



IP Multicast: Multicast Optimization Configuration Guide, Cisco IOS Release 15S

Americas Headquarters

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

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

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

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

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

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

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

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

© 2012 Cisco Systems, Inc. All rights reserved.



CONTENTS

Optimizing PIM Sparse Mode in a Large IP Multicast Deployment 1

Finding Feature Information 1

Prerequisites for Optimizing PIM Sparse Mode in a Large IP Multicast Deployment 1

Information About Optimizing PIM Sparse Mode in a Large IP Multicast Deployment 2

PIM Registering Process 2

PIM Version 1 Compatibility 2

PIM Designated Router 3

PIM Sparse-Mode Register Messages 3

Preventing Use of Shortest-Path Tree to Reduce Memory Requirement 3

PIM Shared Tree and Source Tree - Shortest-Path Tree 3

Benefit of Preventing or Delaying the Use of the Shortest-Path Tree 4

How to Optimize PIM Sparse Mode in a Large IP Multicast Deployment 5

Optimizing PIM Sparse Mode in a Large Deployment 5

Configuration Examples for Optimizing PIM Sparse Mode in a Large IP Multicast Deployment 7

Optimizing PIM Sparse Mode in a Large IP Multicast Deployment Example 7

Additional References 7

Feature Information for Optimizing PIM Sparse Mode in a Large IP Multicast Deployment 8

Multicast Subsecond Convergence 11

Finding Feature Information 11

Prerequisites for Multicast Subsecond Convergence 11

Restrictions for Multicast Subsecond Convergence 11

Information About Multicast Subsecond Convergence 12

Benefits of Multicast Subsecond Convergence 12

Multicast Subsecond Convergence Scalability Enhancements 12

PIM Router Query Messages 12

Reverse Path Forwarding 12

RPF Checks 13

Triggered RPF Checks 13

RPF Failover 13

Topology Changes and Multicast Routing Recovery	13
How to Configure Multicast Subsecond Convergence	14
Modifying the Periodic RPF Check Interval	14
What to Do Next	15
Configuring PIM RPF Failover Intervals	15
What to Do Next	16
Modifying the PIM Router Query Message Interval	16
What to Do Next	17
Verifying Multicast Subsecond Convergence Configurations	17
Configuration Examples for Multicast Subsecond Convergence	18
Example Modifying the Periodic RPF Check Interval	18
Example Configuring PIM RPF Failover Intervals	18
Modifying the PIM Router Query Message Interval Example	19
Additional References	19
Feature Information for Multicast Subsecond Convergence	20
IP Multicast Load Splitting across Equal-Cost Paths	21
Finding Feature Information	21
Prerequisites for IP Multicast Load Splitting across Equal-Cost Paths	21
Information About IP Multicast Load Splitting across Equal-Cost Paths	21
Load Splitting Versus Load Balancing	22
Default Behavior for IP Multicast When Multiple Equal-Cost Paths Exist	22
Methods to Load Split IP Multicast Traffic	24
Overview of ECMP Multicast Load Splitting	24
ECMP Multicast Load Splitting Based on Source Address Using the S-Hash Algorithm	25
ECMP Multicast Load Splitting Based on Source and Group Address Using the Basic S-G-Hash Algorithm	25
Predictability As a By-Product of Using the S-Hash and Basic S-G-Hash Algorithms	25
Polarization As a By-Product of Using the S-Hash and Basic S-G-Hash Algorithms	25
ECMP Multicast Load Splitting Based on Source Group and Next-Hop Address	26
Effect of ECMP Multicast Load Splitting on PIM Neighbor Query and Hello Messages for RPF Path Selection	27
Effect of ECMP Multicast Loading Splitting on Assert Processing in PIM-DM and DF Election in Bidir-PIM	28
Effect of ECMP Multicast Load Splitting on the PIM Assert Process in PIM-SM and PIM-SSM	29
ECMP Multicast Load Splitting and Reconvergence When Unicast Routing Changes	30

Use of BGP with ECMP Multicast Load Splitting	30
Use of ECMP Multicast Load Splitting with Static Mroutes	30
Alternative Methods of Load Splitting IP Multicast Traffic	31
How to Load Split IP Multicast Traffic over ECMP	31
Enabling ECMP Multicast Load Splitting	31
Prerequisites for IP Multicast Load Splitting - ECMP	32
Restrictions	32
Enabling ECMP Multicast Load Splitting Based on Source Address	32
Enabling ECMP Multicast Load Splitting Based on Source and Group Address	34
Enabling ECMP Multicast Load Splitting Based on Source Group and Next-Hop Address	36
Configuration Examples for Load Splitting IP Multicast Traffic over ECMP	38
Example Enabling ECMP Multicast Load Splitting Based on Source Address	38
Example Enabling ECMP Multicast Load Splitting Based on Source and Group Address	38
Example Enabling ECMP Multicast Load Splitting Based on Source Group and Next-Hop Address	38
Additional References	39
Feature Information for Load Splitting IP Multicast Traffic over ECMP	39
Configuring Multicast Admission Control	43
Finding Feature Information	43
Prerequisites for Configuring Multicast Admission Control	43
Information About Configuring Multicast Admission Control	43
Multicast Admission Control	44
Multicast Admission Control Features	44
Global and Per MVRF Mroute State Limit	45
Global and Per MVRF Mroute State Limit Feature Design	45
Mechanics of Global and Per MVRF Mroute State Limiters	45
MSDP SA Limit	46
MSDP SA Limit Feature Design	46
Mechanics of MSDP SA Limiters	46
Tips for Configuring MSDP SA Limiters	47
IGMP State Limit	47
IGMP State Limit Feature Design	47
Mechanics of IGMP State Limiters	47
Per Interface Mroute State Limit	48
Per Interface Mroute State Limit Feature Design	49

Mechanics of Per Interface Mroute State Limiters	50
Tips for Configuring Per Interface Mroute State Limiters	51
Bandwidth-Based CAC for IP Multicast	51
Bandwidth-Based CAC for IP Multicast Feature Design	51
Mechanics of the Bandwidth-Based Multicast CAC Policies	51
Tips for Configuring Bandwidth-Based CAC Policies for IP Multicast	52
How to Configure Multicast Admission Control	52
Configuring Global and Per MVRF Mroute State Limiters	52
Prerequisites	53
Configuring a Global Mroute State Limiter	53
What to Do Next	54
Configuring Per MVRF Mroute State Limiters	54
Configuring MSDP SA Limiters	56
Configuring IGMP State Limiters	58
Prerequisites	58
Configuring Global IGMP State Limiters	58
What to Do Next	59
Configuring Per Interface IGMP State Limiters	59
Configuring Per Interface Mroute State Limiters	61
What to Do Next	62
Configuring Bandwidth-Based Multicast CAC Policies	62
What to Do Next	65
Monitoring Per Interface Mroute State Limiters and Bandwidth-Based Multicast CAC Policies	66
Configuration Examples for Configuring Multicast Admission Control	67
Configuring Global and Per MVRF Mroute State Limiters Example	68
Configuring MSDP SA Limiters Example	68
Example: Configuring IGMP State Limiters	68
Example Configuring Per Interface Mroute State Limiters	69
Example: Configuring Bandwidth-Based Multicast CAC Policies	71
Additional References	73
Feature Information for Configuring Multicast Admission Control	75
IPv6 Multicast: Bandwidth-Based Call Admission Control	79
Finding Feature Information	79
Information About IPv6 Multicast: Bandwidth-Based Call Admission Control	79

Bandwidth-Based CAC for IPv6 Multicast	79
How to Implement IPv6 Multicast Bandwidth-Based Call Admission Control	80
Configuring the Global Limit for Bandwidth-Based CAC in IPv6	80
Configuring an Access List for Bandwidth-Based CAC in IPv6	81
Configuring the Interface Limit for Bandwidth-Based CAC in IPv6	82
Configuration Examples for IPv6 Multicast Bandwidth-Based Call Admission Control	83
Example: Configuring the Global Limit for Bandwidth-Based CAC	83
Example: Configuring an Access List for Bandwidth-Based CAC in IPv6	83
Example: Configuring the Interface Limit for Bandwidth-Based CAC in IPv6	83
Additional References	83
Feature Information for IPv6 Multicast: Bandwidth-Based Call Admission Control	84
PIM Dense Mode State Refresh	87
Finding Feature Information	87
Prerequisite for PIM Dense Mode State Refresh	87
Restrictions on PIM Dense Mode State Refresh	87
Information About PIM Dense Mode State Refresh	88
PIM Dense Mode State Refresh Overview	88
Benefits of PIM Dense Mode State Refresh	88
How to Configure PIM Dense Mode State Refresh	88
Configuring PIM Dense Mode State Refresh	88
Verifying PIM Dense Mode State Refresh Configuration	89
Monitoring and Maintaining PIM DM State Refresh	89
Configuration Examples for PIM Dense Mode State Refresh	90
Example: Originating Processing and Forwarding PIM Dense Mode State Refresh Control Messages	90
Example: Processing and Forwarding PIM Dense Mode State Refresh Control Messages	90
Additional References	91



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.

- [Finding Feature Information, page 1](#)
- [Prerequisites for Optimizing PIM Sparse Mode in a Large IP Multicast Deployment, page 1](#)
- [Information About Optimizing PIM Sparse Mode in a Large IP Multicast Deployment, page 2](#)
- [How to Optimize PIM Sparse Mode in a Large IP Multicast Deployment, page 5](#)
- [Configuration Examples for Optimizing PIM Sparse Mode in a Large IP Multicast Deployment, page 7](#)
- [Additional References, page 7](#)
- [Feature Information for Optimizing PIM Sparse Mode in a Large IP Multicast Deployment, page 8](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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

Prerequisites for Optimizing PIM Sparse Mode in a Large IP Multicast Deployment

- You must have PIM sparse mode running in your network.
- If you plan to use a group list to control to which groups the shortest-path tree (SPT) threshold applies, you must have configured your access list before performing the task.

Information About Optimizing PIM Sparse Mode in a Large IP Multicast Deployment

- [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 3](#)

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, page 2](#)

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

Devices configured for IP multicast send PIM hello messages to determine which device will be the designated router (DR) for each LAN segment (subnet). The hello messages contain the device's IP address, and the device 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 devices send PIM router query messages every 30 seconds. By enabling a device to send PIM hello messages more often, the device can discover unresponsive neighbors more quickly. As a result, the device can implement failover or recovery procedures more efficiently. It is appropriate to make this change only on redundant devices 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.

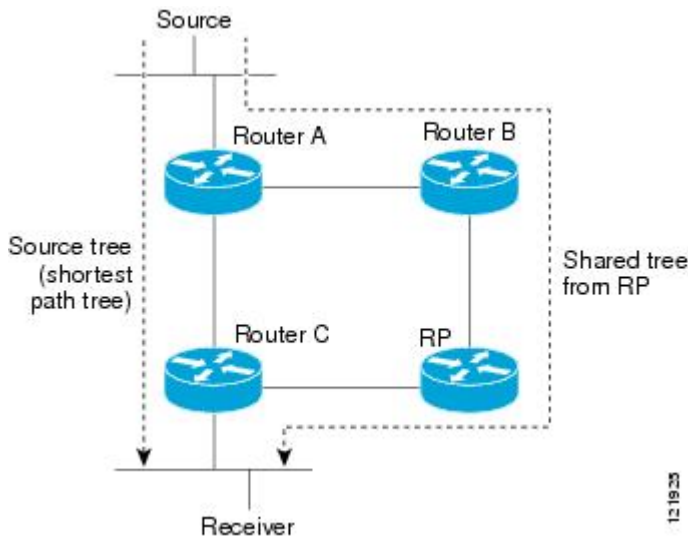
- [PIM Shared Tree and Source Tree - Shortest-Path Tree, page 3](#)
- [Benefit of Preventing or Delaying the Use of the Shortest-Path Tree, page 4](#)

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 the figure. 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 device (Router C in [Benefit of Preventing or Delaying the Use of the Shortest-Path Tree, page 4](#)). 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 device 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 device 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 device 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

- [Optimizing PIM Sparse Mode in a Large Deployment, page 5](#)

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	<code>enable</code>	Enables privileged EXEC mode.
	Example:	
	<code>Router> enable</code>	<ul style="list-style-type: none"> • Enter your password if prompted.

Command or Action	Purpose
<p>Step 2 <code>configure terminal</code></p> <p>Example:</p> <pre>Router# configure terminal</pre>	Enters global configuration mode.
<p>Step 3 <code>ip pim register-rate-limit rate</code></p> <p>Example:</p> <pre>Router(config)# ip pim register-rate-limit 10</pre>	<p>(Optional) Sets a limit on the maximum number of PIM sparse mode register messages sent per second for each (S, G) routing entry.</p> <ul style="list-style-type: none"> 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.
<p>Step 4 <code>ip pim spt-threshold {kpbs infinity}[group-list access-list]</code></p> <p>Example:</p> <pre>Router(config)# ip pim spt-threshold infinity group-list 5</pre>	<p>(Optional) Specifies the threshold that must be reached before moving to the shortest-path tree.</p> <ul style="list-style-type: none"> 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
<p>Step 5 <code>interface type number</code></p> <p>Example:</p> <pre>Router(config)# interface ethernet 0</pre>	<p>Configures an interface.</p> <ul style="list-style-type: none"> 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.
<p>Step 6 <code>ip pim query-interval period [msec]</code></p> <p>Example:</p> <pre>Router(config-if)# ip pim query-interval 1</pre>	<p>(Optional) Configures the frequency at which multicast routers send PIM router query messages.</p> <ul style="list-style-type: none"> Perform this step only on redundant routers on the edge of a PIM domain. The default query interval is 30 seconds. The <i>period</i> 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

- [Optimizing PIM Sparse Mode in a Large IP Multicast Deployment Example, page 7](#)

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
!
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco IOS IP SLAs commands	Cisco IOS IP Multicast Command Reference
PIM Sparse Mode concepts and configuration	“Configuring Basic IP Multicast” module or “Configuring IP Multicast in IPv6 Networks” module

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 XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Optimizing PIM Sparse Mode in a Large IP Multicast Deployment

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

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

Table 1 Feature Information for Optimizing PIM Sparse Mode in a Large IP Multicast Deployment

Feature Name	Releases	Feature Configuration Information
PIM Version 2	12.2(4)T	Protocol Independent Multicast (PIM) version 2 builds upon the success of the existing PIMv1 base, has two basic operating modes: sparse-mode and dense-mode, and is suitable for large networks with heterogeneous links and devices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

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



Multicast 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.

- [Finding Feature Information, page 11](#)
- [Prerequisites for Multicast Subsecond Convergence, page 11](#)
- [Restrictions for Multicast Subsecond Convergence, page 11](#)
- [Information About Multicast Subsecond Convergence, page 12](#)
- [How to Configure Multicast Subsecond Convergence, page 14](#)
- [Configuration Examples for Multicast Subsecond Convergence, page 18](#)
- [Additional References, page 19](#)
- [Feature Information for Multicast Subsecond Convergence, page 20](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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

Prerequisites for Multicast 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

Devices that use the subsecond designated router (DR) failover enhancement must be able to process hello interval information arriving in milliseconds. Devices that are congested or do not have enough CPU cycles to process the hello interval can assume that the Protocol Independent Multicast (PIM) neighbor is disconnected, although this may not be the case.

Information About Multicast Subsecond Convergence

- [Benefits of Multicast Subsecond Convergence, page 12](#)
- [Multicast Subsecond Convergence Scalability Enhancements, page 12](#)
- [PIM Router Query Messages, page 12](#)
- [Reverse Path Forwarding, page 12](#)
- [RPF Checks, page 13](#)
- [Triggered RPF Checks, page 13](#)
- [RPF Failover, page 13](#)
- [Topology Changes and Multicast Routing Recovery, page 13](#)

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 devices. Before the introduction of this feature, the device could send the PIM hellos only every few seconds. By enabling a device to send PIM hello messages more often, this feature allows the device to discover unresponsive neighbors more quickly. As a result, the device 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.

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.

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.

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 device. 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 device with PIM RPF changes while the routing table is still converging.

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

- [Modifying the Periodic RPF Check Interval, page 14](#)
- [Configuring PIM RPF Failover Intervals, page 15](#)
- [Modifying the PIM Router Query Message Interval, page 16](#)
- [Verifying Multicast Subsecond Convergence Configurations, page 17](#)

Modifying the Periodic RPF Check Interval

Perform this optional task to modify the intervals at which periodic RPF checks occur.



Note

Cisco recommends that you do *not* change 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	<p>enable</p> <p>Example:</p> <pre>Device> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example:</p> <pre>Device# configure terminal</pre>	<p>Enters global configuration mode.</p>
Step 3	<p>ip multicast rpf interval <i>seconds</i> [list <i>access-list</i> route-map <i>route-map</i>]</p> <p>Example:</p> <pre>Device(config)# ip multicast rpf interval 10</pre>	<p>Configures the periodic RPF check intervals to occur at a specified interval, in seconds.</p>

- [What to Do Next, page 15](#)

What to Do Next

Proceed to the [Configuring PIM RPF Failover Intervals, page 15](#) 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, page 16](#) to modify the interval at which IGMP host query messages are sent. Proceed to the [What to Do Next, page 15](#) to display information about and to verify information regarding the Multicast Subsecond Convergence feature.

Configuring PIM RPF Failover Intervals

Perform this optional task to configure the intervals at which PIM RPF failover will be triggered by changes in the routing tables.



Note

Cisco recommends that you do *not* modify 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: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip multicast rpf backoff <i>minimum maximum</i> [disable] Example: Device(config)# ip multicast rpf backoff 100 2500	Configures the minimum and the maximum backoff intervals.

- [What to Do Next, page 16](#)

What to Do Next

Proceed to the [Modifying the PIM Router Query Message Interval, page 16](#) to modify the interval at which IGMP host query messages are sent. Proceed to the [What to Do Next, page 16](#) 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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot / subslot / port*
4. **ip pim query-interval** *period* [msec]

DETAILED STEPS

Command or Action	Purpose
Step 1 enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2 configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3 interface <i>type slot / subslot / port</i> Example: Device(config)# interface gigabitethernet 1/0/0	Specifies the interface and enters interface configuration mode.
Step 4 ip pim query-interval <i>period</i> [msec] Example: Device(config-if)# ip pim query-interval 45	Configures the frequency at which multicast routers send PIM router query messages.

- [What to Do Next, page 17](#)

What to Do Next

Proceed to the [What to Do Next, page 17](#) to display and verify information about 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**

DETAILED STEPS

Step 1 enable

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

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:

Example:

```
Device# show ip pim interface GigabitEthernet 1/0/0
Address      Interface          Ver/   Nbr   Query  DR    DR
              Mode              Count  Intvl Prior
172.16.1.4   GigabitEthernet1/0/0 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 XE software.

The following is sample output from the **show ip pim neighbor** command:

Example:

```
Device# show ip pim neighbor
PIM Neighbor Table
Neighbor      Interface          Uptime/Expires   Ver   DR
Address                               Prio/Mode
172.16.1.3    GigabitEthernet1/0/0 00:03:41/250 msec v2    1 / S
```

Configuration Examples for Multicast Subsecond Convergence

- [Example Modifying the Periodic RPF Check Interval, page 18](#)
- [Example Configuring PIM RPF Failover Intervals, page 18](#)
- [Modifying the PIM Router Query Message Interval Example, page 19](#)

Example Modifying the Periodic RPF Check Interval

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  
!
```

Example Configuring PIM RPF Failover Intervals

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 gigabitethernet0/0/1
 ip address 172.16.2.1 255.255.255.0
 ip pim query-interval 100 msec
 ip pim sparse-mode
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	<i>Cisco IOS Master Commands List, All Releases</i>
Cisco IOS IP Multicast commands	<i>Cisco IOS IP Multicast Command Reference</i>

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 XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Multicast Subsecond Convergence

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

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

Table 2 Feature Information for Multicast Subsecond Convergence

Feature Name	Releases	Feature Information
Multicast Subsecond Convergence	12.0(22)S 12.2(14)S 12.2(15)T Cisco IOS XE Release 2.1 15.0(1)S Cisco IOS XE Release 3.1.0SG Cisco IOS XE Release 3.2SE	<p>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.</p> <p>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.</p>

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

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



IP Multicast Load Splitting across Equal-Cost Paths

This module describes how to load split IP multicast traffic from different sources, or from different sources and groups, over Equal Cost Multipath (ECMP) to take advantage of multiple paths through the network.

- [Finding Feature Information, page 21](#)
- [Prerequisites for IP Multicast Load Splitting across Equal-Cost Paths, page 21](#)
- [Information About IP Multicast Load Splitting across Equal-Cost Paths, page 21](#)
- [How to Load Split IP Multicast Traffic over ECMP, page 31](#)
- [Configuration Examples for Load Splitting IP Multicast Traffic over ECMP, page 38](#)
- [Additional References, page 39](#)
- [Feature Information for Load Splitting IP Multicast Traffic over ECMP, page 39](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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

Prerequisites for IP Multicast Load Splitting across Equal-Cost Paths

IP multicast is enabled on the device using the tasks described in the “Configuring Basic IP Multicast” module of the *IP Multicast: PIM Configuration Guide*.

Information About IP Multicast Load Splitting across Equal-Cost Paths

- [Load Splitting Versus Load Balancing, page 22](#)
- [Default Behavior for IP Multicast When Multiple Equal-Cost Paths Exist, page 22](#)
- [Methods to Load Split IP Multicast Traffic, page 24](#)
- [Overview of ECMP Multicast Load Splitting, page 24](#)
- [Effect of ECMP Multicast Load Splitting on PIM Neighbor Query and Hello Messages for RPF Path Selection, page 27](#)
- [Effect of ECMP Multicast Loading Splitting on Assert Processing in PIM-DM and DF Election in Bidir-PIM, page 28](#)
- [Effect of ECMP Multicast Load Splitting on the PIM Assert Process in PIM-SM and PIM-SSM, page 29](#)
- [ECMP Multicast Load Splitting and Reconvergence When Unicast Routing Changes, page 30](#)
- [Use of BGP with ECMP Multicast Load Splitting, page 30](#)
- [Use of ECMP Multicast Load Splitting with Static Mroutes, page 30](#)
- [Alternative Methods of Load Splitting IP Multicast Traffic, page 31](#)

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 equal-cost multipath (ECMP) multicast load splitting methods and result in better load-sharing in networks where there are many traffic streams that utilize approximately the same amount of bandwidth.

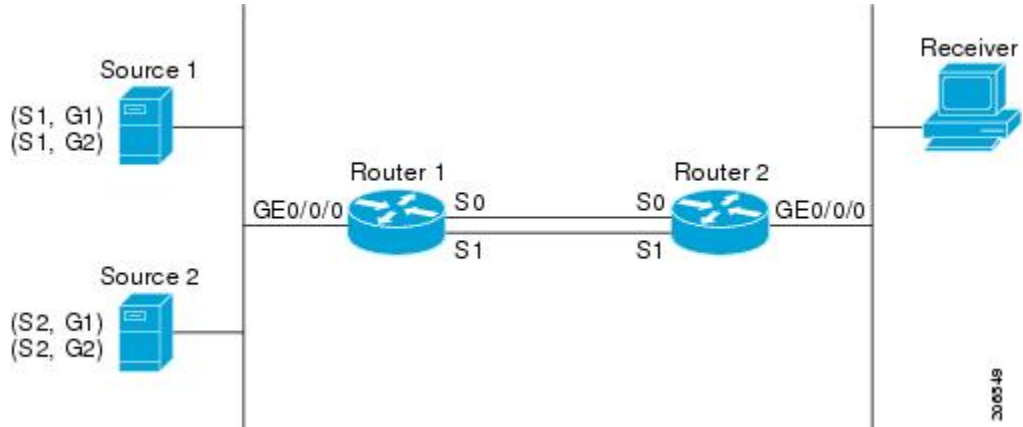
If there are just a few (S, G) or (*, G) states flowing across a set of equal-cost links, the chance that they are well balanced is quite low. To overcome this limitation, precalculated source addresses--for (S, G) states or rendezvous point (RP) addresses for (*, G) states, can be used to achieve a 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.

The figure 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 2 Default Behavior for IP Multicast When Multiple Equal-Cost Paths Exist



In the figure, 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 Device 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 Device 2) displays serial interface 0 and serial interface 1 on Device 1 as equal-cost next-hop PIM neighbors of Device 2.

Without further configuration, IPv4 multicast traffic in the topology illustrated in the figure 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 Device 1 are 10.1.1.1 and 10.1.2.1, respectively. Given that scenario, in the case of PIM-SM and PIM-SSM, Device 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 the figure. In the case of PIM-DM, Device 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 Device 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 Device 1.

IPv4 RPF lookups are performed by intermediate multicast device 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 device 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, page 25](#) section.
- ECMP multicast load splitting method based on source and group address--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, page 25](#) section.
- ECMP multicast load splitting method based on source, group, and next-hop address--You can configure ECMP multicast load splitting using the **ip multicast multipath** command with the **s-g-hash** and **next-hop-based** keywords. Entering this form of the command enables ECMP multicast load splitting 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, page 26](#) 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, page 25](#)
- [ECMP Multicast Load Splitting Based on Source and Group Address Using the Basic S-G-Hash Algorithm, page 25](#)
- [Predictability As a By-Product of Using the S-Hash and Basic S-G-Hash Algorithms, page 25](#)
- [Polarization As a By-Product of Using the S-Hash and Basic S-G-Hash Algorithms, page 25](#)
- [ECMP Multicast Load Splitting Based on Source Group and Next-Hop Address, page 26](#)

ECMP Multicast Load Splitting Based on Source Address Using the S-Hash Algorithm

ECMP multicast load splitting traffic based on source address uses the S-hash algorithm, enabling the RPF interface for each (*, G) or (S, G) state to 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.

ECMP Multicast Load Splitting Based on Source and Group Address Using the Basic S-G-Hash Algorithm

ECMP multicast load splitting based on source and group address uses 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 device this hash is being calculated on.



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

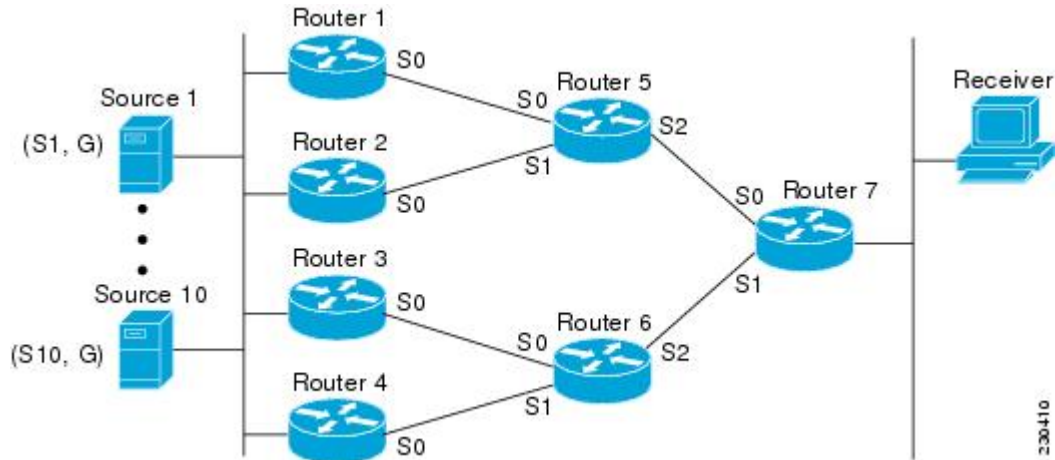
The method used by ECMP multicast load splitting in IPv4 multicast allows for consistent load splitting in a network where the same number of equal-cost paths are present in multiple places in a topology. If an RP address or source addresses are calculated once to have flows split across N paths, then they will be split across those N paths in the same way in all places in the topology. Consistent load splitting allows for predictability, which, in turn, 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

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.

The figure 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 3 Polarization Topology



In the topology illustrated in the figure, 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

Configuring ECMP multicast load splitting based on source, group, and next-hop address 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 device (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 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 device, 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 device 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

If load splitting of IP multicast traffic over ECMP 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 device from which PIM hello (or PIMv1 query) messages are received. For example, consider a device 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 devices 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 devices sends PIM hello messages, then 10.1.2.1 would be selected. This deference 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](http://ftpeng.cisco.com/ipmulticast/config-notes/static-mroutes.txt) configuration note on the Cisco IOS IP multicast FTP site, which is available at: [ftp://ftpeng.cisco.com/ipmulticast /config-notes/static-mroutes.txt](http://ftpeng.cisco.com/ipmulticast/config-notes/static-mroutes.txt).

When load splitting of IP multicast traffic over ECMP 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

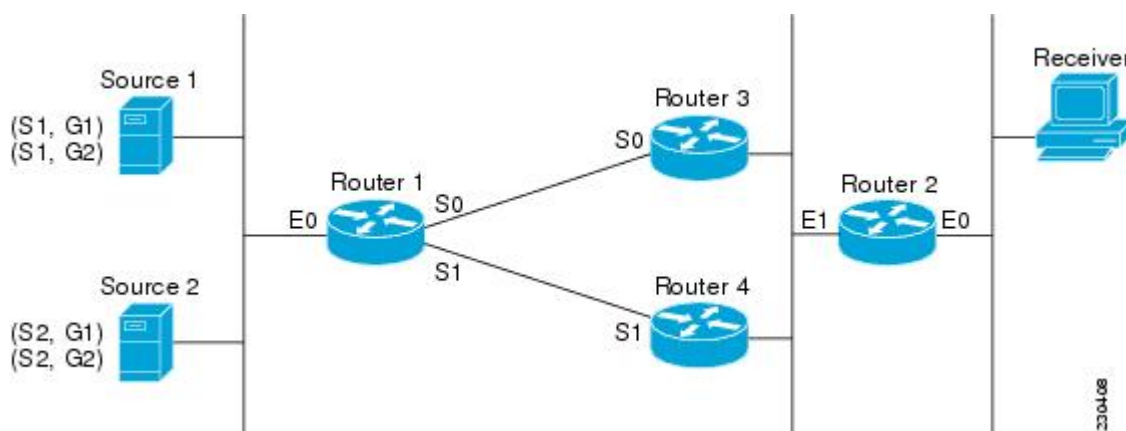
messages are received from that neighbor--it only depends on the presence or absence of an equal-cost route entry.

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 device; it does not have an effect on designated forwarder (DF) election in bidir-PIM or the assert processing on upstream devices in PIM-DM.

The figure 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 4 ECMP Multicast Load Splitting and Assert Processing in PIM-DM and DF Election in Bidir-PIM



In the figure, Device 2 has two equal-cost paths to S1 and S2 and the RP addresses on Device 1. Both paths are across Gigabit Ethernet interface 1/0/0: one path towards Device 3 and one path towards Device 4. For PIM-SM and PIM-SSM (*, G) and (S, G) RPF selection, there is no difference in the behavior of Device 2 in this topology versus Device 2 in the topology illustrated in the figure. There is, however, a difference when using PIM-DM or bidir-PIM.

If PIM-DM is used in the topology illustrated in the figure, Device 3 and Device 4 would start flooding traffic for the states onto Gigabit Ethernet interface 1/0/0 and would use the PIM assert process to elect one device among them to forward the traffic and to avoid traffic duplication. As both Device 3 and Device 4 would have the same route cost, the device with the higher IP address on Gigabit Ethernet interface 1/0/0 would always win the assert process. As a result, if PIM-DM is used in this topology, traffic would not be load split across Device 3 and Device 4.

If bidir-PIM is used in the topology illustrated in the figure, a process called DF election would take place between Device 2, Device 3, and Device 4 on Gigabit Ethernet interface 1/0/0. The process of DF election would elect one device for each RP to forward traffic across Gigabit Ethernet interface 1/0/0 for any groups using that particular RP, based on the device 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 device that has the higher IP address configured on Gigabit Ethernet interface 1/0/0 (either Device 3 or Device 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 Gigabit Ethernet interface 1/0/0.

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 devices. Changing them would require some form of protocol change that would also need to be agreed upon by the participating devices. RPF selection is a purely device local policy and, thus, can be enabled or disabled without protocol changes individually on each device.

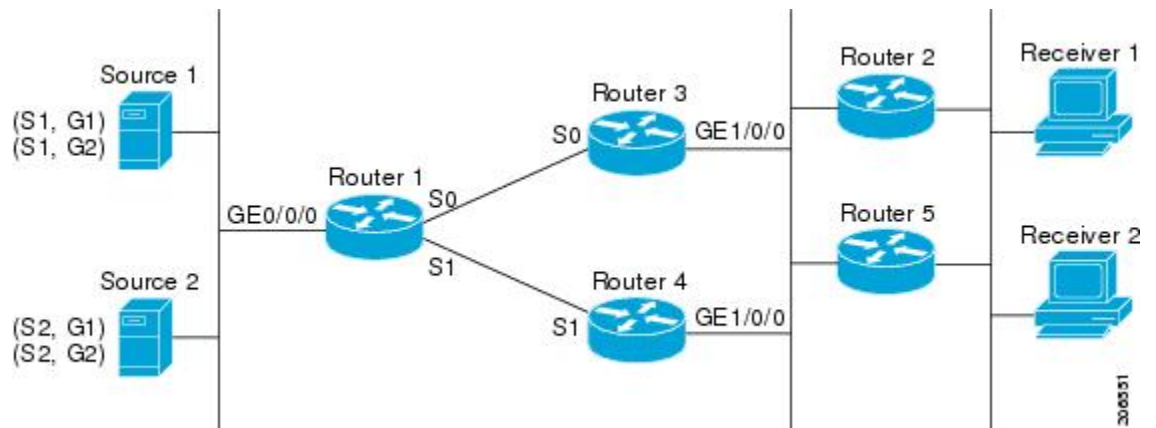
For PIM-DM and bidir-PIM, configuring ECMP multicast load splitting with the **ip multicast multipath** command is only effective in topologies where the equal-cost paths are not upstream PIM neighbors on the same LAN, but rather neighbors on different LANs or point-to-point links.

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.

The figure 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 5 ECMP Multicast Load Splitting and the PIM Assert Process in PIM-SM and PIM-SSM



In the topology illustrated in the figure, if both Device 2 and Device 5 are Cisco devices 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 devices would have Device 3 and Device 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 Device 3 or Device 4) and send their PIM joins to this neighbor.

If Device 5 and Device 2 are inconsistently configured with the **ip multicast multipath** command, or if Device 5 is a third-party device, then Device 2 and Device 5 may choose different RPF neighbors for some (*, G) or (S, G) states. For example Device 2 could choose Device 3 for a particular (S, G) state or Device 5 could choose Device 4 for a particular (S, G) state. In this scenario, Device 3 and Device 4 would both start to forward traffic for that state onto Gigabit Ethernet interface 1/0/0, see each other's forwarded traffic, and--to avoid traffic duplication--start the assert process. As a result, for that (S, G) state, the device

with the higher IP address for Gigabit Ethernet interface 1/0/0 would forward the traffic. However, both Device 2 and Device 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 device 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 devices on a LAN are consistently configured Cisco devices.

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 load splitting of IP multicast traffic over ECMP 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 device 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. Once configured, when BGP installs the remote next-hop information, RPF lookups will execute recursively 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.

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 to specify the equal-cost paths for load splitting. You cannot use static mroutes to configure equal-cost paths because the 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](http://ftpeng.cisco.com/ipmulticast/config-notes/static-mroutes.txt) configuration note on the Cisco IOS IP multicast FTP site at [ftp://ftpeng.cisco.com/ipmulticast/config-notes/static-mroutes.txt](http://ftpeng.cisco.com/ipmulticast/config-notes/static-mroutes.txt).

You can specify only static mroutes for equal-cost multipaths in IPv4 multicast; however, those static mroutes would only apply to multicast, or you can specify that the equal-cost multipaths apply to both unicast and multicast routing. In IPv6 multicast, there is no such restriction. Equal-cost multipath mroutes can be configured for static IPv6 mroutes that apply to only unicast routing, only multicast routing, or both unicast and multicast routing.

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. 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.

Before load splitting IP multicast traffic across equal-cost paths over a tunnel, you must configure CEF per-packet load balancing or else the GRE packets will not be load balanced per packet.

How to Load Split IP Multicast Traffic over ECMP

- [Enabling ECMP Multicast Load Splitting, page 31](#)

Enabling ECMP Multicast Load Splitting

Perform the following tasks to load split IP multicast traffic across multiple equal-cost paths, based on source address.

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.

- [Prerequisites for IP Multicast Load Splitting - ECMP, page 32](#)
- [Restrictions, page 32](#)
- [Enabling ECMP Multicast Load Splitting Based on Source Address, page 32](#)

- [Enabling ECMP Multicast Load Splitting Based on Source and Group Address, page 34](#)
- [Enabling ECMP Multicast Load Splitting Based on Source Group and Next-Hop Address, page 36](#)

Prerequisites for IP Multicast Load Splitting - ECMP

- You must have an adequate number of sources (at least more than two sources) to enable ECMP multicast load splitting based on source address.
- You must have multiple paths available to the RP to configure ECMP multicast load splitting.



Note

Use the **show ip route** command with either the IP address of the source for the *ip-address* argument or the IP address of the RP to validate that there are multiple paths available to the source or RP, respectively. If you do not see multiple paths in the output of the command, you will not be able to configure ECMP multicast load splitting.

- When using PIM-SM with shortest path tree (SPT) forwarding, the T-bit must be set for the forwarding of all (S, G) states.
- Before 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, page 30](#) section.

Restrictions

- 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.
- 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.
- 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.

Enabling ECMP Multicast Load Splitting Based on Source Address

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 device on which the hash is being calculated.

**Note**

Enable ECMP multicast load splitting on the device that is 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 device connecting to more than one outgoing interfaces.

- You must have an adequate number of sources (at least more than two sources) to enable ECMP multicast load splitting based on source address.
- You must have multiple paths available to the RP to configure ECMP multicast load splitting.

**Note**

Use the **show ip route** command with either the IP address of the source for the *ip-address* argument or the IP address of the RP to validate that there are multiple paths available to the source or RP, respectively. If you do not see multiple paths in the output of the command, you will not be able to configure ECMP multicast load splitting.

- When using PIM-SM with shortest path tree (SPT) forwarding, the T-bit must be set for the forwarding of all (S, G) states.
- Before 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, page 30](#) section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast multipath**
4. Repeat step 3 on all the devices in a redundant topology.
5. **exit**
6. **show ip rpf source-address [group-address]**
7. **show ip route ip-address**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

Command or Action	Purpose
<p>Step 3 <code>ip multicast multipath</code></p> <p>Example:</p> <pre>Device(config)# ip multicast multipath</pre>	<p>Enables ECMP multicast load splitting based on source address using the S-hash algorithm.</p> <ul style="list-style-type: none"> Because this command changes the way an RPF neighbor is selected, it must be configured consistently on all devices in a redundant topology to avoid looping. This 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 a different IP address for each interface in a device on which this command is to be configured. This 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.
<p>Step 4 Repeat step 3 on all the devices in a redundant topology.</p>	--
<p>Step 5 <code>exit</code></p> <p>Example:</p> <pre>Device(config)# exit</pre>	Exits global configuration mode and returns to privileged EXEC mode.
<p>Step 6 <code>show ip rpf source-address [group-address]</code></p> <p>Example:</p> <pre>Device# show ip rpf 10.1.1.2</pre>	<p>(Optional) Displays the information that IP multicast routing uses to perform the RPF check.</p> <ul style="list-style-type: none"> Use this command to verify RPF selection so as to ensure that IP multicast traffic is being properly load split.
<p>Step 7 <code>show ip route ip-address</code></p> <p>Example:</p> <pre>Device# show ip route 10.1.1.2</pre>	<p>(Optional) Displays the current state of the IP routing table.</p> <ul style="list-style-type: none"> Use this command to verify that there multiple paths available to a source or RP for ECMP multicast load splitting. For the <i>ip-address</i> 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

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 device on which the hash is being calculated.

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.

**Note**

Enable ECMP multicast load splitting on the device that is 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 device connecting to more than one outgoing interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast multipath s-g-hash basic**
4. Repeat Step 3 on all the devices in a redundant topology.
5. **exit**
6. **show ip rpf source-address [group-address]**
7. **show ip route ip-address**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip multicast multipath s-g-hash basic Example: Device(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. <ul style="list-style-type: none"> • Because this command changes the way an RPF neighbor is selected, it must be configured consistently on all devices in a redundant topology to avoid looping.
Step 4	Repeat Step 3 on all the devices in a redundant topology.	--

Command or Action	Purpose
Step 5 <code>exit</code> Example: <pre>Device(config)# exit</pre>	Exits global configuration mode and returns to privileged EXEC mode.
Step 6 <code>show ip rpf source-address [group-address]</code> Example: <pre>Device# show ip rpf 10.1.1.2</pre>	(Optional) Displays the information that IP multicast routing uses to perform the RPF check. <ul style="list-style-type: none"> Use this command to verify RPF selection so as to ensure that IP multicast traffic is being properly load split.
Step 7 <code>show ip route ip-address</code> Example: <pre>Device# show ip route 10.1.1.2</pre>	(Optional) Displays the current state of the IP routing table. <ul style="list-style-type: none"> Use this command to verify that there multiple paths available to a source or RP for ECMP multicast load splitting. For the <i>ip-address</i> 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

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

- enable**
- configure terminal**
- ip multicast multipath s-g-hash next-hop-based**
- Repeat Steps 1 through 3 on all the routers in a redundant topology.
- end**
- show ip rpf source-address [group-address]**
- show ip route ip-address**

DETAILED STEPS

Command or Action	Purpose
<p>Step 1 <code>enable</code></p> <p>Example:</p> <pre>Router> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
<p>Step 2 <code>configure terminal</code></p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 <code>ip multicast multipath s-g-hash next-hop-based</code></p> <p>Example:</p> <pre>Router(config)# ip multicast multipath s-g-hash next-hop-based</pre>	<p>Enables ECMP multicast load splitting based on source, group, and next-hop-address using the next-hop-based S-G-hash algorithm.</p> <ul style="list-style-type: none"> • 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. <p>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.</p>
<p>Step 4 Repeat Steps 1 through 3 on all the routers in a redundant topology.</p>	<p>--</p>
<p>Step 5 <code>end</code></p> <p>Example:</p> <pre>Router(config)# end</pre>	<p>Exits global configuration mode and returns to privileged EXEC mode.</p>
<p>Step 6 <code>show ip rpf source-address [group-address]</code></p> <p>Example:</p> <pre>Router# show ip rpf 10.1.1.2</pre>	<p>(Optional) Displays the information that IP multicast routing uses to perform the RPF check.</p> <ul style="list-style-type: none"> • Use this command to verify RPF selection so as to ensure that IP multicast traffic is being properly load split.

Command or Action	Purpose
<p>Step 7 <code>show ip route <i>ip-address</i></code></p> <p>Example:</p> <pre>Router# show ip route 10.1.1.2</pre>	<p>(Optional) Displays the current state of the IP routing table.</p> <ul style="list-style-type: none"> Use this command to verify that there multiple paths available to a source or RP for ECMP multicast load splitting. For the <i>ip-address</i> 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

- [Example Enabling ECMP Multicast Load Splitting Based on Source Address, page 38](#)
- [Example Enabling ECMP Multicast Load Splitting Based on Source and Group Address, page 38](#)
- [Example Enabling ECMP Multicast Load Splitting Based on Source Group and Next-Hop Address, page 38](#)

Example Enabling ECMP Multicast Load Splitting Based on Source Address

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
```

Example Enabling ECMP Multicast Load Splitting Based on Source and Group Address

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
```

Example Enabling ECMP Multicast Load Splitting Based on Source Group and Next-Hop Address

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

Related Documents

Related Topic	Document Title
Cisco IOS commands	<i>Cisco IOS Master Commands List, All Releases</i>
Cisco IOS IP Multicast commands	<i>Cisco IOS IP Multicast Command Reference</i>

Standards and RFCs

Standard/RFC	Title
<i>RFC 4601</i>	Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification

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 XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Load Splitting IP Multicast Traffic over ECMP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software

release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 3 **Feature Information for Load Splitting IP Multicast Traffic over ECMP**

Feature Name	Releases	Feature Information
IP Multicast Load Splitting across Equal-Cost Paths	--	<p>The IP Multicast Load Splitting Across Equal Paths feature enables load splitting of IP multicast traffic across equal-cost paths. Prior to this feature, when there were equal-cost paths between routers, IP multicast packets traversed only one path. If a tunnel was configured, the same next hop was always used, and no load splitting occurred.</p> <p>The following commands were introduced or modified: ip multicast multipath.</p>

Feature Name	Releases	Feature Information
IP Multicast Load Splitting-- Equal Cost Multipath (ECMP) Using S, G and Next Hop	12.2(33)SRB 15.0(1)M 15.0(1)S	<p>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.</p> <p>The following commands were introduced or modified: ip multicast multipath.</p>

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

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



Configuring Multicast Admission Control

This module describes how to implement multicast admission control in an IP multicast network. Multicast admission control features are configured on multicast-enabled routers to prevent control plane overload, ensure proper resource allocation, and provide multicast Call Admission Control (CAC) capabilities.

- [Finding Feature Information, page 43](#)
- [Prerequisites for Configuring Multicast Admission Control, page 43](#)
- [Information About Configuring Multicast Admission Control, page 43](#)
- [How to Configure Multicast Admission Control, page 52](#)
- [Configuration Examples for Configuring Multicast Admission Control, page 67](#)
- [Additional References, page 73](#)
- [Feature Information for Configuring Multicast Admission Control, page 75](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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

Prerequisites for Configuring Multicast Admission Control

IP multicast is enabled and the Protocol Independent Multicast (PIM) interfaces are configured using the tasks described in the “Configuring Basic IP Multicast” module.

Information About Configuring Multicast Admission Control

- [Multicast Admission Control, page 44](#)
- [Multicast Admission Control Features, page 44](#)
- [Global and Per MVRF Mroute State Limit, page 45](#)
- [MSDP SA Limit, page 46](#)
- [IGMP State Limit, page 47](#)

- [Per Interface Mroute State Limit](#), page 48
- [Bandwidth-Based CAC for IP Multicast](#), page 51

Multicast Admission Control

As the popularity of network video applications grows among consumers, admission control functions--which govern transmission and reception of multicast traffic based on available network resources--are vital. Without admission control, some users may receive degraded multicast streams, rendering programs unwatchable, and others may receive a “Network Busy” message or nothing at all as network resources are overtaxed. Network admission control is important in maintaining a high quality of experience for digital video consumers.

The goals of multicast admission control features, therefore, are as follows:

- Protect the router from control plane overload to ensure that memory and CPU resources on multicast-enabled routers are not overrun by multicast route (mroute) states or denial-of-service (DoS) attacks from multicast packets.
- Enable proper resource allocation (on a global, per MVRF, or per interface basis) to ensure that multicast services are delivered to subscribers per their IP Service Level Agreements (SLAs) and to minimize the effects of DoS attacks on subscribers.
- Provide multicast CAC capabilities to prevent bandwidth resources (interfaces, subnetworks) from being congested and to enable service providers to offer more flexible and refined content and subscriber-based policies.

Multicast Admission Control Features

The Cisco IOS software supports the following multicast admission control features:

- Global and Per MVRF Mroute State Limit

The **ip multicast route-limit** command allows for the configuration of global and per MVRF state limiters, which impose limits on the number of multicast routes (mroutes) that can be added to the global table or to a particular Multicast Virtual Routing and Forwarding (MVRF) table.

- MSDP SA Limit

The **ip msdp sa-limit** command allows for the configuration of MSDP SA limiters, which impose limits on the number of Multicast Source Discovery Protocol (MSDP) Source Active (SA) messages that can be cached on a router.

For more information about MSDP, see the “Using MSDP to Interconnect Multiple PIM-SM Domains” module.

- IGMP State Limit

This feature allows for the configuration of IGMP state limiters, which impose limits on mroute states resulting from Internet Group Management Protocol (IGMP) membership reports (IGMP joins).

- Per Interface Mroute State Limit

This feature allows for the configuration of per interface mroute state limiters, which impose mroute state limits for different access control list (ACL)-classified sets of multicast traffic on an interface.

- Bandwidth-Based CAC for IP Multicast

This feature allows for the configuration of bandwidth-based multicast CAC policies, which allow for bandwidth-based CAC on a per interface basis.

These admission control features may be invoked by service providers and enterprise network administrators based on different criteria, including the service package an end user has purchased or the privileges an enterprise user is entitled to.

Global and Per MVRF Mroute State Limit

The **ip multicast route-limit** command allows for the configuration of global and per MVRF mroute state limiters, which impose limits on the number of mroutes that can be added to the global table or to a particular MVRF table, respectively.

Global mroute state limiters are used to limit the number of mroutes that can be added to the global table on a router. Configuring a global mroute state limiter can protect a router in the event of a multicast DoS attack (by preventing mroutes from overrunning the router).

Per VRF mroute state limiters are used to limit the number of mroutes that can be added to an MVRF table on a Multicast VPN (MVPN) provider edge (PE) router. Configuring per MVRF mroute state limits can be used to ensure the fair sharing of mroutes between different MVRFs on an MVPN PE router.

- [Global and Per MVRF Mroute State Limit Feature Design, page 45](#)
- [Mechanics of Global and Per MVRF Mroute State Limiters, page 45](#)

Global and Per MVRF Mroute State Limit Feature Design

Global and per MVRF mroute state limiters are configured using the **ip multicast route-limit** command in global configuration mode. The syntax of the **ip multicast route-limit** command is as follows:

```
ip multicast [vrf vrf-name] route-limit limit [threshold]
```

Issuing the **ip multicast route-limit** command without the optional **vrf** keyword and *vrf-name* arguments configures a global mroute state limiter. The optional **vrf** keyword and *vrf-name* arguments are used with the **ip multicast limit** command to configure per MVRF mroute state limiters.



Note

When configuring global and per VRF mroute state limiters, you can only configure one limit for the global table and one limit per MVRF table.

The value specified for the required *limit* argument defines the maximum number of mroutes that can be added to either the global table or a particular MVRF table, respectively.



Note

Global and per MVRF mroute state limiters operate independently and can be used alone or together, depending upon the admission control requirements of your network.

In addition, for both global and per MVRF mroute state limiters, the optional *threshold* argument is available to set mroute threshold limits.

Mechanics of Global and Per MVRF Mroute State Limiters

The mechanics of global and per MVRF mroute state limiters are as follows:

- Each time the state for an mroute is created on a router, the Cisco IOS software checks to see if the limit for the global mroute state limiter (if the mroute is associated with the global table) or the limit for the per MVRF mroute state limiter (if the mroute is associated with the MVRF table) has been reached.

- States for mroutes that exceed the configured limit for the global or the per MVRF mroute state limiter are not created on the router, and a warning message in the following format is generated:

```
% MROUTE-4-ROUTE LIMIT : <current mroute count> exceeded multicast route-limit of
<mroute limit value>
```

- When an mroute threshold limit is also configured for the global or the per MVRF mroute state limiter, each time the state for an mroute is created on a router, the Cisco IOS software also checks to see if the mroute threshold limit has been reached. If the mroute threshold limit is exceeded, a warning message in the following format is generated:

```
% MROUTE-4-ROUTE LIMIT WARNING : multicast route-limit warning <current mroute count>
threshold <mroute threshold value>
```

Warning messages continue to be generated until the number of mroutes exceeds the configured limit or until the number of mroute states falls below the configured mroute threshold limit.

MSDP SA Limit

The **ip msdp sa-limit** command allows for the configuration of MSDP SA limiters, which impose limits on the number of MSDP Source Active (SA) messages that an MSDP-enabled router can accept (can be cached) from an MSDP peer. This command provides a means to protect an MSDP-enabled router from denial of service (DoS) attacks.

- [MSDP SA Limit Feature Design, page 46](#)
- [Mechanics of MSDP SA Limiters, page 46](#)
- [Tips for Configuring MSDP SA Limiters, page 47](#)

MSDP SA Limit Feature Design

MSDP SA limiters are configured using the **ip msdp sa-limit** command in global configuration mode. The syntax of the **ip msdp sa-limit** command is as follows:

```
ip msdp [vrf vrf-name] sa-limit {peer-address | peer-name} sa-limit
```

For the required *peer-address* argument or *peer-name* argument, specify either the MSDP peer address or MSDP peer name of the peer to be limited.

For the required *sa-limit* argument, specify the maximum number of SA messages that can be accepted (cached) from the specified peer. The range is from 1 to 2147483646.



Note

In an MVPN environment, the optional **vrf** keyword and *vrf-name* argument are used to specify the MVRF associated with the MSDP peer. When an MVRF is specified, the MSDP SA limiter is applied to the specified MSDP peer associated with the specified MVRF.

Mechanics of MSDP SA Limiters

- When MSDP SA limiters are configured, the router maintains a per-peer count of SA messages stored in the SA cache.
- SA messages that exceed the limit configured for an MSDP peer are ignored.

- If the router receives SA messages in excess of the configured limit from an MSDP peer, a warning in the following format is generated (once a minute) until the cache is cleared:

```
%MSDP-4-SA_LIMIT: SA from peer <peer address or name>, RP <RP address> for <mroute>
exceeded sa-limit of <configured SA limit for MSDP peer>
```

Tips for Configuring MSDP SA Limiters

- We recommended that you configure MSDP SA limiters for all MSDP peerings on the router.
- An appropriately low MSDP SA limit should be configured on peerings with a stub MSDP region (an MSDP peer that may have some further downstream peers but does not act as a transit for SA messages across the rest of the Internet).
- An appropriately high SA limit should be configured for all MSDP peerings that act as transits for MSDP SA messages across the Internet.

IGMP State Limit

The IGMP State Limit feature allows for the configuration of IGMP state limiters, which impose limits on mroute states resulting from IGMP membership reports (IGMP joins) on a global or per interface basis. Membership reports exceeding the configured limits are not entered into the IGMP cache. 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.



Note

IGMP state limiters impose limits on the number of mroute states resulting from IGMP, IGMP v3lite, and URL Rendezvous Directory (URD) membership reports on a global or per interface basis.

- [IGMP State Limit Feature Design, page 47](#)
- [Mechanics of IGMP State Limiters, page 47](#)

IGMP State Limit Feature Design

- Configuring IGMP state limiters in global configuration mode specifies a global limit on the number of IGMP membership reports that can be cached.
- Configuring IGMP state limiters in interface configuration mode specifies a limit on the number of IGMP membership reports on a per interface basis.
- Use ACLs to prevent groups or channels from being counted against the interface limit. A standard or an extended ACL can be specified. A standard ACL can be used to define the (*, G) state to be excluded from the limit on an interface. An extended ACLs can be used to define the (S, G) state to be excluded from the limit on an interface. An extended ACL also can be used to define the (*, G) state to be excluded from the limit on an interface, by specifying 0.0.0.0 for the source address and source wildcard--referred to as (0, G)--in the permit or deny statements that compose the extended access list.
- You can only configure one global limit per device and one limit per interface.

Mechanics of IGMP State Limiters

The mechanics of IGMP state limiters are as follows:

- Each time a router receives an IGMP membership report for a particular group or channel, the Cisco IOS software checks to see if either the limit for the global IGMP state limiter or the limit for the per interface IGMP state limiter has been reached.
- If only a global IGMP state limiter has been configured and the limit has not been reached, IGMP membership reports are honored. When the configured limit has been reached, subsequent IGMP membership reports are then ignored (dropped) and a warning message in one of the following formats is generated:
 - %IGMP-6-IGMP_GROUP_LIMIT: IGMP limit exceeded for <group (*, group address)> on <interface type number> by host <ip address>
 - %IGMP-6-IGMP_CHANNEL_LIMIT: IGMP limit exceeded for <channel (source address, group address)> on <interface type number> by host <ip address>
- If only per interface IGMP state limiters are configured, then each limit is only counted against the interface on which it was configured.
- If both a global IGMP state limiter and per interface IGMP state limiters are configured, the limits configured for the per interface IGMP state limiters are still enforced but are constrained by the global limit.

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 when all the multicast flows roughly utilize the same amount of bandwidth.

The Per Interface Mroute State Limit feature essentially is a complete superset of the IGMP State Limit feature (with the exception that it does not support a global limit). The Per Interface Mroute State Limit feature, moreover, is more flexible and powerful (albeit more complex) than the IGMP State Limit feature but is not intended to be a replacement for it because there are applications that suit both features.

The main differences between the Per Interface Mroute State Limit feature and the IGMP State Limit feature are as follows:

- The Per Interface Mroute State Limit feature allows multiple limits to be configured on an interface, whereas the IGMP State Limit feature allows only one limit to be configured on an interface. The Per Interface Mroute State Limit feature, thus, is more flexible than the IGMP State Limit feature in that it allows multiple limits to be configured for different sets of multicast traffic on an interface.
- The Per Interface Mroute State Limit feature can be used to limit both IGMP and PIM joins, whereas the IGMP State Limit feature can only be used to limit IGMP joins. The IGMP State Limit feature, thus, is more limited in application in that it is best suited to be configured on an edge router to limit the number of groups that receivers can join on an outgoing interface. The Per Interface Mroute State Limit feature has a wider application in that it can be configured to limit IGMP joins on an outgoing interface, to limit PIM joins (for Any Source Multicast [ASM] groups or Source Specific Multicast [SSM] channels) on an outgoing interface connected to other routers, to limit sources behind an incoming interface from sending multicast traffic, or to limit sources directly connected to an incoming interface from sending multicast traffic.

**Note**

Although the PIM Interface Mroute State Limit feature allows you to limit both IGMP and PIM joins, it does not provide the capability to limit PIM or IGMP joins separately because it does not take into account whether the state is created as a result of an IGMP or PIM join. As such, the IGMP State Limit feature is more specific in application because it specifically limits IGMP joins.

- The Per Interface Mroute State Limit feature allows you to specify limits according to the direction of traffic; that is, it allows you to specify limits for outgoing interfaces, incoming interfaces, and for incoming interfaces having directly connected multicast sources. The IGMP State Limit feature, however, only can be used to limit outgoing interfaces. The Per Interface State Mroute State Limit feature, thus, is wider in scope in that it can be used to limit mroute states for both incoming and outgoing interfaces from both sources and receivers, whereas the IGMP State Limit feature is more narrow in scope in that it can only be used to limit mroute states for receivers on an LAN by limiting the number of IGMP joins on an outgoing interface.

Both the IGMP State Limit and Per Interface Mroute State Limit features provide a rudimentary multicast CAC mechanism that can be used to provision bandwidth utilization on an interface when all multicast flows roughly utilize the same amount of bandwidth. The Bandwidth-Based CAC for IP Multicast feature, however, offers a more flexible and powerful alternative for providing multicast CAC in network environments where IP multicast flows utilize different amounts of bandwidth.

**Note**

For more information about the Bandwidth-Based CAC for IP Multicast feature, see the [Bandwidth-Based CAC for IP Multicast](#), page 51.

- [Per Interface Mroute State Limit Feature Design](#), page 49
- [Mechanics of Per Interface Mroute State Limiters](#), page 50
- [Tips for Configuring Per Interface Mroute State Limiters](#), page 51

Per Interface Mroute State Limit Feature Design

The Per Interface Mroute State Limit feature is configured using the **ip multicast limit** command in interface configuration mode. An **ip multicast limit** command configured on an interface is called an per interface mroute state limiter. A per interface mroute state limiter is defined by direction, ACL, and maximum number of mroutes. Each per interface mroute state limiter maintains a counter to ensure that the maximum number of mroutes is not exceeded.

The following forms of the **ip multicast limit** command are available to configure per interface mroute state limiters:

- **ip multicast limit** *access-list max-entries*

This command limits mroute state creation for an ACL-classified set of traffic on an interface when the interface is an outgoing (egress) interface, and limits mroute outgoing interface list (olist) membership when the interface is an incoming (ingress) Reverse Path Forwarding (RPF) interface.

This type of per interface mroute state limiter limits mroute state creation--by accounting each time an mroute permitted by the ACL is created or deleted--and limits mroute olist membership--by accounting each time that an mroute olist member permitted by the ACL is added or removed.

Entering this form of the command (that is, with no optional keywords) is equivalent to specifying the **ip multicast limit rpf** and **ip multicast limit out** forms of the command.

- **ip multicast limit connected** *access-list max-entries*

This command limits mroute state creation for an ACL-classified set of multicast traffic on an incoming (RPF) interface that is directly connected to a multicast source by accounting each time that an mroute permitted by the ACL is created or deleted.

- **ip multicast limit out** *access-list max-entries*

This command limits mroute olist membership on an outgoing interface for an ACL-classified set of multicast traffic by accounting each time that an mroute olist member permitted by the ACL is added or removed.

- **ip multicast limit rpf** *access-list max-entries*

This command limits mroute state creation for an ACL-classified set of multicast traffic on an incoming (RPF) interface by accounting each time an mroute permitted by the ACL is created or deleted.

For the required *access-list* argument, specify the ACL that defines the IP multicast traffic to be limited on an interface. A standard or extended ACL can be specified. Standard ACLs can be used to define the (*, G) state to be limited on an interface. Extended ACLs can be used to define the (S, G) state to be limited on an interface. Extended ACLs also can be used to define the (*, G) state to be limited on an interface, by specifying 0.0.0.0 for the source address and source wildcard--referred to as (0, G)--in the permit or deny statements that compose the extended access list.

Mechanics of Per Interface Mroute State Limiters

The mechanics of per interface mroute state limiters are as follows:

- Each time the state for an mroute is created or deleted and each time an olist member is added or removed, the software searches for a corresponding per interface mroute state limiter that matches the mroute.
- When an mroute is created or deleted, the software searches for a per interface mroute state limiter configured on the incoming (RPF) interface that matches the mroute to be created or deleted. When an olist member is added or removed, the software searches for a per interface mroute state limiter configured on the outgoing interface that matches the mroute to be added or removed.
- A top-down search is performed using the list of configured per interface mroute state limiters. Only per interface mroute state limiters that match the direction of traffic are considered. The first per interface mroute state limiter that matches is used for limiting (sometimes referred to as accounting). A match is found when the ACL permits the mroute state.
- When a match is found, the counter of the per interface mroute state limiter is updated (increased or decreased). If no per interface mroute state limiter is found that matches an mroute, no accounting is performed for the mroute (because there is no counter to update).
- The amount with which to update the counter is called the cost (sometimes referred to as the cost multiplier). The default cost is 1.



Note

A per interface mroute state limiter always allows the deletion of an mroute or the removal of an interface from the olist. In those cases, the respective per interface mroute state limiter decreases the counter by the value of the cost multiplier. In addition, RPF changes to an existing mroute are always allowed (in order to not affect existing traffic). However, a per interface mroute state limiter only allows the creation of an mroute or the addition of an mroute olist member if adding the cost does not exceed the maximum number of mroutes permitted.

Tips for Configuring Per Interface Mroute State Limiters

- To ensure that all mroutes are accounted, you can configure a per interface mroute state limiter whose ACL contains a permit-any statement and set the value of zero (0) for maximum entries. Configuring an mroute state limiter in this manner effectively denies all fall through states, which may be a way to prevent a multicast DoS attack in and out of the interface.
- When creating an ACL, remember that, by default, the end of the ACL contains an implicit deny-any statement for everything if it did not find a match before reaching the end.
- An explicit deny statement for a specific mroute in an ACL can be used to specify the state that will not match the ACL which will prevent the ACL from being accounted. If an mroute matches a deny statement, the search immediately continues to the next configured mroute state limiter. Configuring an explicit deny statement in an ACL can be more efficient than forcing the mroute to fall through an ACL by using an implicit deny-any statement at the end of the ACL.

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 (referred to as *bandwidth-based multicast CAC policies*). This feature can be used to provide bandwidth-based multicast CAC on a per interface basis in network environments where the multicast flows utilize different amounts of bandwidth.

- [Bandwidth-Based CAC for IP Multicast Feature Design, page 51](#)
- [Mechanics of the Bandwidth-Based Multicast CAC Policies, page 51](#)
- [Tips for Configuring Bandwidth-Based CAC Policies for IP Multicast, page 52](#)

Bandwidth-Based CAC for IP Multicast Feature Design

Bandwidth-based multicast CAC policies are configured using the **ip multicast limit cost** command in global configuration mode. The syntax of the **ip multicast limit cost** command is as follows:

```
ip multicast [vrf vrf-name] limit cost access-list cost-multiplier
```

For the required *access-list* argument, specify the ACL that defines the IP multicast traffic for which to apply a cost. A standard or extended ACL can be specified. Standard ACLs can be used to define the (*, G) state. Extended ACLs can be used to define the (S, G) state. Extended ACLs also can be used to define the (*, G) state, by specifying 0.0.0.0 for the source address and source wildcard--referred to as (0, G)--in the permit or deny statements that compose the extended access list.

For the required *cost-multiplier* argument, specify the cost value to be applied to mroutes that match the ACL associated with the bandwidth-based multicast CAC policy. The range is 0 to 2147483647.



Note

In an MVPN environment, the optional **vrf** keyword and *vrf-name* argument are used to specify that the cost be applied only to mroutes associated with MVRP specified for the *vrf-name* argument.

Mechanics of the Bandwidth-Based Multicast CAC Policies

The mechanics of bandwidth-based multicast CAC policies are as follows:

- Once an mroute matches an ACL configured for a per interface mroute state limiter, the Cisco IOS software performs a top-down search from the global or per MVRF list of configured bandwidth-based multicast CAC policies to determine if a cost should be applied to the mroute.
- A cost is applied to the first bandwidth-based CAC policy that matches the mroute. A match is found when the ACL applied to the bandwidth-based CAC policy permits the mroute state.
- The counter for the mroute state limiter either adds or subtracts the cost configured for the *cost-multiplier* argument. If no costs are configured or if the mroute does not match any of the configured bandwidth-based CAC policies, the default cost of 1 is used.

Tips for Configuring Bandwidth-Based CAC Policies for IP Multicast

- To ensure that a particular cost applies to all mroutes being limited, you can configure a bandwidth-based CAC policy whose ACL contains a **permit any** statement. Configuring a bandwidth-based CAC policy in this manner effectively ensures that the default cost is not applied to any mroutes being limited.
- Configuring a bandwidth-based CAC policy with a cost of 0 for the *cost-multiplier* argument can be used to skip the accounting of certain mroutes (for example, to prevent Auto-RP groups or a specific multicast channel from being accounted).
- An explicit deny statement for a specific mroute in an ACL can be used to specify the state that will not match the ACL (thus, preventing the ACL from being accounted). If an mroute matches a deny statement, the search immediately continues to the next configured bandwidth-based CAC policy. Configuring an explicit deny statement in an ACL can be more efficient than forcing the mroute to fall through an ACL (by means of the implicit **deny any** statement at the end of the ACL).

How to Configure Multicast Admission Control

- [Configuring Global and Per MVRF Mroute State Limiters, page 52](#)
- [Configuring MSDP SA Limiters, page 56](#)
- [Configuring IGMP State Limiters, page 58](#)
- [Configuring Per Interface Mroute State Limiters, page 61](#)
- [Configuring Bandwidth-Based Multicast CAC Policies, page 62](#)
- [Monitoring Per Interface Mroute State Limiters and Bandwidth-Based Multicast CAC Policies, page 66](#)

Configuring Global and Per MVRF Mroute State Limiters

Perform the following optional tasks to configure global and per MVRF mroute state limiters.

Global mroute state limiters are used to limit the number of mroutes that can be added to the global table on a router. Configuring a global mroute state limiter can protect a router in the event of a multicast DoS attack (by preventing mroutes from overrunning the router).

Per VRF mroute state limiters are used to limit the number of mroutes that can be added to an MVRF table on an MVPN PE router. Configuring per MVRF mroute state limits can be used to ensure the fair sharing of mroutes between different MVRFs on an MVPN PE router.



Note

Global and per MVRF mroute state limiters operate independently and can be used alone or together, depending upon the admission control requirements of your network.



Note

When configuring global and per VRF mroute state limiters, you can only configure one limit for the global table and one limit per MVRF table.

The following tasks explain how to configure global and per MVRF mroute state limiters:

- [Prerequisites, page 53](#)
- [Configuring a Global Mroute State Limiter, page 53](#)
- [What to Do Next, page 54](#)
- [Configuring Per MVRF Mroute State Limiters, page 54](#)

Prerequisites

- These tasks assume 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.
- Before configuring per MVRF mroute state limiters, the MVRFs on the PE router must be configured using the tasks described in the “ Configuring Multicast VPN ” module.

Configuring a Global Mroute State Limiter

Perform this task to limit the number of mroutes that can be added to the global table. States for mroutes that exceed the global mroute limit will not be created.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast route-limit** *limit* [*threshold*]
4. **end**
5. **show ip mroute count**

DETAILED STEPS

Command or Action	Purpose
<p>Step 1 enable</p> <p>Example:</p> <pre>Router> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.

Command or Action	Purpose
Step 2 <code>configure terminal</code> Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3 <code>ip multicast route-limit limit [threshold]</code> Example: <pre>Router(config)# ip multicast route-limit 1500 1460</pre>	Limits the number of mroutes that can be added to the global table. <ul style="list-style-type: none"> For the required <i>limit</i> argument, specify the limit on the number of mroutes that can be added to the global table. The range is from 1 to 2147483647. Use the optional <i>threshold</i> argument to set an mroute threshold limit. The range is from 1 to 2147483647.
Step 4 <code>end</code> Example: <pre>Router(config)# end</pre>	Ends the current configuration session and returns to privileged EXEC mode.
Step 5 <code>show ip mroute count</code> Example: <pre>Router# show ip mroute count</pre>	(Optional) Displays mroute data and packet count statistics. <ul style="list-style-type: none"> Use this command to verify the number of mroutes in the global table.

What to Do Next

Proceed to the [Configuring Per MVRF Mroute State Limiters, page 54](#) task to configure per MVRF mroute state limiters on a PE router.

Configuring Per MVRF Mroute State Limiters

Perform this optional task to configure per MVRF mroute state limiters to limit the number of mroutes that can be added to a particular MVRF table. This feature can be configured on a PE router to ensure the fair sharing of mroutes between different MVRFs on the router. States for mroutes that exceed the per MVRF mroute limiter are not created.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast vrf *vrf-name* route-limit *limit* [*threshold*]**
4. Repeat Step 3 to configure additional per VRF mroute state limiters for other VRFs on an MVPN PE router.
5. **end**
6. **show ip mroute vrf *vrf-name* count**

DETAILED STEPS

Command or Action	Purpose
Step 1 enable Example: <pre>Router> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2 configure terminal Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3 ip multicast vrf <i>vrf-name</i> route-limit <i>limit</i> [<i>threshold</i>] Example: <pre>Router(config)# ip multicast vrf red route-limit 1500 1460</pre>	Limits the number of mroutes that can be added to a particular MVRF table. <ul style="list-style-type: none"> • For the vrf keyword and <i>vrf-name</i> argument, specify the MVRF for which to apply the limit. • For the required <i>limit</i> argument, specify the limit on the number of mroutes that can be added to the MVRF table (for the specified MVRF). The range is from 1 to 2147483647. • Use the optional <i>threshold</i> argument to set an mroute threshold limit. The range is from 1 to 2147483647
Step 4 Repeat Step 3 to configure additional per VRF mroute state limiters for other VRFs on an MVPN PE router.	--
Step 5 end Example: <pre>Router(config)# end</pre>	Ends the current configuration session and returns to privileged EXEC mode.

Command or Action	Purpose
Step 6 <code>show ip mroute vrf vrf-name count</code> Example: <pre>Router# show ip mroute vrf red count</pre>	(Optional) Displays mroute data and packet count statistics related to the specified MVRF. <ul style="list-style-type: none"> Use this command to verify the number of mroutes in a particular MVRF table.

Configuring MSDP SA Limiters

Perform this optional 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 DoS attacks.



Note

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

This task assumes that you are running MSDP and have configured MSDP peers using the tasks described in the “ Using MSDP to Interconnect Multiple PIM-SM Domains ” module.

SUMMARY STEPS

- enable**
- configure terminal**
- ip msdp [vrf vrf-name] sa-limit {peer-address | peer-name} sa-limit**
- Repeat Step 3 to configure additional per MVRF mroute state limiters for other MVRFs on an MVPN PE router.
- end**
- show ip msdp count**
- show ip msdp peer [peer-address | peer-name]**
- show ip msdp summary**

DETAILED STEPS

Command or Action	Purpose
Step 1 <code>enable</code> Example: <pre>Router> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.

Command or Action	Purpose
<p>Step 2 <code>configure terminal</code></p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 <code>ip msdp [vrf vrf-name] sa-limit {peer-address peer-name} sa-limit</code></p> <p>Example:</p> <pre>Router(config)# ip msdp sa-limit 192.168.10.1 100</pre>	<p>Limits the number of SA messages allowed in the SA cache from the specified MSDP.</p> <ul style="list-style-type: none"> • Use the optional vrf keyword and <i>vrf-name</i> argument to specify the MVRF associated with the MSDP peer. When an MVRF is specified, the MSDP SA limiter is applied to the specified MSDP peer associated with the specified MVRF. • For the required <i>peer-address</i> argument or <i>peer-name</i> argument, specify either the MSDP peer address or MSDP peer name of the peer to be limited. • For the required <i>sa-limit</i> argument, specify the maximum number of SA messages that can be accepted (cached) from the specified peer. The range is from 1 to 2147483646.
<p>Step 4 Repeat Step 3 to configure additional per MVRF mroute state limiters for other MVRFs on an MVPN PE router.</p>	<p>--</p>
<p>Step 5 <code>end</code></p> <p>Example:</p> <pre>Router(config)# end</pre>	<p>Exits global configuration mode and returns to privileged EXEC mode.</p>
<p>Step 6 <code>show ip msdp count</code></p> <p>Example:</p> <pre>Router# show ip msdp count</pre>	<p>(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.</p>
<p>Step 7 <code>show ip msdp peer [peer-address peer-name]</code></p> <p>Example:</p> <pre>Router# show ip msdp peer</pre>	<p>(Optional) Displays detailed information about MSDP peers.</p> <p>Note The output of this command displays the number of SA messages received from MSDP peers that are stored in the cache.</p>
<p>Step 8 <code>show ip msdp summary</code></p> <p>Example:</p> <pre>Router# show ip msdp summary</pre>	<p>(Optional) Displays MSDP peer status.</p> <p>Note The output of this command displays a per-peer “SA Count” field that displays the number of SAs stored in the SA cache.</p>

Configuring IGMP State Limiters



Note

IGMP state limiters impose limits on the number of mroute states resulting from IGMP, IGMP v3lite, and URD membership reports on a global or per interface basis.

- [Prerequisites, page 58](#)
- [Configuring Global IGMP State Limiters, page 58](#)
- [What to Do Next, page 59](#)
- [Configuring Per Interface IGMP State Limiters, page 59](#)

Prerequisites

- These tasks assume 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.
- All ACLs you intend to apply to per interface IGMP state limiters should be configured prior to beginning this configuration task; otherwise, IGMP membership reports for all groups and channels are counted against the configured limits. For information about how to configure ACLs, see the “Creating an IP Access List and Applying It to an Interface” module.

Configuring Global IGMP State Limiters

Perform this optional task to configure one global IGMP state limiter per device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp limit *number***
4. **end**
5. **show ip igmp groups**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.
	Example:	
	Device> enable	<ul style="list-style-type: none"> • Enter your password if prompted.

Command or Action	Purpose
Step 2 <code>configure terminal</code> Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 3 <code>ip igmp limit number</code> Example: Device(config)# <code>ip igmp limit 150</code>	Configures a global limit on the number of mroute states resulting from IGMP membership reports (IGMP joins).
Step 4 <code>end</code> Example: Device(config-if)# <code>end</code>	Ends the current configuration session and returns to privileged EXEC mode.
Step 5 <code>show ip igmp groups</code> Example: Device# <code>show ip igmp groups</code>	(Optional) Displays the multicast groups with receivers that are directly connected to the device and that were learned through IGMP.

What to Do Next

Proceed to the [Configuring Per Interface IGMP State Limiters, page 59](#) task to configure per interface IGMP state limiters.

Configuring Per Interface IGMP State Limiters

Perform this optional task to configure a per interface IGMP state limiter.

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `interface type number`
4. `ip igmp limit number [except access-list]`
5.
 - `exit`
 - `end`
6. `show ip igmp interface [type number]`
7. `show ip igmp groups`

DETAILED STEPS

Command or Action	Purpose
<p>Step 1 <code>enable</code></p> <p>Example:</p> <pre>Device> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> Enter your password if prompted.
<p>Step 2 <code>configure terminal</code></p> <p>Example:</p> <pre>Device# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 <code>interface type number</code></p> <p>Example:</p> <pre>Device(config)# interface GigabitEthernet0/0</pre>	<p>Enters interface configuration mode.</p> <ul style="list-style-type: none"> Specify an interface that is connected to hosts.
<p>Step 4 <code>ip igmp limit number [except access-list]</code></p> <p>Example:</p> <pre>Device(config-if)# ip igmp limit 100</pre>	<p>Configures a per interface limit on the number of mroutes states created as a result of IGMP membership reports (IGMP joins).</p>
<p>Step 5</p> <ul style="list-style-type: none"> <code>exit</code> <code>end</code> <p>Example:</p> <pre>Device(config-if)# exit Device(config-if)# end</pre>	<ul style="list-style-type: none"> (Optional) Ends the current configuration session and returns to global configuration mode. Repeat steps 3 and 4 to configure a per interface limiter on another interface. Ends the current configuration session and returns to privileged EXEC mode.
<p>Step 6 <code>show ip igmp interface [type number]</code></p> <p>Example:</p> <pre>Device# show ip igmp interface</pre>	<p>(Optional) Displays information about the status and configuration of IGMP and multicast routing on interfaces.</p>
<p>Step 7 <code>show ip igmp groups</code></p> <p>Example:</p> <pre>Device# show ip igmp groups</pre>	<p>(Optional) Displays the multicast groups with receivers that are directly connected to the device and that were learned through IGMP.</p>

Configuring Per Interface Mroute State Limiters

Perform this task to prevent DoS attacks or to provide a multicast CAC mechanism for controlling bandwidth when all multicast flows utilize approximately the same amount of bandwidth.

All ACLs to be applied to per interface mroute state limiters must be configured prior to beginning this configuration task; otherwise, the limiters are ignored. For information about how to configure ACLs, see the “Creating an IP Access List and Applying It to an Interface” module of the *Security Configuration Guide: Access Control Lists* guide.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip multicast limit** [**connected** | **out** | **rpf**] *access-list max-entries*
5. Repeat Step 4 to configure additional per interface mroute state limiters on this interface.
6. Repeat Steps 3 and Step 4 to configure per interface mroute state limiters on additional interfaces.
7. **end**

DETAILED STEPS

Command or Action	Purpose
<p>Step 1 enable</p> <p>Example:</p> <pre>Device> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
<p>Step 2 configure terminal</p> <p>Example:</p> <pre>Device# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 interface <i>type number</i></p> <p>Example:</p> <pre>Device(config)# interface GigabitEthernet0/0</pre>	<p>Enters interface configuration mode for the specified interface type and number.</p>
<p>Step 4 ip multicast limit [connected out rpf] <i>access-list max-entries</i></p> <p>Example:</p> <pre>Device(config-if)# ip multicast limit 15 100</pre>	<p>Configures per interface mroute state limiters.</p>

Command or Action	Purpose
Step 5 Repeat Step 4 to configure additional per interface mroute state limiters on this interface.	--
Step 6 Repeat Steps 3 and Step 4 to configure per interface mroute state limiters on additional interfaces.	--
Step 7 end	Returns to privileged EXEC mode.
Example:	
Device(config-if)# end	

- [What to Do Next, page 62](#)

What to Do Next

Proceed to the Monitoring Per Interface Mroute State Limiters and Bandwidth-Based Multicast CAC Policies task to monitor per interface mroute state limiters.

Configuring Bandwidth-Based Multicast CAC Policies

Perform this optional task to configure bandwidth-based multicast CAC policies. Bandwidth-based multicast CAC policies provide the capability to assign costs to mroutes that are being limited by per interface mroute state limiters. This task can be used to provide bandwidth-based multicast CAC on a per interface basis in network environments where the multicast flows utilize different amounts of bandwidth. Bandwidth-based multicast CAC policies can be applied globally or per MVRF.

- 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.
- All ACLs you intend to apply to bandwidth-based multicast CAC policies should be configured prior to beginning this configuration task; otherwise, the limiters are ignored. For information about how to configure ACLs, see the “ Creating an IP Access List and Applying It to an Interface ” module.



Note

You can omit Steps 3 to 7 if you have already configured the per interface mroute state limiters for which to apply costs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip multicast limit** [**connected** | **out** | **rpf**] *access-list max-entries*
- 5.
- 6.
7. Repeat Step 4 if you want to configure additional mroute state limiters on the interface.
8. Repeat Step 3 and Step 4 if you want to configure mroute state limiters on additional interfaces.
9. **exit**
10. **ip multicast** [**vrf** *vrf-name*] **limit cost** *access-list cost-multiplier*
11. Repeat Step 8 if you want to apply additional costs to mroutes.
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>enable</p> <p>Example:</p> <pre>Router> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
Step 3	<p>interface <i>type number</i></p> <p>Example:</p> <pre>Router(config)# interface GigabitEthernet 0/0</pre>	<p>Enters interface configuration mode for the specified interface type and number.</p>

Command or Action	Purpose
<p>Step 4 <code>ip multicast limit [connected out rpf] access-list max-entries</code></p> <p>Example:</p> <pre>Router(config-if)# ip multicast limit acl-test 100</pre> <p>Example:</p>	<p>Configures mroute state limiters on an interface.</p> <ul style="list-style-type: none"> • Specify the ip multicast limit command with no optional keywords to limit mroute state creation for an ACL-classified set of traffic on an interface when the interface is an outgoing (egress) interface, and to limit mroute olist membership when the interface is an incoming (ingress) Reverse Path Forwarding (RPF) interface.
<p>Step 5</p>	<ul style="list-style-type: none"> • <ul style="list-style-type: none"> ◦ This type of mroute state limiter limits mroute state creation by accounting each time an mroute permitted by the ACL is created or deleted and limits mroute olist membership by accounting each time that an mroute olist member permitted by the ACL is added or removed. ◦ Entering this form of the command (that is, with no optional keywords) is equivalent to specifying the ip multicast limit rpf and ip multicast limit out forms of the command.
<p>Step 6</p>	<ul style="list-style-type: none"> • Use the optional connected keyword to configure an mroute state limiter that limits mroute states created for an ACL-classified set of multicast traffic on an incoming (RPF) interface that is directly connected to a multicast source by accounting each time that an mroute permitted by the ACL is created or deleted. • Use the optional out keyword to configure an mroute state limiter that limits mroute olist membership on an outgoing interface for an ACL-classified set of multicast traffic by accounting each time that an mroute olist member permitted by the ACL is added or removed. • Use the optional rpf keyword to configure an mroute state limiter that limits the number of mroute states created for an ACL-classified set of multicast traffic on an incoming (RPF) interface by accounting each time an mroute permitted by the ACL is created or deleted. • For the required <i>access-list</i> argument, specify the ACL that defines the IP multicast traffic to be limited on an interface. <ul style="list-style-type: none"> ◦ Standard ACLs can be used to define the (*, G) state to be limited on an interface. ◦ Extended ACLs can be used to define the (S, G) state to be limited on an interface. Extended ACLs also can be used to define the (*, G) state to be limited on an interface, by specifying 0.0.0.0 for the source address and source wildcard--referred to as (0, G)--in the permit or deny statements that compose the extended access list. • For the required <i>max-entries</i> argument, specify the maximum number of mroutes permitted by the per interface mroute state limiter. The range is from 0 to 2147483647.

Command or Action	Purpose
Step 7 Repeat Step 4 if you want to configure additional mroute state limiters on the interface.	--
Step 8 Repeat Step 3 and Step 4 if you want to configure mroute state limiters on additional interfaces.	--
Step 9 exit Example: Router(config-if)# exit	Exits interface configuration mode, and returns to global configuration mode.
Step 10 ip multicast [vrf vrf-name] limit cost access-list cost-multiplier Example: Router(config)# ip multicast limit cost acl-MP2SD-channels 4000	Applies costs to per interface mroute state limiters. <ul style="list-style-type: none"> • Use the optional vrf keyword and <i>vrf-name</i> argument to specify that the cost be applied only to mroutes associated with MVRF specified for the <i>vrf-name</i> argument. • For the required <i>access-list</i> argument, specify the ACL that defines the IP multicast traffic for which to apply a cost. <ul style="list-style-type: none"> ◦ Standard ACLs can be used to define the (*, G) state. ◦ Extended ACLs can be used to define the (S, G) state. Extended ACLs also can be used to define the (*, G) state, by specifying 0.0.0.0 for the source address and source wildcard--referred to as (0, G)--in the permit or deny statements that compose the extended access list. • For the required <i>cost-multiplier</i> argument, specify the cost value to be applied to mroutes that match the ACL associated with the bandwidth-based multicast CAC policy. The range is 0 to 2147483647.
Step 11 Repeat Step 8 if you want to apply additional costs to mroutes.	--
Step 12 end Example: Router(config-if)# end	Exits interface configuration mode, and enters privileged EXEC mode.

- [What to Do Next, page 65](#)

What to Do Next

Proceed to the Monitoring Per Interface Mroute State Limiters and Bandwidth-Based Multicast CAC Policies task to monitor bandwidth-based multicast CAC policies.

Monitoring Per Interface Mroute State Limiters and Bandwidth-Based Multicast CAC Policies

Perform this optional task to monitor per interface mroute state limiters and bandwidth-based multicast CAC policies.

SUMMARY STEPS

1. **enable**
2. **debug ip mrouting limits** [*group-address*]
3. **show ip multicast limit** *type number*
4. **clear ip multicast limit** [*type number*]

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **debug ip mrouting limits** [*group-address*]

Displays debugging information about configured per interface mroute state limiters and bandwidth-based multicast CAC policies.

The following output is from the **debug ip mrouting limits** command. The output displays the following events:

- An mroute state being created and the corresponding per interface mroute state limiter counter being increased by the default cost of 1 on incoming Ethernet interface 1/0.
- An mroute olist member being removed from the olist and the corresponding per interface mroute limiter being decreased by the default cost of 1 on outgoing Ethernet interface 1/0.
- An mroute being denied by the per interface mroute state limiter because the maximum number of mroute states has been reached.
- An mroute state being created and the corresponding per interface mroute state limiter counter being increased by the cost of 2 on incoming Ethernet interface 1/0.
- An mroute olist member being removed from the olist and the corresponding per interface mroute limiter being decreased by a cost of 2 on outgoing Ethernet interface 1/0.

Example:

```
device# debug ip mrouting limits
```

```
MRL(0): incr-ed acl 'rpf-list' to (13 < max 32), [n:0,p:0], (main) GigabitEthernet0/0,
(10.41.0.41, 225.30.200.60)
MRL(0): decr-ed acl 'out-list' to (10 < max 32), [n:0,p:0], (main) GigabitEthernet0/0, (*,
225.40.202.60)
MRL(0): Add mroute (10.43.0.43, 225.30.200.60) denied for GigabitEthernet0/2, acl std-list, (16 =
max 16)
```

```
MRL(0): incr-ed limit-acl `rpf-list' to (12 < max 32), cost-acl 'cost-list' cost 2, [n:0,p:0],
(main) GigabitEthernet0/0, (10.41.0.41, 225.30.200.60)
MRL(0): decr-ed limit-acl `out-list' to (8 < max 32), cost-acl 'cost-list' cost 2, [n:0,p:0],
(main) GigabitEthernet0/0, (*, 225.40.202.60)
```

Step 3

show ip multicast limit *type number*

Displays counters related to mroute state limiters configured on the interfaces on the router.

For each per interface mroute state limiter shown in the output, the following information is displayed:

- The direction of traffic that the per mroute state limiter is limiting.
- The ACL referenced by the per interface mroute state limiter that defines the IP multicast traffic being limited.
- Statistics, enclosed in parenthesis, which track the current number of mroutes being limited less the configured limit. Each time the state for an mroute is created or deleted and each time an outgoing interface list (olist) member is added or removed, the counters for matching per interface mroute state limiters are increased or decreased accordingly.
- The exceeded counter, which tracks the total number of times that the limit configured for the per interface mroute state limiter has been exceeded. Each time an mroute is denied due to the configured limit being reached, the exceeded counter is increased by a value of 1.

The following is sample output from the **show ip multicast limit** command with the *type number* arguments. In this example, information about mroute state limiters configured on Gigabit Ethernet interface 0/0 is displayed.

Example:

```
Device# show ip multicast limit GigabitEthernet 0/0

Interface GigabitEthernet 0/0
  Multicast Access Limits
  out acl out-list (1 < max 32) exceeded 0
  rpf acl rpf-list (6 < max 32) exceeded 0
  con acl conn-list (0 < max 32) exceeded 0
```

Step 4

clear ip multicast limit [*type number*]

Resets the exceeded counter for per interface mroute state limiters.

The following example shows how to reset exceeded counters for per interface mroute state limiters configured on Gigabit Ethernet interface 0/0:

Example:

```
Device# clear ip multicast limit interface GigabitEthernet 0/0
```

Configuration Examples for Configuring Multicast Admission Control

- [Configuring Global and Per MVRP Mroute State Limiters Example, page 68](#)
- [Configuring MSDP SA Limiters Example, page 68](#)
- [Example: Configuring IGMP State Limiters, page 68](#)
- [Example Configuring Per Interface Mroute State Limiters, page 69](#)

In this example, a service provider is offering 300 Standard Definition (SD) TV channels. Each SD channel utilizes approximately 4 Mbps.

The service provider must provision the Gigabit Ethernet interfaces on the PE device connected to the Digital Subscriber Line Access Multiplexers (DSLAMs) as follows: 50% of the link's bandwidth (500 Mbps) must be available to subscribers of the Internet, voice, and video on demand (VoD) service offerings while the remaining 50% (500 Mbps) of the link's bandwidth must be available to subscribers of the SD channel offerings.

Because each SD channel utilizes the same amount of bandwidth (4 Mbps), per interface IGMP state limiters can be used to provide the necessary CAC to provision the services being offered by the service provider. To determine the required CAC needed per interface, the total number of channels is divided by 4 (because each channel utilizes 4 Mbps of bandwidth). The required CAC needed per interface, therefore, is as follows:

$$500\text{Mbps} / 4\text{Mbps} = 125 \text{ mroutes}$$

Once the required CAC is determined, the service provider uses the results to configure the per IGMP state limiters required to provision the Gigabit Ethernet interfaces on the PE device. Based on the network's CAC requirements, the service provider must limit the SD channels that can be transmitted out a Gigabit Ethernet interface (at any given time) to 125. Configuring a per interface IGMP state limit of 125 for the SD channels provisions the interface for 500 Mbps of bandwidth, the 50% of the link's bandwidth that must always be available (but never exceeded) for the SD channel offerings.

The following configuration shows how the service provider uses a per interface mroute state limiter to provision interface Gigabit Ethernet 0/0 for the SD channels and Internet, Voice, and VoD services being offered to subscribers:

```
interface GigabitEthernet0/0
description --- Interface towards the DSLAM ---
.
.
.
ip igmp limit 125
```

Example Configuring Per Interface Mroute State Limiters

The following example shows how to configure per interface mroute state limiters to provide multicast CAC in a network environment where all the multicast flows roughly utilize the same amount of bandwidth.

an mroute state limit of 75 for the SD channels offered in the Basic Service bundle provisions the interface for 300 Mbps of bandwidth (the 60% of the link's bandwidth that must always be available to [but never exceeded by] the subscribers of the Basic Services bundle).

- For the Premium Services bundle, the service provider must limit the number of Premium Service SD channels that can be transmitted out a Gigabit Ethernet interface (at any given time) to 25. Configuring an mroute state limit of 25 for the SD channels offered in the Premium Service bundle provisions the interface for 100 Mbps of bandwidth (the 20% of the link's bandwidth that must always be available to [but never exceeded by] the subscribers of the Premium Service bundle).
- For the Gold Services bundle, the service provider must limit the number of Gold Service SD channels that can be transmitted out a Gigabit Ethernet interface (at any given time) to 25. Configuring an mroute state limit of 25 for the SD channels offered in the Gold Service bundle provisions the interface for 100 Mbps of bandwidth (the 20% of the link's bandwidth that must always be available to [but never exceeded by] the subscribers of the Gold Service bundle).

The service provider then configures three ACLs to be applied to per interface mroute state limiters. Each ACL defines the SD channels for each SD channel bundle to be limited on an interface:

- acl-basic--The ACL that defines the SD channels offered in the basic service.
- acl-premium--The ACL that defines the SD channels offered in the premium service.
- acl-gold--The ACL that defines the SD channels offered in the gold service.

These ACLs are then applied to per interface mroute state limiters configured on the PE device's Gigabit Ethernet interfaces.

For this example, three per interface mroute state limiters are configured on Gigabit Ethernet interface 0/0 to provide the multicast CAC needed to provision the interface for the SD channel bundles being offered to subscribers:

- An mroute state limit of 75 for the SD channels that match acl-basic.
- An mroute state limit of 25 for the SD channels that match acl-premium.
- An mroute state limit of 25 for the SD channels that match acl-gold.

The following configuration shows how the service provider uses per interface mroute state limiters to provision Gigabit Ethernet interface 0/0 for the SD channel bundles and Internet, Voice, and VoD services being offered to subscribers:

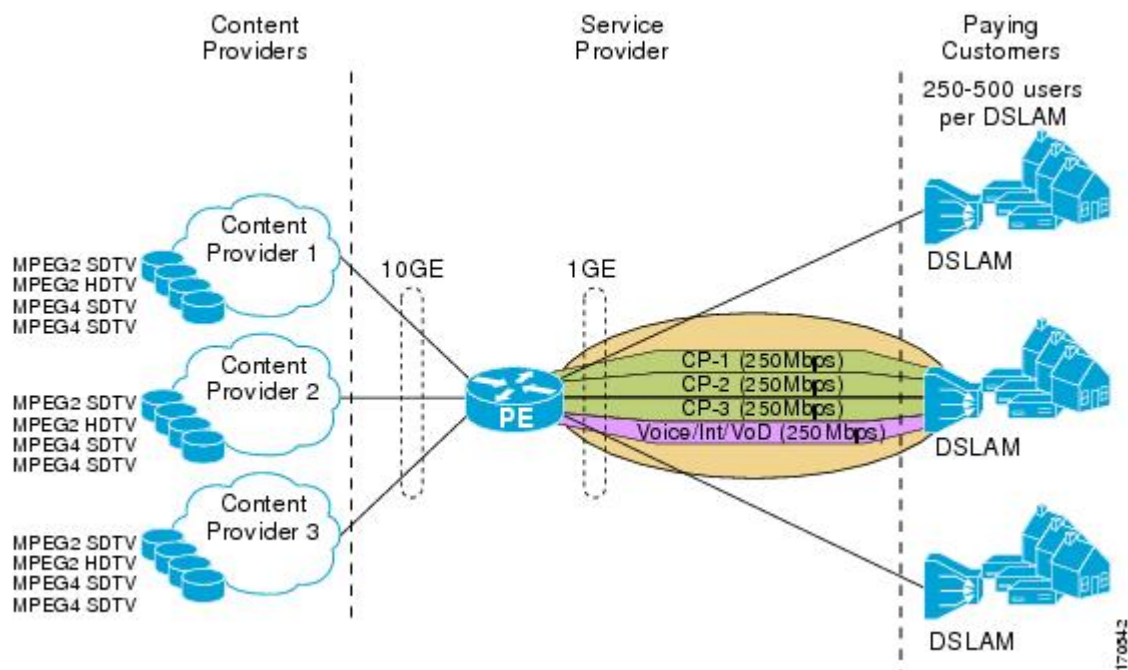
```
interface GigabitEthernet0/0
description --- Interface towards the DSLAM ---
.
.
.
ip multicast limit out acl-basic 75
ip multicast limit out acl-premium 25
ip multicast limit out acl-gold 25
```

Example: Configuring Bandwidth-Based Multicast CAC Policies

The following example shows how to configure bandwidth-based multicast CAC policies to provide multicast CAC in a network environment where the multicast flows utilize different amounts of bandwidth.

This example uses the topology illustrated in the figure.

Figure 8 Bandwidth-Based CAC for IP Multicast Example Topology



In this example, three content providers are providing TV services across a service provider core. The content providers are broadcasting TV channels that utilize different amounts of bandwidth:

- MPEG-2 SDTV channels--4 Mbps per channel.
- MPEG-2 HDTV channels--18 Mbps per channel.
- MPEG-4 SDTV channels--1.6 Mbps per channel.
- MPEG-4 HDTV channels--6 Mbps per channel.

The service provider needs to provision the fair sharing of bandwidth between these three content providers to its subscribers across Gigabit Ethernet interfaces. The service provider, thus, determines that it needs to provision each Gigabit Ethernet interface on the PE device connected to the DSLAMs as follows:

- 250 Mbps per content provider.
- 250 Mbps for Internet, voice, and VoD services.

The service provider then configures three ACLs:

- acl-CP1-channels--The ACL that defines the channels being offered by the content provider CP1.
- acl-CP2-channels--The ACL that defines the channels being offered by the content provider CP2.
- acl-CP3-channels--The ACL that defines the channels being offered by the content provider CP3.

Because the content providers are broadcasting TV channels that utilize different amounts of bandwidth, the service provider needs to determine the values that need to be configured for the per interface mroute state limiters and bandwidth-based multicast CAC policies to provide the fair sharing of bandwidth required between the content providers.

Prior to the introduction of the Bandwidth-Based CAC for IP Multicast feature, per interface mroute state limiters were based strictly on the number of flows. The introduction of cost multipliers by the Bandwidth-Based CAC for IP Multicast feature expands how per interface mroute state limiters can be defined. Instead

of defining the per interface mroute state limiters based on the number of multicast flows, the service provider looks for a common unit of measure and decides to represent the per interface mroute state limiters in kilobits per second (Kbps). The service provider then configures three per interface mroute state limiters, one per content provider. Because the link is a Gigabit, the service provider sets each limit to 250000 (because 250000 Kbps equals 250 Mbps, the number of bits that service provider needs to provision per content provider).

The service provider needs to further provision the fair sharing of bandwidth between the content providers, which can be achieved by configuring bandwidth-based multicast CAC policies. The service provider decides to create four bandwidth-based CAC policies, one policy per channel based on bandwidth. For these policies, the service provider configures the following ACLs:

- acl-MP2SD-channels--Defines all the MPEG-2 SD channels offered by the three content providers.
- acl-MP2HD-channels--Defines all the MPEG-2 HD channels offered by the three content providers.
- acl-MP4SD-channels--Defines all the MPEG-4 SD channels offered by the three content providers.
- acl-MP4HD-channels--Defines all the MPEG-4 HD channels offered by the three content providers.

For each policy, a cost multiplier (represented in Kbps) is defined for each ACL that is based on the bandwidth of the channels defined in the ACL:

- 4000--Represents the 4 Mbps MPEG-2 SD channels.
- 18000--Represents the 18 Mbps MPEG-2 HD channels.
- 1600--Represents the 1.6 Mbps MPEG-4 SD channels.
- 6000--Represents the 6 Mbps MPEG-4 HD channels.

The following configuration example shows how the service provider used per interface mroute state limiters with bandwidth-based multicast CAC policies to provision Gigabit Ethernet interface 0/0 for the fair sharing of bandwidth required between the three content providers:

```
!
ip multicast limit cost acl-MP2SD-channels 4000
ip multicast limit cost acl-MP2HD-channels 18000
ip multicast limit cost acl-MP4SD-channels 1600
ip multicast limit cost acl-MP4HD-channels 6000
!
.
.
.
!
interface GigabitEthernet0/0
 ip multicast limit out acl-CP1-channels 250000
 ip multicast limit out acl-CP2-channels 250000
 ip multicast limit out acl-CP3-channels 250000
!
```

Additional References

The following sections provide references related to configuring multicast admission control.

Related Documents

Related Topic	Document Title
Overview of the IP multicast technology area	“ IP Multicast Technology Overview ” module

Related Topic	Document Title
Concepts, tasks, and examples for configuring an IP multicast network using PIM	“ Configuring a Basic IP Multicast Network ” module
Concepts, tasks, and examples for using MSDP to interconnection multiple PIM-SM domains	“ Using MSDP to Interconnect Multiple PIM-SM Domains ” module
Multicast commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	<i>Cisco IOS IP Multicast Command Reference</i>

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.	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, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>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.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	<p>http://www.cisco.com/techsupport</p>

Feature Information for Configuring Multicast Admission Control

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

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

Table 4 Feature Information for Configuring Multicast Admission Control

Feature Name	Releases	Feature Information
Bandwidth-Based CAC for IP Multicast	12.2(33)SRB 12.4(15)T 12.2(33)SXI	<p>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.</p> <p>The following command was introduced by this feature: ip multicast limit cost.</p>

Feature Name	Releases	Feature Information
IGMP State Limit	12.2(14)S 12.2(15)T 15.0(1)S	<p>The IGMP State Limit feature introduces the capability to limit the number of mroute states resulting from IGMP membership states on a per interface or global basis. Membership reports exceeding the configured limits are not entered into the IGMP cache. 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.</p> <p>The following commands were introduced or modified by this feature: ip igmp limit(global), ip igmp limit(interface), show ip igmp interface.</p>
Per Interface Mroute State Limit	12.3(14)T 12.2(33)SRB 12.2(33)SXI	<p>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.</p> <p>The following commands were introduced or modified by this feature: clear ip multicast limit, debug ip mrouting limits, ip multicast limit, show ip multicast limit.</p>

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams,

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



IPv6 Multicast: Bandwidth-Based Call Admission Control

- [Finding Feature Information, page 79](#)
- [Information About IPv6 Multicast: Bandwidth-Based Call Admission Control, page 79](#)
- [How to Implement IPv6 Multicast Bandwidth-Based Call Admission Control, page 80](#)
- [Configuration Examples for IPv6 Multicast Bandwidth-Based Call Admission Control, page 83](#)
- [Additional References, page 83](#)
- [Feature Information for IPv6 Multicast: Bandwidth-Based Call Admission Control, page 84](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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

Information About IPv6 Multicast: Bandwidth-Based Call Admission Control

- [Bandwidth-Based CAC for IPv6 Multicast, page 79](#)

Bandwidth-Based CAC for IPv6 Multicast

The bandwidth-based call admission control (CAC) for IPv6 multicast feature implements 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 use different amounts of bandwidth.

This feature limits and accounts for IPv6 multicast state in detail. When this feature is configured, interfaces can be limited to the number of times they may be used as incoming or outgoing interfaces in the IPv6 multicast PIM topology.

With this feature, device administrators can configure global limit cost commands for state matching access lists and specify which cost multiplier to use when accounting such state against the interface limits. This

feature provides the required flexibility to implement bandwidth-based local CAC policy by tuning appropriate cost multipliers for different bandwidth requirements.

How to Implement IPv6 Multicast Bandwidth-Based Call Admission Control

- [Configuring the Global Limit for Bandwidth-Based CAC in IPv6, page 80](#)
- [Configuring an Access List for Bandwidth-Based CAC in IPv6, page 81](#)
- [Configuring the Interface Limit for Bandwidth-Based CAC in IPv6, page 82](#)

Configuring the Global Limit for Bandwidth-Based CAC in IPv6

Device administrators can configure global limit cost commands for state matching access lists.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 multicast [vrf vrf-name] limit cost access-list cost-multiplier**

DETAILED STEPS

Command or Action	Purpose
Step 1 enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2 configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3 ipv6 multicast [vrf vrf-name] limit cost access-list cost-multiplier Example: Device (config)# ipv6 multicast limit cost costlist1 2	Applies a cost to mroutes that match per-interface mroute state limiters in IPv6.

Configuring an Access List for Bandwidth-Based CAC in IPv6

In bandwidth-based CAC for IPv6, device administrators can configure global limit cost commands for state matching access lists. Perform this task to configure an access list to configure a state matching access list.

or

deny

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 access-list** *access-list-name*
4.
 - **permit**
 - **deny**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ipv6 access-list <i>access-list-name</i> Example: Device(config)# ipv6 access-list costlist1	Defines an IPv6 access list and places the device in IPv6 access list configuration mode.
Step 4	<ul style="list-style-type: none"> • permit • deny Example: Device(config)# permit any ff03::1/64 Device(config)# deny any ff03::1/64	Sets conditions for an IPv6 access list.

Configuring the Interface Limit for Bandwidth-Based CAC in IPv6

Bandwidth-based CAC for IPv6 counts per-interface IPv6 mroute states using cost multipliers. With this feature, device administrators can specify which cost multiplier to use when accounting such state against the interface limits.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 address** {*ipv6-address / prefix-length* | *prefix-name sub-bits / prefix-length*}
5. **ipv6 multicast limit** [**connected** | **rpf** | **out**] *limit-acl max*

DETAILED STEPS

Command or Action	Purpose
Step 1 enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2 configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3 interface <i>type number</i> Example: Device(config)# interface FastEthernet 1/3	Specifies an interface type and number, and places the device in interface configuration mode.
Step 4 ipv6 address { <i>ipv6-address / prefix-length</i> <i>prefix-name sub-bits / prefix-length</i> } Example: Device(config-if)# ipv6 address FE80::40:1:3 link-local	Configures an IPv6 address based on an IPv6 general prefix.
Step 5 ipv6 multicast limit [connected rpf out] <i>limit-acl max</i> Example: Device (config-if)# ipv6 multicast limit out acl1 10	Configures per-interface mroute state limiters in IPv6.

Configuration Examples for IPv6 Multicast Bandwidth-Based Call Admission Control

- [Example: Configuring the Global Limit for Bandwidth-Based CAC](#), page 83
- [Example: Configuring an Access List for Bandwidth-Based CAC in IPv6](#), page 83
- [Example: Configuring the Interface Limit for Bandwidth-Based CAC in IPv6](#), page 83

Example: Configuring the Global Limit for Bandwidth-Based CAC

The following example configures the global limit on the source device.

```
ipv6 multicast limit cost cost-list 2
```

Example: Configuring an Access List for Bandwidth-Based CAC in IPv6

The following example shows how to configure an access list to use for bandwidth-based CAC:

```
ipv6 access-list cost-list
 permit any ff03::1/64
```

Example: Configuring the Interface Limit for Bandwidth-Based CAC in IPv6

The following example configures the interface limit on the source device's outgoing interface Ethernet 1/3.

```
interface Ethernet1/3
 ipv6 address FE80::40:1:3 link-local
 ipv6 address 2001:DB8:1:1:3/64
 ipv6 multicast limit out acl1 10
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	<i>IPv6 Configuration Guide</i>
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	<i>Cisco IOS IPv6 Command Reference</i>
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	<i>IPv6 RFCs</i>

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
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Multicast: Bandwidth-Based Call Admission Control

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

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

Table 5 **Feature Information for IPv6 Multicast: Bandwidth-Based Call Admission Control**

Feature Name	Releases	Feature Information
IPv6 Multicast: Bandwidth-Based Call Admission Control	12.2(40)SG 3.2.0SG 15.0(2)SG 12.2(33)SRE Cisco IOS XE Release 2.6 15.0(1)S	This feature can be used to provide bandwidth-based CAC on a per-interface basis in network environments where the multicast flows use different amounts of bandwidth. The following commands were introduced or modified: ipv6 access-list , ipv6 address , ipv6 multicast limit , ipv6 multicast limit cost .
mCAC enhancement: configurable treshold notification for mCAC limits	Cisco IOS XE Release 2.6	This feature enables system notifications when actual simultaneous multicast channel numbers exceeds or fall below some percentage (called threshold percantage). The following command were introduced or modified by this feature: ipv6 multicast limit , ipv6 multicast limit rate .

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

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



PIM Dense Mode State Refresh

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.

- [Finding Feature Information, page 87](#)
- [Prerequisite for PIM Dense Mode State Refresh, page 87](#)
- [Restrictions on PIM Dense Mode State Refresh, page 87](#)
- [Information About PIM Dense Mode State Refresh, page 88](#)
- [How to Configure PIM Dense Mode State Refresh, page 88](#)
- [Configuration Examples for PIM Dense Mode State Refresh, page 90](#)
- [Additional References, page 91](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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

Prerequisite for PIM Dense Mode State Refresh

- PIM dense mode is enabled on an interface. By default, all PIM devices that are running a Cisco software release that supports the PIM Dense Mode State Refresh feature automatically process and forward state refresh control messages.

Restrictions on PIM Dense Mode State Refresh

- All devices in a PIM dense mode network must run a software release 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 devices on the same LAN. Specifically, the same origination interval must be configured on each device interface that is directly connected to the LAN.

Information About PIM Dense Mode State Refresh

- [PIM Dense Mode State Refresh Overview, page 88](#)
- [Benefits of PIM Dense Mode State Refresh, page 88](#)

PIM Dense Mode State Refresh Overview

The PIM Dense Mode State Refresh feature is an extension of the PIM Version 2 multicast routing architecture.

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 devices 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.

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 device in the distribution tree.

Benefits of PIM Dense Mode State Refresh

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 devices 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.

How to Configure PIM Dense Mode State Refresh

- [Configuring PIM Dense Mode State Refresh, page 88](#)
- [Verifying PIM Dense Mode State Refresh Configuration, page 89](#)
- [Monitoring and Maintaining PIM DM State Refresh, page 89](#)

Configuring PIM Dense Mode State Refresh

There are no configuration tasks for enabling the PIM Dense Mode State Refresh feature. By default, all PIM devices that are running a Cisco 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 device, use the **ip pim state-refresh disable** global configuration command. To enable state refresh again if it has been disabled, use the **no 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 device, use the following commands beginning in global configuration mode:

Command	Purpose
Device(config)# interface <i>type number</i>	Specifies an interface and places the device in interface configuration mode.
Device(config-if)# ip pim state-refresh origination-interval [<i>interval</i>]	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 <i>interval</i> argument. The default interval is 60 seconds. The interval range is 1 second 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 sample output indicates that processing, forwarding, and origination of state refresh control messages is enabled.

```
Device# show ip pim interface fastethernet 0/1/0 detail
FastEthernet0/1/0 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.

```
Device# show ip pim neighbor
PIM Neighbor Table
Neighbor          Interface          Uptime/Expires    Ver  DR
Address
172.16.5.1        Ethernet1/1        00:09:03/00:01:41 v2    1 / B S
```

Monitoring and Maintaining PIM DM State Refresh

The following sample output from the **debug ip pim** command includes PIM Dense Mode State Refresh control messages that are sent and received by a specified PIM device:

```
Device# 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 GigabitEthernet1/1/0 for (172.16.8.3,239.0.0.1)
TTL=9
```

The following output from the **show ip mroute** command displays the resulting prune timer changes for GigabitEthernet interface1/0/0 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 device.) 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 devices were discovered. In the second output from the **show ip mroute** command, the prune timer has been reset to 00:02:55.

```
Device# 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:GigabitEthernet1/1/0, RPF nbr 172.16.5.2
  Outgoing interface list:
    GigabitEthernet1/0/0, Prune/Dense, 00:09:43/00:02:06
Device#
*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
Device# 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:GigabitEthernet1/1/0, RPF nbr 172.16.5.2
  Outgoing interface list:
    GigabitEthernet1/0/0, Prune/Dense, 00:09:55/00:02:55
```

Configuration Examples for PIM Dense Mode State Refresh

- [Example: Originating Processing and Forwarding PIM Dense Mode State Refresh Control Messages, page 90](#)
- [Example: Processing and Forwarding PIM Dense Mode State Refresh Control Messages, page 90](#)

Example: Originating Processing and Forwarding PIM Dense Mode State Refresh Control Messages

The following example is for a PIM device that is originating, processing, and forwarding PIM Dense Mode State Refresh control messages on Fast Ethernet interface 0/1/0 every 60 seconds:

```
ip multicast-routing distributed
interface FastEthernet0/1/0
 ip address 172.16.8.1 255.255.255.0
 ip pim state-refresh origination-interval 60
 ip pim dense-mode
```

Example: Processing and Forwarding PIM Dense Mode State Refresh Control Messages

The following example is for a PIM device that is just processing and forwarding PIM Dense Mode State Refresh control messages on Fast Ethernet interface 1/1/0:

```
ip multicast-routing
interface FastEthernet1/1/0
 ip address 172.16.7.3 255.255.255.0
 ip pim dense-mode
```

Additional References

Related Documents

Related Topic	Document Title
The PIM Dense Mode State Refresh feature is an extension of the PIM Version 2 multicast routing architecture	“Configuring Basic IP Multicast” module
IP multicast commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	<i>Cisco IOS IP Multicast Command Reference</i>

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
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>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.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	<p>http://www.cisco.com/techsupport</p>

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

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