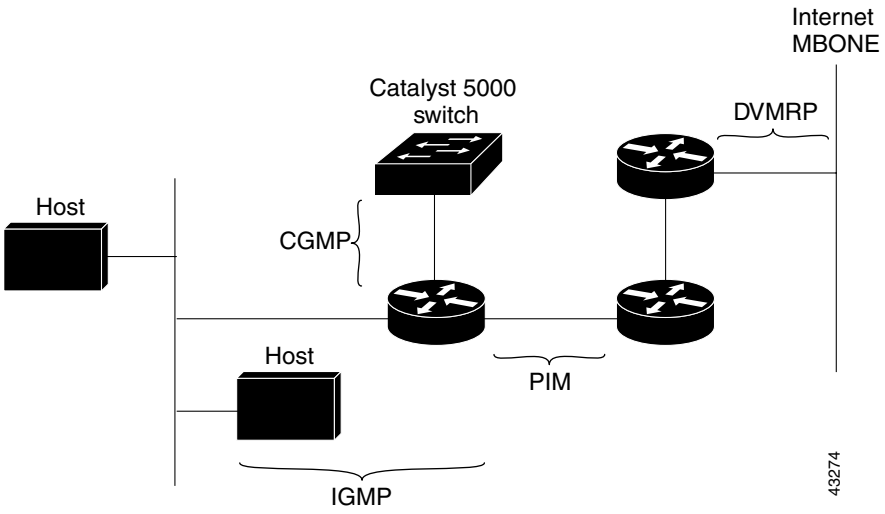
- Internet Group Management Protocol (IGMP) is used between hosts on a LAN and the routers on that LAN to track the multicast groups of which hosts are members.
- Protocol Independent Multicast (PIM) is used between routers so that they can track which multicast packets to forward to each other and to their directly connected LANs.
- Distance Vector Multicast Routing Protocol (DVMRP) is the protocol used on the MBONE (the multicast backbone of the Internet). The Cisco IOS software supports PIM-to-DVMRP interaction.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

- Cisco Group Management Protocol (CGMP) is a protocol used on routers connected to Catalyst switches to perform tasks similar to those performed by IGMP.
- RGMP is a protocol used on routers connected to Catalyst switches or networking devices functioning as Layer 2 switches to restrict IP multicast traffic. Specifically, the protocol enables a router to communicate to a switch the IP multicast group for which the router would like to receive or forward traffic.

Figure 88 shows where these protocols operate within the IP multicast environment.

Figure 88 IP Multicast Routing Protocols



Note CGMP and RGMP cannot interoperate on the same switched network. If RGMP is enabled on a switch or router interface, CGMP is automatically disabled on that switch or router interface; if CGMP is enabled on a switch or router interface, RGMP is automatically disabled on that switch or router interface.

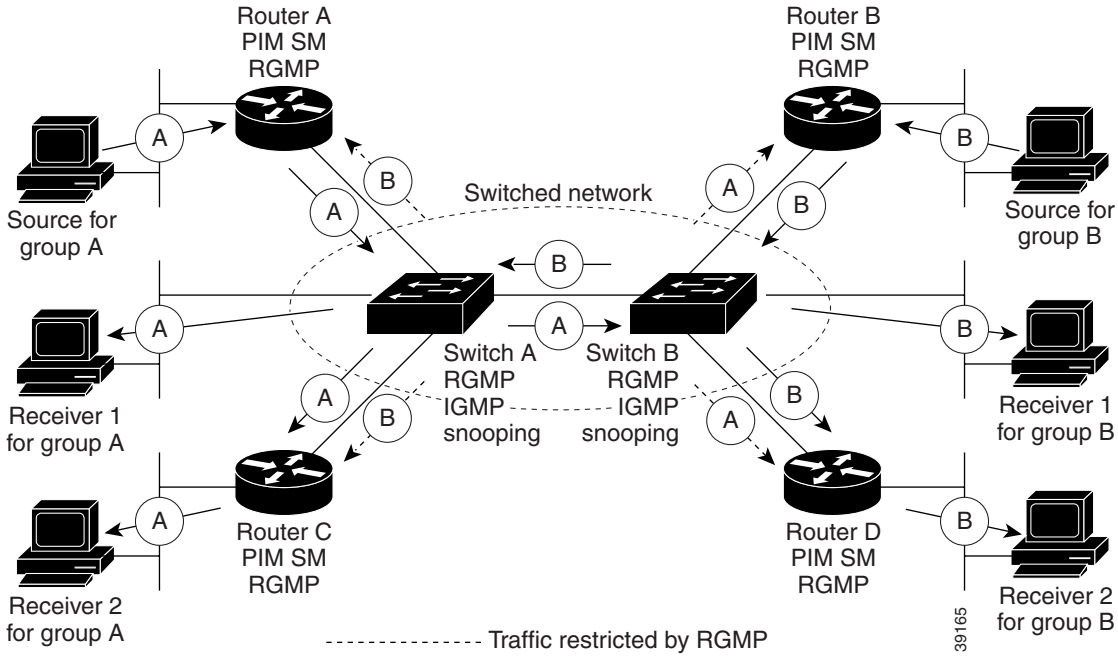
RGMP Overview

RGMP enables a router to communicate to a switch the IP multicast group for which the router would like to receive or forward traffic. RGMP is designed for switched Ethernet backbone networks running PIM sparse mode (PIM-SM) or sparse-dense mode.

Note RGMP-enabled switches and router interfaces in a switched network support directly connected, multicast-enabled hosts that receive multicast traffic. RGMP-enabled switches and router interfaces in a switched network do not support directly connected, multicast-enabled hosts that source multicast traffic. A multicast-enabled host can be a PC, a workstation, or a multicast application running in a router.

Figure 89 shows a switched Ethernet backbone network running PIM in sparse mode, RGMP, and IGMP snooping.

Figure 89 RGMP in a Switched Network

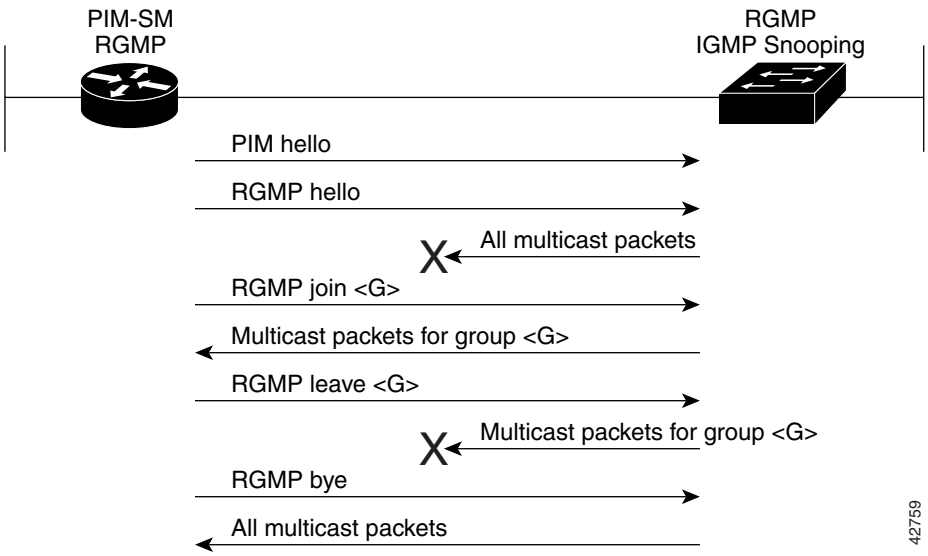


In Figure 89, the sources for the two different multicast groups (the source for group A and the source for group B) send traffic into the same switched network. Without RGMP, traffic from source A is unnecessarily flooded from switch A to switch B, then to router B and router D. Also, traffic from source B is unnecessarily flooded from switch B to switch A, then to router A and router C. With RGMP enabled on all routers and switches in this network, traffic from source A would not flood router B and router D. Also, traffic from source B would not flood router A and router C. Traffic from both sources would still flood the link between switch A and switch B. Flooding over this link would still occur because RGMP does not restrict traffic on links toward other RGMP-enabled switches with routers behind them.

By restricting unwanted multicast traffic in a switched network, RGMP increases the available bandwidth for all other multicast traffic in the network and saves the processing resources of the routers.

Figure 90 shows the RGMP messages sent between an RGMP-enabled router and an RGMP-enabled switch.

Figure 90 RGMP Messages



The router sends simultaneous PIM hello (or a PIM query message if PIM Version 1 is configured) and RGMP hello messages to the switch. The PIM hello message is used to locate neighboring PIM routers. The RGMP hello message instructs the switch to restrict all multicast traffic on the interface from which the switch received the RGMP hello message.

Note RGMP messages are sent to the multicast address 224.0.0.25, which is the local-link multicast address reserved by the Internet Assigned Numbers Authority (IANA) for sending IP multicast traffic from routers to switches.

If RGMP is not enabled on both the router and the switch, the switch automatically forwards all multicast traffic out the interface from which the switch received the PIM hello message.

The router sends the switch an RGMP join <G> message (where G is the multicast group address) when the router wants to receive traffic for a specific multicast group. The RGMP join message instructs the switch to forward multicast traffic for group <G> out the interface from which the switch received the RGMP hello message.

Note The router sends the switch an RGMP join <G> message for a multicast group even if the router is only forwarding traffic for the multicast group into a switched network. By joining a specific multicast group, the router can determine if another router is also forwarding traffic for the multicast group into the same switched network. If two routers are forwarding traffic for a specific multicast group into the same switched network, the two routers use the PIM assert mechanism to determine which router should continue forwarding the multicast traffic into the network.

The router sends the switch an RGMP leave <G> message when the router wants to stop receiving traffic for a specific multicast group. The RGMP leave message instructs the switch to stop forwarding the multicast traffic on the port from which the switch received the PIM and RGMP hello messages.

Note An RGMP-enabled router cannot send an RGMP leave <G> message until the router does not receive or forward traffic from any source for a specific multicast group (if multiple sources exist for a specific multicast group).

The router sends the switch an RGMP bye message when RGMP is disabled on the router. The RGMP bye message instructs the switch to forward the router all IP multicast traffic on the port from which the switch received the PIM and RGMP hello messages, as long as the switch continues to receive PIM hello messages on the port.

RGMP Configuration Task List

To configure RGMP, perform the tasks described in the following sections. The tasks in the first two section are required; the tasks in the remaining section are optional.

- Enabling RGMP (Required)
- Verifying RGMP Configuration (Optional)

See the end of this chapter for the section "RGMP Configuration Example."

Prerequisites

Before you enable RGMP, ensure that the following features are enabled on your router:

- IP routing
- IP multicast
- PIM in sparse mode, sparse-dense mode, source specific mode, or bidirectional mode

If your router is in a bidirectional group, make sure to enable RGMP only on interfaces that do not function as a designated forwarder (DF). If you enable RGMP on an interface that functions as a DF, the interface will not forward multicast packets up the bidirectional shared tree to the rendezvous point (RP).

You must have the following features enabled on your switch:

- IP multicast
- IGMP snooping

Note Refer to the Catalyst switch software documentation for RGMP switch configuration tasks and command information.

Enabling RGMP

To enable RGMP, use the following commands on all routers in your network beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type number`	Specifies the router interface on which you want to configure RGMP and enters interface configuration mode.
Step 2	`Router(config-if)# ip rgmp`	Enables RGMP on a specified interface.

See the "RGMP Configuration Example" section later in this chapter for an example of how to configure RGMP.

Verifying RGMP Configuration

To verify that RGMP is enabled on the correct interfaces, use the **show ip igmp interface** EXEC command:

```
Router> show ip igmp interface

Ethernet1/0 is up, line protocol is up
  Internet address is 10.0.0.0/24
  IGMP is enabled on interface
  Current IGMP version is 2
→ RGMP is enabled
  IGMP query interval is 60 seconds
  IGMP querier timeout is 120 seconds
  IGMP max query response time is 10 seconds
  Last member query response interval is 1000 ms
  Inbound IGMP access group is not set
  IGMP activity: 1 joins, 0 leaves
  Multicast routing is enabled on interface
  Multicast TTL threshold is 0
  Multicast designated router (DR) is 10.0.0.0 (this system)
  IGMP querying router is 10.0.0.0 (this system)
  Multicast groups joined (number of users):
      224.0.1.40(1)
```

Note If RGMP is not enabled on an interface, no RGMP information is displayed in the **show ip igmp interface** command output for that interface.

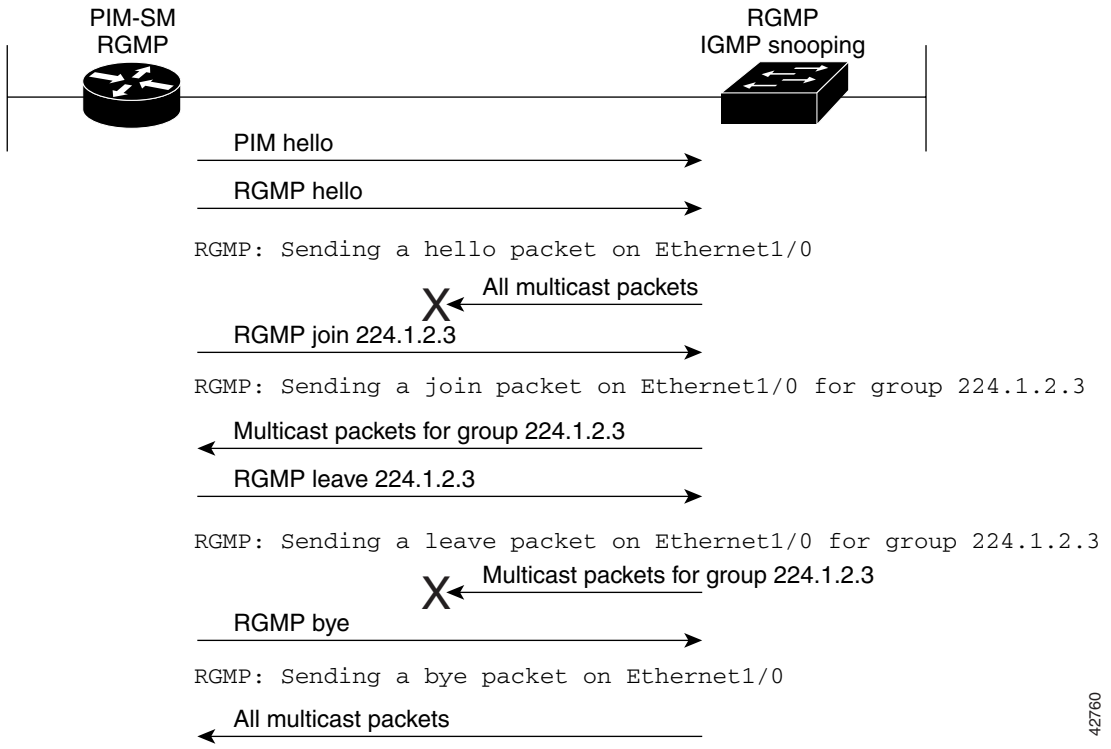
Monitoring and Maintaining RGMP

To enable RGMP debugging, use the following command in privileged EXEC mode:

Command	Purpose
`Router# debug ip rgmp [group-name | group-address]`	Logs debug messages sent by an RGMP-enabled router. Using the command without arguments logs RGMP Join <G> and RGMP leave <G> messages for all multicast groups configured on the router. Using the command with arguments logs RGMP join <G> and RGMP leave <G> messages for the specified group.

Figure 91 shows the debug messages that are logged by an RGMP-enabled router as the router sends RGMP join <G> and RGMP leave <G> messages to an RGMP-enabled switch.

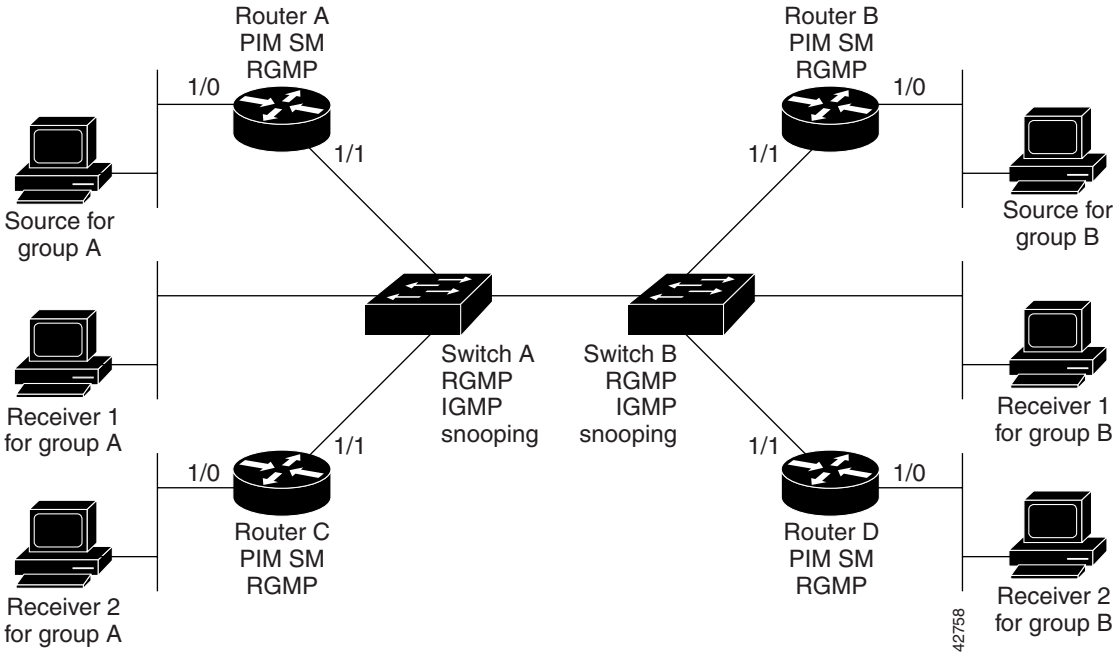
Figure 91 RGMP Debug Messages



RGMP Configuration Example

This section provides an RGMP configuration example that shows the individual configurations for the routers and switches shown in Figure 92.

Figure 92 RGMP Configuration Example



Router A Configuration

```
ip routing
ip multicast-routing

interface ethernet 1/0
  ip address 10.0.0.1 255.0.0.0
  ip pim sparse-dense-mode
  no shutdown

interface ethernet 1/1
  ip address 10.1.0.1 255.0.0.0
  ip pim sparse-dense-mode
  ip rgmp
  no shutdown
```

Router B Configuration

```
ip routing
ip multicast-routing

interface ethernet 1/0
  ip address 10.2.0.1 255.0.0.0
  ip pim sparse-dense-mode
  no shutdown

interface ethernet 1/1
  ip address 10.3.0.1 255.0.0.0
  ip pim sparse-dense-mode
  ip rgmp
  no shutdown
```

Router C Configuration

```
ip routing
ip multicast-routing
```

```
interface ethernet 1/0
  ip address 10.4.0.1 255.0.0.0
  ip pim sparse-dense-mode
  no shutdown

interface ethernet 1/1
  ip address 10.5.0.1 255.0.0.0
  ip pim sparse-dense-mode
  ip rgmp
  no shutdown
```

Router D Configuration

```
ip routing
ip multicast-routing

interface ethernet 1/0
  ip address 10.6.0.1 255.0.0.0
  ip pim sparse-dense-mode
  no shutdown

interface ethernet 1/1
  ip address 10.7.0.1 255.0.0.0
  ip pim sparse-dense-mode
  ip rgmp
  no shutdown
```

Switch A Configuration

```
Switch> (enable) set igmp enable
Switch> (enable) set rgmp enable
```

Switch B Configuration

```
Switch> (enable) set igmp enable
Switch> (enable) set rgmp enable
```

Configuring DVMRP Interoperability

This modules describes the Distance Vector Multicast Routing Protocol (DVMRP) Interoperability feature. Cisco routers run Protocol Independent Multicast (PIM), and know enough about DVMRP to successfully forward multicast packets to and receive packets from a DVMRP neighbor. It is also possible to propagate DVMRP routes into and through a PIM cloud. The Cisco IOS software propagates DVMRP routes and builds a separate database for these routes on each router, but PIM uses this routing information to make the packet-forwarding decision. Cisco IOS software does not implement the complete DVMRP.

DVMRP builds a parent-child database using a constrained multicast model to build a forwarding tree rooted at the source of the multicast packets. Multicast packets are initially flooded down this source tree. If redundant paths are on the source tree, packets are not forwarded along those paths. Forwarding occurs until prune messages are received on those parent-child links, which further constrains the broadcast of multicast packets.

DVMRP is implemented in the equipment of many vendors and is based on the public-domain mrouted program. The Cisco IOS software supports dynamic discovery of DVMRP routers and can interoperate with them over traditional media such as Ethernet and FDDI, or over DVMRP-specific tunnels.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release.

Basic DVMRP Interoperability Configuration Task List

To configure basic interoperability with DVMRP machines, perform the tasks described in the following sections. The tasks in the first section are required; the tasks in the remaining sections are optional.

- Configuring DVMRP Interoperability (Required)
- Configuring a DVMRP Tunnel (Optional)
- Advertising Network 0.0.0.0 to DVMRP Neighbors (Optional)

For more advanced DVMRP interoperability features, see the section "Advanced DVMRP Interoperability Configuration Task List" later in this chapter.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Configuring DVMRP Interoperability

Cisco multicast routers using PIM can interoperate with non-Cisco multicast routers that use the DVMRP.

PIM routers dynamically discover DVMRP multicast routers on attached networks. Once a DVMRP neighbor has been discovered, the router periodically sends DVMRP report messages advertising the unicast sources reachable in the PIM domain. By default, directly connected subnets and networks are advertised. The router forwards multicast packets that have been forwarded by DVMRP routers and, in turn, forwards multicast packets to DVMRP routers.

You can configure which sources are advertised and which metrics are used by configuring the **ip dvmrp metric** interface configuration command. You can also direct all sources learned via a particular unicast routing process to be advertised into DVMRP.

The mrouted protocol is a public-domain implementation of DVMRP. It is necessary to use mrouted Version 3.8 (which implements a nonpruning version of DVMRP) when Cisco routers are directly connected to DVMRP routers or interoperate with DVMRP routers over an multicast backbone (MBONE) tunnel. DVMRP advertisements produced by the Cisco IOS software can cause older versions of mrouted to corrupt their routing tables and those of their neighbors. Any router connected to the MBONE should have an access list to limit the number of unicast routes that are advertised via DVMRP.

To configure the sources that are advertised and the metrics that are used when DVMRP report messages are sent, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip dvmrp metric` *metric* [**list** *access-list*] [*protocol process-id*]	Configures the metric associated with a set of destinations for DVMRP reports.

A more sophisticated way to achieve the same results as the preceding command is to use a route map instead of an access list. Thus, you have a finer granularity of control. To subject unicast routes to route map conditions before they are injected into DVMRP, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip dvmrp metric` *metric* [**route-map** *map-name*]	Subjects unicast routes to route map conditions before they are injected into DVMRP.

Responding to mrinfo Requests

The Cisco IOS software answers mrinfo requests sent by mrouted systems and Cisco routers. The software returns information about neighbors on DVMRP tunnels and all of the interfaces of the router. This information includes the metric (which is always set to 1), the configured TTL threshold, the status of the interface, and various flags. The **mrinfo** EXEC command can also be used to query the router itself, as in the following example:

```
mm1-7kd# mrinfo

  171.69.214.27 (mm1-7kd.cisco.com) [version cisco 11.1] [flags: PMS]:
  171.69.214.27 -> 171.69.214.26 (mm1-r7kb.cisco.com) [1/0/pim/querier]
  171.69.214.27 -> 171.69.214.25 (mm1-45a.cisco.com) [1/0/pim/querier]
  171.69.214.33 -> 171.69.214.34 (mm1-45c.cisco.com) [1/0/pim]
  171.69.214.137 -> 0.0.0.0 [1/0/pim/querier/down/leaf]
```

```
171.69.214.203 -> 0.0.0.0 [1/0/pim/querier/down/leaf]
171.69.214.18 -> 171.69.214.20 (mm1-45e.cisco.com) [1/0/pim]
171.69.214.18 -> 171.69.214.19 (mm1-45c.cisco.com) [1/0/pim]
171.69.214.18 -> 171.69.214.17 (mm1-45a.cisco.com) [1/0/pim]
```

See the "DVMRP Interoperability Example" section later in this chapter for an example of how to configure a PIM router to interoperate with a DVMRP router.

Configuring a DVMRP Tunnel

The Cisco IOS software supports DVMRP tunnels to the MBONE. You can configure a DVMRP tunnel on a router if the other end is running DVMRP. The software then sends and receives multicast packets over the tunnel. This strategy allows a PIM domain to connect to the DVMRP router in the case where all routers on the path do not support multicast routing. You cannot configure a DVMRP tunnel between two routers.

When a Cisco router runs DVMRP over a tunnel, it advertises sources in DVMRP report messages much as it does on real networks. In addition, the software caches DVMRP report messages it receives and uses them in its Reverse Path Forwarding (RPF) calculation. This behavior allows the software to forward multicast packets received over the tunnel.

When you configure a DVMRP tunnel, you should assign a tunnel an address in the following two cases:

- To enable the sending of IP packets over the tunnel
- To indicate whether the Cisco IOS software should perform DVMRP summarization

You can assign an IP address either by using the **ip address** interface configuration command, or by using the **ip unnumbered** interface configuration command to configure the tunnel to be unnumbered. Either of these two methods allows IP multicast packets to flow over the tunnel. The software will not advertise subnets over the tunnel if the tunnel has a different network number from the subnet. In this case, the software advertises only the network number over the tunnel.

To configure a DVMRP tunnel, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# interface tunnel number`	Specifies a tunnel interface in global configuration mode and puts the router into interface configuration mode.
Step 2	`Router(config-if)# tunnel source ip-address`	Sets the source address of the tunnel interface. This address is the IP address of the interface on the router.
Step 3	`Router(config-if)# tunnel destination ip-address`	Sets the destination adddress of the tunnel interface. This address is the IP address of the mrouted multitask router.
Step 4	`Router(config-if)# tunnel mode dvmrp`	Configures a DVMRP tunnel.
Step 5	`Router(config-if)# ip address address mask` or `Router(config-if)# ip unnumbered type number`	Assigns an IP address to the interface. or Configures the interface as unnumbered.
Step 6	`Router(config-if)# ip pim [dense-mode | sparse-mode]`	Configures PIM on the interface.
Step 7	`Router(config-if)# ip dvmrp accept-filter access-list [distance | ip neighbor-list access-list]`	Configures an acceptance filter for incoming DVMRP reports.

See the "DVMRP Tunnel Example" section later in this chapter for an example of how to configure a DVMRP tunnel.

Advertising Network 0.0.0.0 to DVMRP Neighbors

The mrouted protocol is a public domain implementation of DVMRP. If your router is a neighbor to an mrouted Version 3.6 device, you can configure the Cisco IOS software to advertise network 0.0.0.0 to the DVMRP neighbor. Do not advertise the DVMRP default into the MBONE. You must specify whether only route 0.0.0.0 is advertised or if other routes can also be specified.

To advertise network 0.0.0.0 to DVMRP neighbors on an interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip dvmrp default-information {originate | only}`	Advertises network 0.0.0.0 to DVMRP neighbors.

Advanced DVMRP Interoperability Configuration Task List

Cisco routers run PIM and know enough about DVMRP to successfully forward multicast packets to receivers and receive multicast packets from senders. It is also possible to propagate DVMRP routes into and through a PIM cloud. PIM uses this information; however, Cisco routers do not implement DVMRP to forward multicast packets.

The basic DVMRP interoperability features are described in the section "Basic DVMRP Interoperability Configuration Task List" earlier in this chapter. To configure more advanced DVMRP interoperability features on a Cisco router, perform the optional tasks described in the following sections:

- Enabling DVMRP Unicast Routing (Optional)
- Limiting the Number of DVMRP Routes Advertised (Optional)
- Changing the DVMRP Route Threshold (Optional)
- Configuring a DVMRP Summary Address (Optional)
- Disabling DVMRP Automatic Summarization (Optional)
- Adding a Metric Offset to the DVMRP Route (Optional)
- Rejecting a DVMRP Nonpruning Neighbor (Optional)
- Configuring a Delay Between DVRMP Reports (Optional)

Enabling DVMRP Unicast Routing

Because policy for multicast routing and unicast routing requires separate topologies, PIM must follow the multicast topology to build loopless distribution trees. Using DVMRP unicast routing, Cisco routers and mrouted machines exchange DVMRP unicast routes, to which PIM can then reverse path forward.

Cisco routers do not perform DVMRP multicast routing among each other, but they can exchange DVMRP routes. The DVMRP routes provide a multicast topology that may differ from the unicast topology. These routes allow PIM to run over the multicast topology, thereby allowing PIM sparse mode over the MBONE topology.

When DVMRP unicast routing is enabled, the router caches routes learned in DVMRP report messages in a DVMRP routing table. PIM prefers DVMRP routes to unicast routes by default, but that preference can be configured.

DVMRP unicast routing can run on all interfaces, including generic routing encapsulation (GRE) tunnels. On DVMRP tunnels, it runs by virtue of DVMRP multicast routing. This feature does not enable DVMRP multicast routing among Cisco routers. However, if there is a DVMRP-capable multicast router, the Cisco router will do PIM/DVMRP multicast routing interaction.

To enable DVMRP unicast routing, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip dvmrp unicast-routing`	Enables DVMRP unicast routing.

Limiting the Number of DVMRP Routes Advertised

By default, only 7000 DVMRP routes will be advertised over an interface enabled to run DVMRP (that is, a DVMRP tunnel, an interface where a DVMRP neighbor has been discovered, or an interface configured to run the **ip dvmrp unicast-routing** interface configuration command).

To change this limit, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip dvmrp route-limit` *count*	Changes the number of DVMRP routes advertised over an interface enabled to run DVMRP.

Changing the DVMRP Route Threshold

By default, 10,000 DVMRP routes may be received per interface within a 1-minute interval. When that rate is exceeded, a syslog message is issued, warning that a route surge might be occurring. The warning is typically used to quickly detect when routers have been misconfigured to inject a large number of routes into the MBONE.

To change the threshold number of routes that trigger the warning, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip dvmrp routehog-notification` *route-count*	Configures the number of routes that trigger a syslog message.

Use the **show ip igmp interface** EXEC command to display a running count of routes. When the count is exceeded, "*** ALERT ***" is appended to the line.

Configuring a DVMRP Summary Address

You can customize the summarization of DVMRP routes if the default classful automatic summarization does not suit your needs. To summarize such routes, specify a summary address by using the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip dvmrp summary-address summary-address mask [metric value]`	Specifies a DVMRP summary address.

Note At least one, more-specific route must be present in the unicast routing table before a configured summary address will be advertised.

Disabling DVMRP Automatic Summarization

By default, the Cisco IOS software performs some level of DVMRP summarization automatically. Disable this function if you want to advertise all routes, not just a summary. If you configure the **ip dvmrp summary-address** interface configuration command and did not configure the **no ip dvmrp auto-summary** command, you get both custom and automatic summaries.

To disable DVMRP automatic summarization, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# no ip dvmrp auto-summary`	Disables DVMRP automatic summarization.

Adding a Metric Offset to the DVMRP Route

By default, the router increments by 1 the metric of a DVMRP route advertised in incoming DVMRP reports. You can change the metric if you want to favor or not favor a certain route. The DVMRP metric is a hop count. Therefore, a very slow serial line of one hop is preferred over a route that is two hops over FDDI or another fast medium.

For example, perhaps a route is learned by Router A and the same route is learned by Router B with a higher metric. If you want to use the path through Router B because it is a faster path, you can apply a metric offset to the route learned by Router A to make it larger than the metric learned by Router B, allowing you to choose the path through Router B.

To change the default metric, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip dvmrp metric-offset [in | out] increment`	Changes the metric added to DVMRP routes advertised in incoming reports.

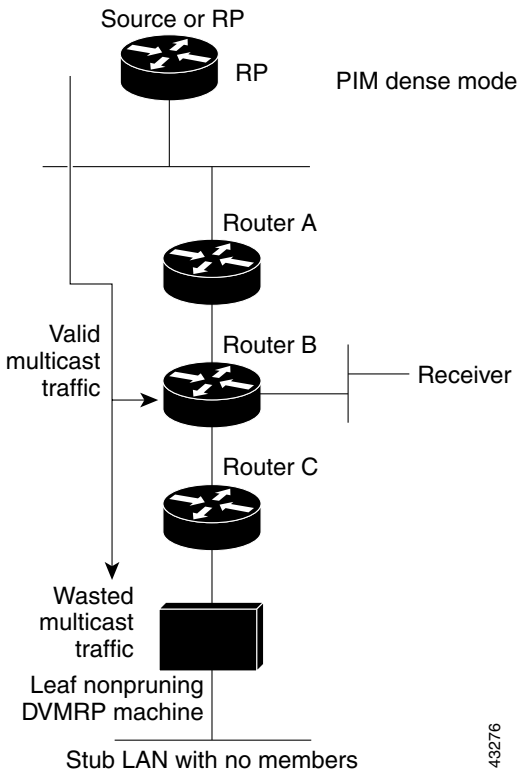
Similar to the **metric** keyword in mrouted configuration files, the following is true when using the **ip dvmrp metric-offset** interface configuration command:

- When you specify the **in** keywordor no keyword, the *increment* value is added to incoming DVMRP reports and is reported in mrinfo replies. The default value for the **in** keyword is 1.
- When you specify the **out** keyword, the *increment* is added to outgoing DVMRP reports for routes from the DVMRP routing table. The default value for the **out** keyword is 0.

Rejecting a DVMRP Nonpruning Neighbor

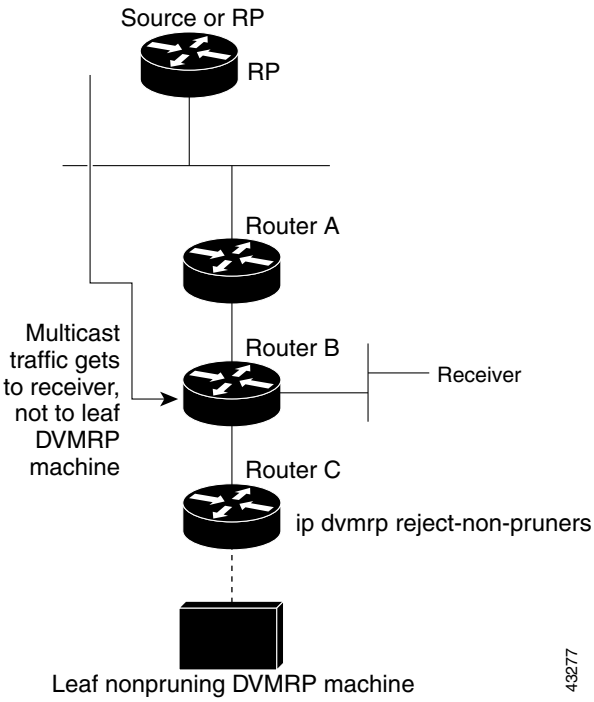
By default, Cisco routers accept all DVMRP neighbors as peers, regardless of their DVMRP capability or lack of. However, some non-Cisco machines run old versions of DVMRP that cannot prune, so they will continuously receive forwarded packets unnecessarily, wasting bandwidth. Figure 93 shows this scenario.

Figure 93 Leaf Nonpruning DVMRP Neighbor



You can prevent a router from peering (communicating) with a DVMRP neighbor if that neighbor does not support DVMRP pruning or grafting. To do so, configure Router C (which is a neighbor to the leaf, nonpruning DVMRP machine) with the **ip dvmrp reject-non-pruners** interface configuration command on the interface to the nonpruning machine. Figure 94 illustrates this scenario. In this case, when the router receives a DVMRP probe or report message without the Prune-Capable flag set, the router logs a syslog message and discards the message.

Figure 94 Router Rejects Nonpruning DVMRP Neighbor



Note that the **ip dvmrp reject-non-pruners** command prevents peering with neighbors only. If there are any nonpruning routers multiple hops away (downstream toward potential receivers) that are not rejected, then a nonpruning DVMRP network might still exist.

To prevent peering with nonpruning DVMRP neighbors, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip dvmrp reject-non-pruners`	Prevents peering with nonpruning DVMRP neighbors.

Configuring a Delay Between DVRMP Reports

You can configure an interpacket delay of a DVMRP report. The delay is the number of milliseconds that elapse between transmissions of sets of packets that constitute a report. The number of packets in the set is determined by the *burst* value, which defaults to 2 packets. The *milliseconds* value defaults to 100 milliseconds.

To change the default values of the delay, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip dvmrp output-report-delay` *milliseconds* [*burst*]	Configures an interpacket delay between DVMRP reports.

Monitoring and Maintaining DVMRP

To clear routes from the DVMRP routing table, use the following command in EXEC mode:

Command	Purpose
`Router# clear ip dvmrp route { * | route}`	Deletes routes from the DVMRP routing table.

To display entries in the DVMRP routing table, use the following command in EXEC mode:

Command	Purpose
`Router# show ip dvmrp route [name | ip-address | type number]`	Displays the entries in the DVMRP routing table.

DVMRP Configuration Examples

This section provides the following DVMRP configuration examples:

- DVMRP Interoperability Example
- DVMRP Tunnel Example

DVMRP Interoperability Example

The following example configures DVMRP interoperability for configurations when the PIM router and the DVMRP router are on the same network segment. In this example, access list 1 advertises the networks (198.92.35.0, 198.92.36.0, 198.92.37.0, 131.108.0.0, and 150.136.0.0) to the DVMRP router, and access list 2 is used to prevent all other networks from being advertised (the **ip dvmrp metric 0** interface configuration command).

```
interface ethernet 0
 ip address 131.119.244.244 255.255.255.0
 ip pim dense-mode
 ip dvmrp metric 1 list 1
 ip dvmrp metric 0 list 2

access-list 1 permit 198.92.35.0 0.0.0.255
access-list 1 permit 198.92.36.0 0.0.0.255
access-list 1 permit 198.92.37.0 0.0.0.255
access-list 1 permit 131.108.0.0 0.0.255.255
access-list 1 permit 150.136.0.0 0.0.255.255
access-list 1 deny   0.0.0.0 255.255.255.255
access-list 2 permit 0.0.0.0 255.255.255.255
```

DVMRP Tunnel Example

The following example configures a DVMRP tunnel:

```
!
ip multicast-routing
!
interface tunnel 0
```

```
 ip unnumbered ethernet 0
 ip pim dense-mode
 tunnel source ethernet 0
 tunnel destination 192.70.92.133
 tunnel mode dvmrp
!
interface ethernet 0
 description Universitat DMZ-ethernet
 ip address 192.76.243.2 255.255.255.0
 ip pim dense-mode
```