



IP Multicast Routing Configuration Guide, Cisco IOS XE Bengaluru 17.5.x (Catalyst 9400 Switches)

First Published: 2021-04-01

Last Modified: 2023-06-30

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 NON-INFRINGEMENT 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.

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.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

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. (1721R)

© 2021 Cisco Systems, Inc. All rights reserved.



CONTENTS

CHAPTER 1

IP Multicast Routing Technology Overview 1

Information About IP Multicast Technology 1

About IP Multicast 1

Role of IP Multicast in Information Delivery 2

IP Multicast Routing Protocols 2

Internet Group Management Protocol 3

Protocol-Independent Multicast 3

Rendezvous Point 3

IGMP Snooping 3

IP Multicast Tables 4

Hardware and Software Forwarding 5

Partial Routes 6

Software Routes 6

Non-Reverse Path Forwarding Traffic 6

Multicast Group Transmission Scheme 7

IP Multicast Boundary 9

IP Multicast Group Addressing 10

IP Class D Addresses 10

IP Multicast Address Scoping 10

Layer 2 Multicast Addresses 12

Cisco Express Forwarding, MFIB, and Layer 2 Forwarding 12

IP Multicast Delivery Modes 14

Source Specific Multicast 14

Multicast Fast Drop 14

Multicast Forwarding Information Base 15

S/M, 224/4 16

Multicast High Availability	16
Additional References for IP Multicast	16

CHAPTER 2

Configuring Basic IP Multicast Routing	17
Restrictions for Basic IP Multicast Routing	17
Information About Basic IP Multicast Routing	17
Multicast Forwarding Information Base Overview	18
Default IP Multicast Routing Configuration	18
How to Configure Basic IP Multicast Routing	19
Configuring Basic IP Multicast Routing	19
Configuring IP Multicast Forwarding	21
Configuring a Static Multicast Route (mroute)	22
Configuring Optional IP Multicast Routing Features	23
Defining the IP Multicast Boundary	23
Configuring sdr Listener Support	25
Monitoring and Maintaining Basic IP Multicast Routing	27
Clearing Caches, Tables, and Databases	27
Displaying System and Network Statistics	27
Configuration Examples for Basic IP Multicast Routing	29
Example: Configuring an IP Multicast Boundary	29
Example: Responding to mrinfo Requests	29
Additional References for Basic IP Multicast Routing	30
Feature History for Basic IP Multicast Routing	30

CHAPTER 3

Configuring Multicast Routing over GRE Tunnel	31
Prerequisites for Configuring Multicast Routing over GRE Tunnel	31
Restrictions for Configuring Multicast Routing over GRE Tunnel	31
Information About Multicast Routing over GRE Tunnel	32
How to Configure Multicast Routing over GRE Tunnel	32
Configuring a GRE Tunnel to Connect Non-IP Multicast Areas	32
Tunneling to Connect Non-IP Multicast Areas Example	33
Additional References for Multicast Routing over GRE Tunnel	35
Feature History for Multicast Routing over GRE Tunnel	35

CHAPTER 4**Configuring IGMP 37**

Prerequisites for IGMP and IGMP Snooping	37
Prerequisites for IGMP Snooping	37
Restrictions for IGMP and IGMP Snooping	38
Restrictions for Configuring IGMP	38
Restrictions for IGMP Snooping	38
Information about IGMP	39
Role of the Internet Group Management Protocol	39
IGMP Multicast Addresses	39
IGMP Versions	39
IGMP Version 1	40
IGMP Version 2	40
IGMP Version 3	40
IGMPv3 Host Signaling	40
IGMP Versions Differences	40
IGMP Join and Leave Process	43
IGMP Join Process	43
IGMP Leave Process	44
IGMP Snooping	44
Joining a Multicast Group	45
Leaving a Multicast Group	47
Immediate Leave	48
IGMP Configurable-Leave Timer	48
IGMP Report Suppression	48
IGMP Snooping and Device Stacks	48
IGMP Filtering and Throttling	49
Default IGMP Configuration	49
Default IGMP Snooping Configuration	50
Default IGMP Filtering and Throttling Configuration	50
How to Configure IGMP	51
Configuring the Device as a Member of a Group	51
Changing the IGMP Version	52
Modifying the IGMP Host-Query Message Interval	53

Changing the Maximum Query Response Time for IGMPv2	55
Configuring the Device as a Statically Connected Member	56
Configuring IGMP Profiles	58
Applying IGMP Profiles	60
Setting the Maximum Number of IGMP Groups	61
Configuring the IGMP Throttling Action	62
Configuring the Device to Forward Multicast Traffic in the Absence of Directly Connected IGMP Hosts	64
Controlling Access to an SSM Network Using IGMP Extended Access Lists	65
How to Configure IGMP Snooping	67
Enabling IGMP Snooping	67
Enabling or Disabling IGMP Snooping on a VLAN Interface	68
Setting the Snooping Method	69
Configuring a Multicast Router Port	70
Configuring a Host Statically to Join a Group	71
Enabling IGMP Immediate Leave	72
Configuring the IGMP Leave Timer	73
Configuring the IGMP Robustness-Variable	75
Configuring the IGMP Last Member Query Count	76
Configuring TCN-Related Commands	77
Controlling the Multicast Flooding Time After a TCN Event	77
Recovering from Flood Mode	78
Disabling Multicast Flooding During a TCN Event	79
Configuring the IGMP Snooping Querier	80
Disabling IGMP Report Suppression	82
Monitoring IGMP	83
Monitoring IGMP Snooping Information	84
Monitoring IGMP Filtering and Throttling Configuration	85
Configuration Examples for IGMP	86
Example: Configuring the Device as a Member of a Multicast Group	86
Example: Controlling Access to Multicast Groups	86
Examples: Configuring IGMP Snooping	86
Example: Configuring IGMP Profiles	87
Example: Applying IGMP Profile	87

Example: Setting the Maximum Number of IGMP Groups	88
Example: Interface Configuration as a Routed Port	88
Example: Interface Configuration as an SVI	88
Example: Configuring the Device to Forward Multicast Traffic in the Absence of Directly Connected IGMP Hosts	89
Controlling Access to an SSM Network Using IGMP Extended Access Lists	89
Example: Denying All States for a Group G	89
Example: Denying All States for a Source S	90
Example: Permitting All States for a Group G	90
Example: Permitting All States for a Source S	90
Example: Filtering a Source S for a Group G	90
Additional References for IGMP	91
Feature History for IGMP	91

CHAPTER 5
Configuring IGMP Proxy 93

Prerequisites for IGMP Proxy	93
Information About IGMP Proxy	93
IGMP Proxy	93
IGMP Proxy for a Single Upstream Interface	93
IGMP Proxy for Multiple Upstream Interfaces	96
How to Configure IGMP Proxy	97
Configuring the Upstream UDL Device for IGMP UDLR	97
Configuring the Downstream UDL Device for IGMP UDLR with IGMP Proxy Support	98
Configuring the Downstream Device for IGMP Proxy Join without UDLR	100
Configuring the Downstream Device for IGMP Proxy for Multiple Upstream Interfaces	102
Configuration Examples for IGMP Proxy	103
Example: Configuring the Upstream UDL Device for IGMP UDLR	103
Example: Configuring the Downstream UDL Device for IGMP UDLR with IGMP Proxy Support	103
Example: Configuring the Downstream Device for IGMP Proxy Join without UDLR	104
Example: Configuring the Downstream Device for IGMP Proxy for Multiple Upstream Interfaces	104
Additional References for IGMP Proxy	104
Feature History for IGMP Proxy	105

CHAPTER 6
IGMP Explicit Tracking 107

Restrictions for IGMP Explicit Tracking	107
Information About IGMP Explicit Tracking	108
IGMP Explicit Tracking	108
Minimal Leave Latencies	108
Faster Channel Changing	108
Improved Diagnostic Capabilities	108
How to Configure IGMP Explicit Tracking	109
Enabling Explicit Tracking Globally	109
Enabling Explicit Tracking on Layer 3 Interfaces	109
Configuration Examples for IGMP Explicit Tracking	110
Example: Enabling Explicit Tracking	110
Verifying IGMP Explicit Tracking	111
Additional References for IGMP Explicit Tracking	113
Feature History for IGMP Explicit Tracking	114

CHAPTER 7

Constraining IP Multicast in Switched Ethernet	115
Prerequisites for Constraining IP Multicast in a Switched Ethernet Network	115
Information About IP Multicast in a Switched Ethernet Network	115
IP Multicast Traffic and Layer 2 Switches	115
CGMP on Catalyst Switches for IP Multicast	116
IGMP Snooping	116
Router-Port Group Management Protocol (RGMP)	116
How to Constrain Multicast in a Switched Ethernet Network	117
Configuring Switches for IP Multicast	117
Configuring IGMP Snooping	117
Enabling CGMP	117
Configuring IP Multicast in a Layer 2 Switched Ethernet Network	118
Configuration Examples for Constraining IP Multicast in a Switched Ethernet Network	119
RGMP Configuration Example	119
Additional References for Constraining IP Multicast in a Switched Ethernet Network	120
Feature History for Constraining IP Multicast in Switched Ethernet	120

CHAPTER 8

Configuring Protocol Independent Multicast (PIM)	123
Prerequisites for PIM	123

Restrictions for PIM	124
PIMv1 and PIMv2 Interoperability	124
Restrictions for Configuring PIM Stub Routing	124
Restrictions for Configuring Auto-RP and BSR	125
Restrictions for Auto-RP Enhancement	126
Information about PIM	126
Protocol Independent Multicast Overview	126
PIM Versions	126
Multicast Source Discovery Protocol (MSDP)	127
PIM Sparse Mode	127
Bidirectional PIM	128
PIM Stub Routing	131
Rendezvous Points	132
Auto-RP	132
The Role of Auto-RP in a PIM Network	133
Multicast Boundaries	133
Sparse-Dense Mode for Auto-RP	134
Auto RP Benefits	135
PIM Domain Border	135
PIMv2 Bootstrap Router	135
Multicast Forwarding	136
Multicast Distribution Source Tree	136
Multicast Distribution Shared Tree	137
Source Tree Advantage	138
Shared Tree Advantage	138
PIM Shared Tree and Source Tree	139
Reverse Path Forwarding	140
RPF Check	141
Default PIM Routing Configuration	142
How to Configure PIM	143
Enabling PIM Stub Routing	143
Configuring a Rendezvous Point	144
Manually Assigning an RP to Multicast Groups	145
Setting Up Auto-RP in a New Internetwork	147

Adding Auto-RP to an Existing Sparse-Mode Cloud	149
Preventing Join Messages to False RPs	152
Filtering Incoming RP Announcement Messages	152
Configuring PIMv2 BSR	154
Defining the PIM Domain Border	154
Defining the IP Multicast Boundary	156
Configuring Candidate BSRs	157
Configuring the Candidate RPs	159
Configuring Sparse Mode with Auto-RP	160
Configuring IPv4 Bidirectional PIM	165
Enabling Bidirectional PIM Globally	165
Configuring the Rendezvous Point for IPv4 Bidirectional PIM Groups	165
Delaying the Use of PIM Shortest-Path Tree	166
Modifying the PIM Router-Query Message Interval	168
Verifying PIM Operations	169
Verifying IP Multicast Operation in a PIM-SM or a PIM-SSM Network	169
Verifying IP Multicast on the First Hop Router	170
Verifying IP Multicast on Routers Along the SPT	171
Verifying IP Multicast Operation on the Last Hop Router	172
Using PIM-Enabled Routers to Test IP Multicast Reachability	176
Configuring Routers to Respond to Multicast Pings	176
Pinging Routers Configured to Respond to Multicast Pings	177
Monitoring and Troubleshooting PIM	178
Monitoring PIM Information	178
Monitoring the RP Mapping and BSR Information	179
Troubleshooting PIMv1 and PIMv2 Interoperability Problems	179
Monitoring IPv4 Bidirectional PIM Information	180
Configuration Examples for PIM	180
Example: Enabling PIM Stub Routing	180
Example: Verifying PIM Stub Routing	181
Example: Manually Assigning an RP to Multicast Groups	181
Example: Configuring Auto-RP	181
Example: Sparse Mode with Auto-RP	181
Example: Defining the IP Multicast Boundary to Deny Auto-RP Information	182

Example: Filtering Incoming RP Announcement Messages	182
Example: Preventing Join Messages to False RPs	182
Example: Configuring Candidate BSRs	182
Example: Configuring Candidate RPs	183
Feature History for PIM	183

CHAPTER 9

Configuring PIM MIB Extension for IP Multicast 185

Information About PIM MIB Extension for IP Multicast	185
PIM MIB Extensions for SNMP Traps for IP Multicast	185
Benefits of PIM MIB Extensions	186
How to Configure PIM MIB Extension for IP Multicast	186
Enabling PIM MIB Extensions for IP Multicast	186
Configuration Examples for PIM MIB Extensions	187
Example Enabling PIM MIB Extensions for IP Multicast	187
Additional References for PIM MIB Extension for IP Multicast	188
Feature History for PIM MIB Extension for IP Multicast	188

CHAPTER 10

Configuring MSDP 189

Prerequisites for Using MSDP to Interconnect Multiple PIM-SM Domains	189
Information About Using MSDP to Interconnect Multiple PIM-SM Domains	189
Benefits of Using MSDP to Interconnect Multiple PIM-SM Domains	189
Use of MSDP to Interconnect Multiple PIM-SM Domains	189
MSDP Message Types	192
SA Messages	192
SA Request Messages	193
SA Response Messages	193
Keepalive Messages	193
SA Message Origination Receipt and Processing	193
SA Message Origination	193
SA Message Receipt	194
SA Message Processing	196
MSDP Peers	196
MSDP MD5 Password Authentication	197
How MSDP MD5 Password Authentication Works	197

Benefits of MSDP MD5 Password Authentication	197
SA Message Limits	197
MSDP Keepalive and Hold-Time Intervals	197
MSDP Connection-Retry Interval	198
Default MSDP Peers	198
MSDP Mesh Groups	199
Benefits of MSDP Mesh Groups	200
SA Origination Filters	200
Use of Outgoing Filter Lists in MSDP	200
Use of Incoming Filter Lists in MSDP	201
TTL Thresholds in MSDP	202
SA Request Messages	202
SA Request Filters	203
How to Use MSDP to Interconnect Multiple PIM-SM Domains	203
Configuring an MSDP Peer	203
Shutting Down an MSDP Peer	204
Configuring MSDP MD5 Password Authentication Between MSDP Peers	205
Troubleshooting Tips	206
Preventing DoS Attacks by Limiting the Number of SA Messages Allowed in the SA Cache from Specified MSDP Peers	207
Adjusting the MSDP Keepalive and Hold-Time Intervals	208
Adjusting the MSDP Connection-Retry Interval	209
Configuring a Default MSDP Peer	209
Configuring an MSDP Mesh Group	210
Controlling SA Messages Originated by an RP for Local Sources	211
Controlling the Forwarding of SA Messages to MSDP Peers Using Outgoing Filter Lists	212
Controlling the Receipt of SA Messages from MSDP Peers Using Incoming Filter Lists	213
Using TTL Thresholds to Limit the Multicast Data Sent in SA Messages	214
Requesting Source Information from MSDP Peers	214
Controlling the Response to Outgoing SA Request Messages from MSDP Peers Using SA Request Filters	215
Configuring an Originating Address Other Than the RP Address	216
Monitoring MSDP	217
Clearing MSDP Connections Statistics and SA Cache Entries	219

Enabling SNMP Monitoring of MSDP	220
Troubleshooting Tips	221
Configuration Examples for Using MSDP to Interconnect Multiple PIM-SM Domains	221
Example: Configuring an MSDP Peer	221
Example: Configuring MSDP MD5 Password Authentication	222
Example: Configuring a Default MSDP Peer	222
Example: Configuring MSDP Mesh Groups	224
Additional References Multicast Source Discovery Protocol	224
Feature History for Multicast Source Discovery Protocol	224

CHAPTER 11

Configuring SSM 227

Prerequisites for Configuring SSM	227
Restrictions for Configuring SSM	227
Information About SSM	229
SSM Components Overview	229
SSM and Internet Standard Multicast (ISM)	229
SSM IP Address Range	229
SSM Operations	230
SSM Mapping	230
Static SSM Mapping	230
DNS-Based SSM Mapping	231
How to Configure SSM	232
Configuring SSM	232
Configuring Source Specific Multicast Mapping	233
Configuring Static SSM Mapping	233
Configuring DNS-Based SSM Mapping	235
Configuring Static Traffic Forwarding with SSM Mapping	236
Monitoring SSM	238
Monitoring SSM Mapping	238
Where to Go Next for SSM	239
Additional References for SSM	239
Feature History for SSM	239

CHAPTER 12

Implementing IPv6 Multicast 241

Information About Implementing IPv6 Multicast Routing	241
IPv6 Multicast Overview	241
IPv6 Multicast Routing Implementation	242
IPv6 Multicast Listener Discovery Protocol	242
Multicast Queriers and Hosts	242
MLD Access Group	243
Explicit Tracking of Receivers	243
Protocol Independent Multicast	243
PIM-Sparse Mode	243
IPv6 BSR: Configure RP Mapping	244
PIM-Source Specific Multicast	244
Routable Address Hello Option	245
PIM IPv6 Stub Routing	245
Rendezvous Point	246
Static Mroutes	247
MRIB	247
MFIB	247
MFIB	247
IPv6 Multicast Process Switching and Fast Switching	248
Multiprotocol BGP for the IPv6 Multicast Address Family	248
Implementing IPv6 Multicast	249
Enabling IPv6 Multicast Routing	249
Customizing and Verifying the MLD Protocol	250
Customizing and Verifying MLD on an Interface	250
Implementing MLD Group Limits	251
Configuring Explicit Tracking of Receivers to Track Host Behavior	253
Resetting the MLD Traffic Counters	253
Clearing the MLD Interface Counters	254
Configuring PIM	255
Configuring PIM-SM and Displaying PIM-SM Information for a Group Range	255
Configuring PIM Options	256
Resetting the PIM Traffic Counters	258
Clearing the PIM Topology Table to Reset the MRIB Connection	259
Configuring PIM IPv6 Stub Routing	260

PIM IPv6 Stub Routing Configuration Guidelines	260
Default IPv6 PIM Routing Configuration	261
Enabling IPV6 PIM Stub Routing	261
Monitoring IPv6 PIM Stub Routing	263
Configuring a BSR	263
Configuring a BSR and Verifying BSR Information	263
Sending PIM RP Advertisements to the BSR	264
Configuring BSR for Use Within Scoped Zones	265
Configuring BSR Switches to Announce Scope-to-RP Mappings	266
Configuring SSM Mapping	267
Configuring Static Mroutes	268
Using MFIB in IPv6 Multicast	269
Verifying MFIB Operation in IPv6 Multicast	269
Resetting MFIB Traffic Counters	270
Additional References	271
Feature History for IPv6 Multicast	271

CHAPTER 13

Configuring MLD Snooping 273

Information About Configuring IPv6 MLD Snooping	273
Understanding MLD Snooping	273
MLD Messages	274
MLD Queries	274
Multicast Client Aging Robustness	275
Multicast Router Discovery	275
MLD Reports	275
MLD Done Messages and Immediate-Leave	275
Topology Change Notification Processing	276
How to Configure IPv6 MLD Snooping	276
Default MLD Snooping Configuration	276
MLD Snooping Configuration Guidelines	277
Enabling or Disabling MLD Snooping on the Switch	277
Enabling or Disabling MLD Snooping on a VLAN	278
Configuring a Static Multicast Group	279
Configuring a Multicast Router Port	280

Enabling MLD Immediate Leave	281
Configuring MLD Snooping Queries	282
Disabling MLD Listener Message Suppression	284
Displaying MLD Snooping Information	284
Configuration Examples for Configuring MLD Snooping	285
Configuring a Static Multicast Group: Example	285
Configuring a Multicast Router Port: Example	286
Enabling MLD Immediate Leave: Example	286
Configuring MLD Snooping Queries: Example	286
Additional References	286
Feature History for MLD Snooping	287

CHAPTER 14

Configuring Multicast Virtual Private Network	289
Prerequisites for Configuring Multicast VPN	289
Restrictions for Configuring Multicast VPN	289
Information About Configuring Multicast VPN	289
Multicast VPN Operation	290
Benefits of Multicast VPN	290
Multicast VPN Routing and Forwarding and Multicast Domains	290
Multicast Distribution Trees	290
Multicast Tunnel Interface	292
MDT Address Family in BGP for Multicast VPN	293
BGP Advertisement Methods for Multicast VPN Support	293
How to Configure Multicast VPN	293
Configuring the Data Multicast Group	293
Configuring a Default MDT Group for a VRF	295
Configuring the MDT Address Family in BGP for Multicast VPN	297
Verifying Information for the MDT Default Group	299
Configuration Examples for Multicast VPN	300
Example: Configuring MVPN and SSM	300
Example: Enabling a VPN for Multicast Routing	300
Example: Configuring the Multicast Group Address Range for Data MDT Groups	300
Example: Limiting the Number of Multicast Routes	301
Additional References for Configuring Multicast VPN	301

Feature History for Multicast VPN	301
-----------------------------------	-----

CHAPTER 15

Configuring MVPNv6 303

Prerequisites for MVPNv6	303
Restrictions for MVPNv6	303
Information about MVPNv6	303
How to Configure MVPNv6	304
Configuring Multicast Routing	304
Configuring MVRF on PE Devices	305
Configuring Routing Protocols Between the PE and CE Devices	306
Configuration Example for MVPNv6	307
Feature History for MVPNv6	308

CHAPTER 16

Configuring Multicast VPN Extranet Support 309

Restrictions for Configuring mVPN Extranet Support	309
Information About mVPN Extranet Support	309
Overview of mVPN Extranet Support	310
mVPN Extranet Support Configuration - Option 1	311
mVPN Extranet Support Configuration - Option 2	312
RPF for mVPN Extranet Support Using Imported Routes	312
RPF for mVPN Extranet Support Using Static Mroutes	313
mVPN Extranet VRF Select	313
How to Configure mVPN Extranet Support	314
Configuring mVPN Support	314
Configuring the Source MVRF on the Receiver PE - Option 1	314
Configuring the Receiver MVRF on the Source PE - Option 2	316
Configuring RPF for mVPN Extranet Support Using Static Mroutes	318
Configuring Group-Based VRF Selection Policies with mVPN Extranet	319
Configuration Examples for mVPN Extranet Support	320
Example: Configuring the Source VRF on the Receiver PE Router- Option 1	320
Example: Configuring the Receiver VRF on the Source PE Router - Option 2	326
Example: Displaying Statistics for mVPN Extranet Support	332
Example: Configuring RPF for mVPN Extranet Support Using Static Mroutes	335
Example: Configuring Group-Based VRF Selection Policies with mVPN Extranet Support	335

Additional References	336
Feature History for mVPN Extranet Support	336

CHAPTER 17

MLDP-Based MVPN	337
MLDP-Based MVPN	337
Prerequisites for MLDP-Based MVPN	337
Restrictions for MLDP-Based VPN	337
Information About MLDP-Based MVPN	338
Overview of MLDP-Based MVPN	338
Initial Deployment of an MLDP-Based MVPN	340
Default MDT Creation	340
Data MDT Scenario	346
P2MP and MP2MP Label Switched Paths	347
Packet Flow in MLDP-based MVPN	348
Realizing an MLDP-based MVPN	348
Overview of MVPN MLDP Partitioned MDT	348
Supported MLDP Profiles	349
How to Configure MLDP-Based MVPN	350
Configuring Initial MLDP Settings	350
Configuring an MLDP-Based MVPN	351
Verifying the Configuration of an MLDP-Based MVPN	352
Configuration Examples for MLDP-Based MVPN	354
Example: Initial Deployment of an MLDP-Based MVPN	354
Default MDT Configuration	355
Data MDT Configuration	359
Example: Configuring MVPN Profile 1 - Default MDT - MLDP MP2MP - PIM C-mcast Signaling	362
Example: Configuring MVPN Profile 13 - Default MDT - MLDP - MP2MP - BGP-AD - BGP C-mcast Signaling	363
Example: Configuring MVPN Profile 14 - Partitioned MDT - MLDP P2MP - BGP-AD - BGP C-mast Signaling	364
Feature History for MLDP-Based MVPN	364

CHAPTER 18

IP Multicast Optimization: Optimizing PIM Sparse Mode in a Large IP Multicast Deployment	367
---	------------

Prerequisites for Optimizing PIM Sparse Mode in a Large IP Multicast Deployment	367
Information About Optimizing PIM Sparse Mode in a Large IP Multicast Deployment	367
PIM Registering Process	367
PIM Version 1 Compatibility	368
PIM Designated Router	368
PIM Sparse-Mode Register Messages	369
Preventing Use of Shortest-Path Tree to Reduce Memory Requirement	369
PIM Shared Tree and Source Tree - Shortest-Path Tree	369
Benefit of Preventing or Delaying the Use of the Shortest-Path Tree	370
How to Optimize PIM Sparse Mode in a Large IP Multicast Deployment	370
Optimizing PIM Sparse Mode in a Large Deployment	370
Configuration Examples for Optimizing PIM Sparse Mode in a Large Multicast Deployment	372
Optimizing PIM Sparse Mode in a Large IP Multicast Deployment Example	372
Additional References for IP Multicast Optimization: Optimizing PIM Sparse Mode in a Large IP Multicast Deployment	373
Feature History for IP Multicast Optimization: Optimizing PIM Sparse Mode in a Large IP Multicast Deployment	373

CHAPTER 19

IP Multicast Optimization: Multicast Subsecond Convergence	375
Prerequisites for Multicast Subsecond Convergence	375
Restrictions for Multicast Subsecond Convergence	375
Information About Multicast Subsecond Convergence	375
Benefits of Multicast Subsecond Convergence	375
Multicast Subsecond Convergence Scalability Enhancements	376
PIM Router Query Messages	376
Reverse Path Forwarding	376
Topology Changes and Multicast Routing Recovery	376
How to Configure Multicast Subsecond Convergence	377
Modifying the PIM Router Query Message Interval	377
Verifying Multicast Subsecond Convergence Configurations	377
Configuration Examples for Multicast Subsecond Convergence	378
Modifying the PIM Router Query Message Interval Example	378
Additional References for IP Multicast Optimization: Multicast Subsecond Convergence	379
Feature History for IP Multicast Optimization Multicast Subsecond Convergence	379

CHAPTER 20

IP Multicast Optimization: IP Multicast Load Splitting across Equal-Cost Paths 381

Prerequisites for IP Multicast Load Splitting across Equal-Cost Paths	381
Information About IP Multicast Load Splitting across Equal-Cost Paths	381
Load Splitting Versus Load Balancing	381
Default Behavior for IP Multicast When Multiple Equal-Cost Paths Exist	382
Methods to Load Split IP Multicast Traffic	383
Overview of ECMP Multicast Load Splitting	384
ECMP Multicast Load Splitting Based on Source Address Using the S-Hash Algorithm	384
ECMP Multicast Load Splitting Based on Source and Group Address Using the Basic S-G-Hash Algorithm	384
Predictability As a By-Product of Using the S-Hash and Basic S-G-Hash Algorithms	384
Polarization As a By-Product of Using the S-Hash and Basic S-G-Hash Algorithms	384
ECMP Multicast Load Splitting Based on Source Group and Next-Hop Address	385
Effect of ECMP Multicast Load Splitting on PIM Neighbor Query and Hello Messages for RPF Path Selection	386
Effect of ECMP Multicast Load Splitting on the PIM Assert Process in PIM-SM and PIM-SSM	387
ECMP Multicast Load Splitting and Reconvergence When Unicast Routing Changes	387
Use of BGP with ECMP Multicast Load Splitting	388
Use of ECMP Multicast Load Splitting with Static Mroutes	388
Alternative Methods of Load Splitting IP Multicast Traffic	388
How to Load Split IP Multicast Traffic over ECMP	389
Enabling ECMP Multicast Load Splitting	389
Prerequisites for IP Multicast Load Splitting - ECMP	389
Restrictions for IP Multicast Load Splitting -ECMP	390
Enabling ECMP Multicast Load Splitting Based on Source Address	390
Enabling ECMP Multicast Load Splitting Based on Source and Group Address	392
Enabling ECMP Multicast Load Splitting Based on Source Group and Next-Hop Address	393
Configuration Examples for Load Splitting IP Multicast Traffic over ECMP	395
Example Enabling ECMP Multicast Load Splitting Based on Source Address	395
Example Enabling ECMP Multicast Load Splitting Based on Source and Group Address	395
Example Enabling ECMP Multicast Load Splitting Based on Source Group and Next-Hop Address	396
Additional References for IP Multicast Optimization: IP Multicast Load Splitting across Equal-Cost Paths	396

Feature History for IP Multicast Optimization: IP Multicast Load Splitting across Equal-Cost Paths	396
--	-----

CHAPTER 21

IP Multicast Optimization: SSM Channel Based Filtering for Multicast	399
Prerequisites for SSM Channel Based Filtering for Multicast Boundaries	399
Information About the SSM Channel Based Filtering for Multicast Boundaries	399
Rules for Multicast Boundaries	399
Benefits of SSM Channel Based Filtering for Multicast Boundaries	400
How to Configure SSM Channel Based Filtering for Multicast Boundaries	400
Configuring Multicast Boundaries	400
Configuration Examples for SSM Channel Based Filtering for Multicast Boundaries	401
Configuring the Multicast Boundaries Permitting and Denying Traffic Example	401
Configuring the Multicast Boundaries Permitting Traffic Example	402
Configuring the Multicast Boundaries Denying Traffic Example	402
Additional References for IP Multicast Optimization: SSM Channel-Based Filtering for Multicast	402
Feature History for IP Multicast Optimization: SSM Channel Based Filtering for Multicast	403

CHAPTER 22

IP Multicast Optimization: IGMP State Limit	405
Prerequisites for IGMP State Limit	405
Restrictions for IGMP State Limit	405
Information About IGMP State Limit	405
IGMP State Limit	405
IGMP State Limit Feature Design	406
Mechanics of IGMP State Limiters	406
How to Configure IGMP State Limit	407
Configuring IGMP State Limiters	407
Configuring Global IGMP State Limiters	407
Configuring Per Interface IGMP State Limiters	407
Configuration examples for IGMP State Limit	408
Configuring IGMP State Limiters Example	409
Additional References	410
Feature History for IP Multicast Optimization: IGMP State Limit	410



CHAPTER 1

IP Multicast Routing Technology Overview

- [Information About IP Multicast Technology, on page 1](#)
- [Additional References for IP Multicast, on page 16](#)

Information About IP Multicast Technology

This section provides information about IP multicast technology.

About IP Multicast

Controlling the transmission rate to a multicast group is not supported.

At one end of the IP communication spectrum is IP unicast, where a source IP host sends packets to a specific destination IP host. In IP unicast, the destination address in the IP packet is the address of a single, unique host in the IP network. These IP packets are forwarded across the network from the source to the destination host by devices. At each point on the path between source and destination, a device uses a unicast routing table to make unicast forwarding decisions, based on the IP destination address in the packet.

At the other end of the IP communication spectrum is an IP broadcast, where a source host sends packets to all hosts on a network segment. The destination address of an IP broadcast packet has the host portion of the destination IP address set to all ones and the network portion set to the address of the subnet. IP hosts, including devices, understand that packets, which contain an IP broadcast address as the destination address, are addressed to all IP hosts on the subnet. Unless specifically configured otherwise, devices do not forward IP broadcast packets, so IP broadcast communication is normally limited to a local subnet.

IP multicasting falls between IP unicast and IP broadcast communication. IP multicast communication enables a host to send IP packets to a group of hosts anywhere within the IP network. To send information to a specific group, IP multicast communication uses a special form of IP destination address called an IP multicast group address. The IP multicast group address is specified in the IP destination address field of the packet.

To multicast IP information, Layer 3 switches and devices must forward an incoming IP packet to all output interfaces that lead to members of the IP multicast group.

We tend to think of IP multicasting and video conferencing as the same thing. Although the first application in a network to use IP multicast is often video conferencing, video is only one of many IP multicast applications that can add value to a company's business model. Other IP multicast applications that have potential for improving productivity include multimedia conferencing, data replication, real-time data multicasts, and simulation applications.

Role of IP Multicast in Information Delivery

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

IP multicast routing enables a host (source) to send packets to a group of hosts (receivers) anywhere within the IP network by using a special form of IP address called the IP multicast group address. The sending host inserts the multicast group address into the IP destination address field of the packet and IP multicast routers and multilayer switches forward incoming IP multicast packets out all interfaces that lead to the members of the multicast group. Any host, regardless of whether it is a member of a group, can send to a group. However, only the members of a group receive the message.

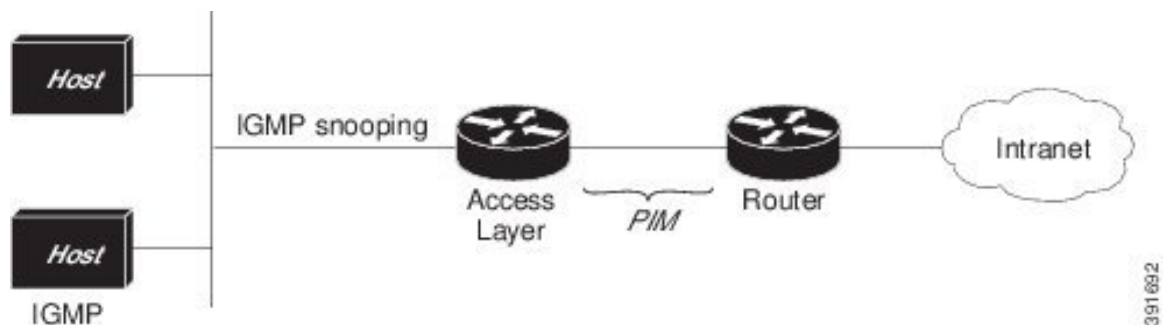
IP Multicast Routing Protocols

The software supports the following protocols to implement IP multicast routing:

- IGMP is used between hosts on a LAN and the routers (and multilayer devices) on that LAN to track the multicast groups of which hosts are members. To participate in IP multicasting, multicast hosts, routers, and multilayer devices must have the Internet Group Management Protocol (IGMP) operating.
- Protocol Independent Multicast (PIM) is used between routers so that they can track which multicast packets to forward to each other and to their directly connected LANs.
- IGMP Snooping is used for multicasting in a Layer 2 switching environment. It helps reduce the flooding of multicast traffic by dynamically configuring Layer 2 interfaces so that multicast traffic is forwarded to only those interfaces associated with IP multicast devices.

This figure shows where these protocols operate within the IP multicast environment.

Figure 1: IP Multicast Routing Protocols



According to IPv4 multicast standards, the MAC destination multicast address begins with 0100:5e and is appended by the last 23 bits of the IP address. For example, if the IP destination address is 239.1.1.39, the MAC destination address is 0100:5e01:0127.

A multicast packet is unmatched when the destination IPv4 address does not match the destination MAC address. The device forwards the unmatched packet in hardware based upon the MAC address table. If the destination MAC address is not in the MAC address table, the device floods the packet to the all port in the same VLAN as the receiving port.

Internet Group Management Protocol

IGMP messages are used by IP multicast hosts to send their local Layer 3 switch or router a request to join a specific multicast group and begin receiving multicast traffic. With some extensions in IGMPv2, IP hosts can also send a request to a Layer 3 switch or router to leave an IP multicast group and not receive the multicast group traffic.

Using the information obtained by using IGMP, a Layer 3 switch or router maintains a list of multicast group memberships on a per-interface basis. A multicast group membership is active on an interface if at least one host on the interface sends an IGMP request to receive multicast group traffic.

Protocol-Independent Multicast

Protocol-Independent Multicast (PIM) is protocol independent because it can leverage whichever unicast routing protocol is used to populate the unicast routing table, including EIGRP, OSPF, BGP, or static route, to support IP multicast.

PIM also uses a unicast routing table to perform the reverse path forwarding (RPF) check function instead of building a completely independent multicast routing table. PIM does not send and receive multicast routing updates between routers like other routing protocols do.

PIM Sparse Mode

PIM Sparse Mode (PIM-SM) uses a pull model to deliver multicast traffic. Only networks with active receivers that have explicitly requested the data are forwarded the traffic. PIM-SM is intended for networks with several different multicasts, such as desktop video conferencing and collaborative computing, that go to a small number of receivers and are typically in progress simultaneously.

Rendezvous Point

If you configure PIM to operate in sparse mode, you must also choose one or more devices to be rendezvous points (RPs). Senders to a multicast group use RPs to announce their presence. Receivers of multicast packets use RPs to learn about new senders. You can configure Cisco IOS software so that packets for a single multicast group can use one or more RPs.

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

A PIM device can be an RP for more than one group. Only one RP address can be used at a time within a PIM domain for the same group. The conditions specified by the access list determine for which groups the device is an RP (as different groups can have different RPs).

IGMP Snooping

IGMP snooping is used for multicasting in a Layer 2 switching environment. With IGMP snooping, a Layer 3 switch or router examines Layer 3 information in the IGMP packets in transit between hosts and a device. When the switch receives the IGMP Host Report from a host for a particular multicast group, the switch adds the host's port number to the associated multicast table entry. When the switch receives the IGMP Leave Group message from a host, it removes the host's port from the table entry.

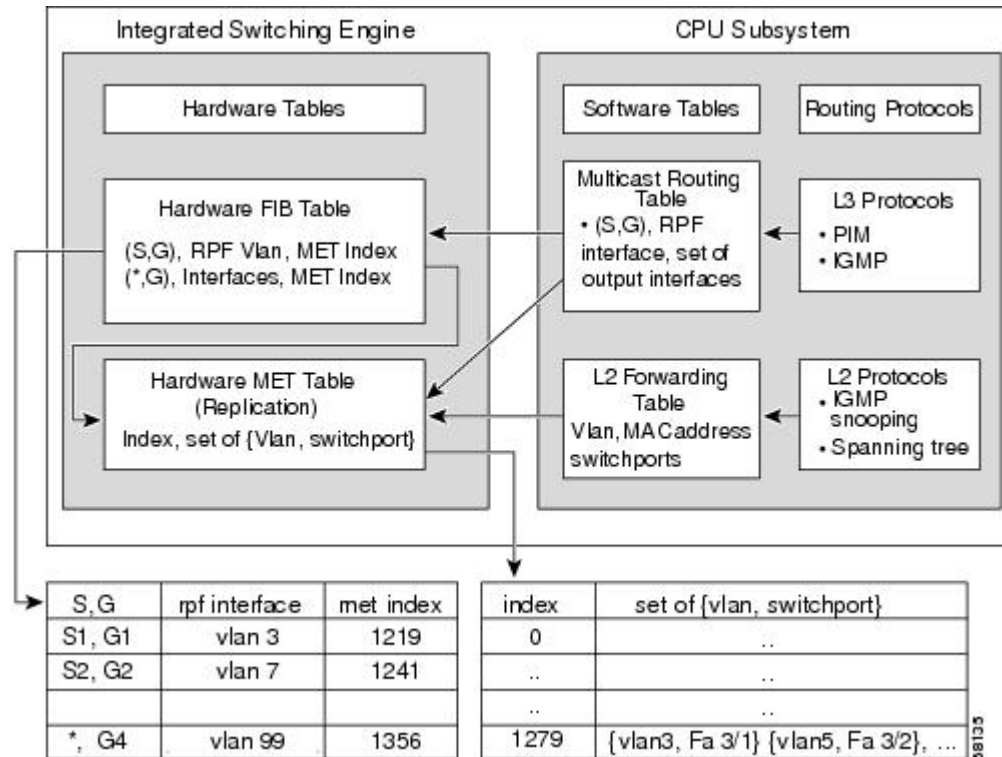
Because IGMP control messages are transmitted as multicast packets, they are indistinguishable from multicast data if only the Layer 2 header is examined. A switch running IGMP snooping examines every multicast data packet to determine whether it contains any pertinent IGMP control information. If IGMP snooping is

implemented on a low end switch with a slow CPU, performance could be severely impacted when data is transmitted at high rates.

IP Multicast Tables

The following illustration shows some key data structures that the device uses to forward IP multicast packets in hardware.

Figure 2: IP Multicast Tables and Protocols



The Integrated Switching Engine maintains the hardware FIB table to identify individual IP multicast routes. Each entry consists of a destination group IP address and an optional source IP address. Multicast traffic flows on primarily two types of routes: (S,G) and (*,G). The (S,G) routes flow from a source to a group based on the IP address of the multicast source and the IP address of the multicast group destination. Traffic on a (*,G) route flows from the PIM RP to all receivers of group G. Only sparse-mode groups use (*,G) routes. The Integrated Switching Engine hardware contains space for a total of 128,000 routes, which are shared by unicast routes, multicast routes, and multicast fast-drop entries.

Output interface lists are stored in the multicast expansion table (MET). The MET has room for up to 32,000 output interface lists. (For RET, we can have up to 102 K entries (32 K used for floodsets, 70,000 used for multicast entries)). The MET resources are shared by both Layer 3 multicast routes and by Layer 2 multicast entries. The actual number of output interface lists available in hardware depends on the specific configuration. If the total number of multicast routes exceed 32,000, multicast packets might not be switched by the Integrated Switching Engine. They would be forwarded by the CPU subsystem at much slower speeds.



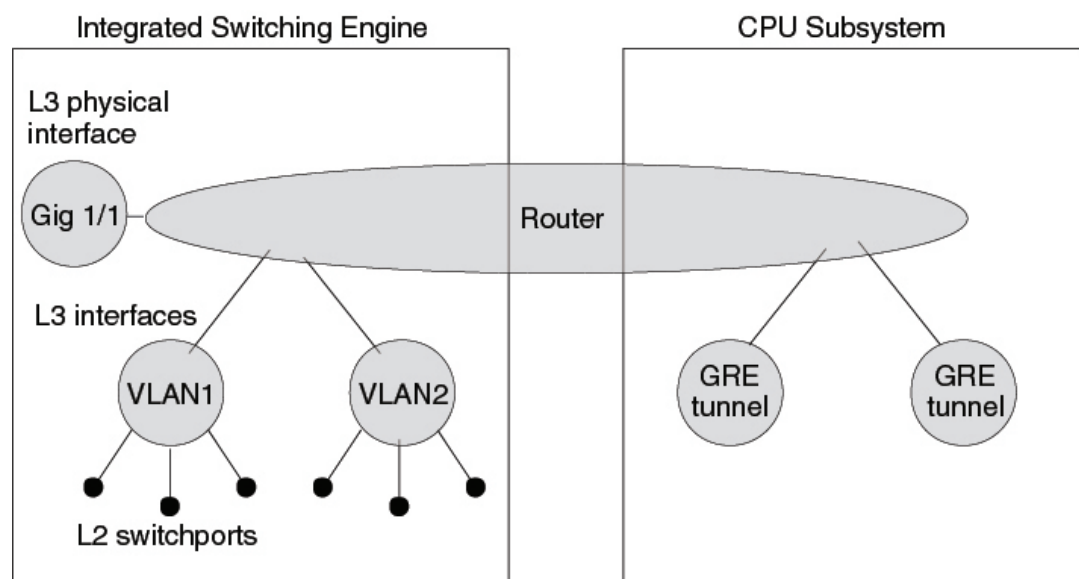
Note For RET, a maximum of 102 K entries is supported (32 K used for floodsets, 70 K used for multicast entries).

Hardware and Software Forwarding

The Integrated Switching Engine forwards the majority of packets in hardware at very high rates of speed. The CPU subsystem forwards exception packets in software. Statistical reports should show that the Integrated Switching Engine is forwarding the vast majority of packets in hardware.

The following illustration shows a logical view of the hardware and software forwarding components.

Figure 3: Hardware and Software Forwarding Components



68127

In the normal mode of operation, the Integrated Switching Engine performs inter-VLAN routing in hardware. The CPU subsystem supports generic routing encapsulation (GRE) tunnels for forwarding in software.

Replication is a particular type of forwarding where, instead of sending out one copy of the packet, the packet is replicated and multiple copies of the packet are sent out. At Layer 3, replication occurs only for multicast packets; unicast packets are never replicated to multiple Layer 3 interfaces. In IP multicasting, for each incoming IP multicast packet that is received, many replicas of the packet are sent out.

IP multicast packets can be transmitted on the following types of routes:

- Hardware routes
- Software routes
- Partial routes

Hardware routes occur when the Integrated Switching Engine hardware forwards all replicas of a packet. Software routes occur when the CPU subsystem software forwards all replicas of a packet. Partial routes occur when the Integrated Switching Engine forwards some of the replicas in hardware and the CPU subsystem forwards some of the replicas in software.

Partial Routes



Note The conditions listed below cause the replicas to be forwarded by the CPU subsystem software, but the performance of the replicas that are forwarded in hardware is not affected.

The following conditions cause some replicas of a packet for a route to be forwarded by the CPU subsystem:

- The switch is configured with the **ip igmp join-group** command as a member of the IP multicast group on the RPF interface of the multicast source.
- The switch is the first-hop to the source in PIM sparse mode. The switch must send PIM-register messages to the RP.

Software Routes



Note If any one of the following conditions is configured on the RPF interface or the output interface, all replication of the output is performed in software.

The following conditions cause all replicas of a packet for a route to be forwarded by the CPU subsystem software:

- The interface is configured with multicast helper.
- The interface is a generic routing encapsulation (GRE) or Distance Vector Multicast Routing Protocol (DVMRP) tunnel.
- The interface uses non-Advanced Research Products Agency (ARPA) encapsulation.

The following packets are always forwarded in software:

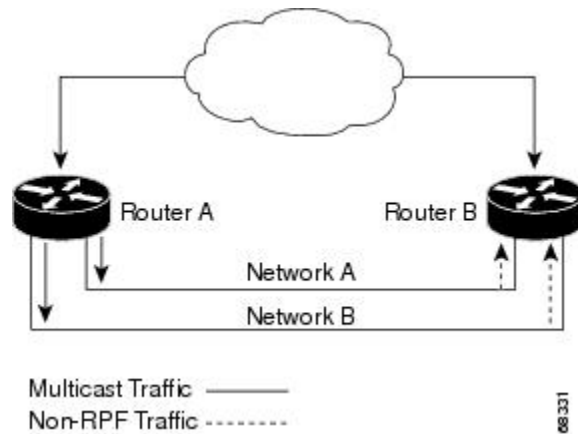
- Packets sent to multicast groups that fall into the range 224.0.0.* (where * is in the range from 0 to 255). This range is used by routing protocols. Layer 3 switching supports all other multicast group addresses.
- Packets with IP options.

Non-Reverse Path Forwarding Traffic

Traffic that fails an Reverse Path Forwarding (RPF) check is called non-RPF traffic. Non-RPF traffic is forwarded by the Integrated Switching Engine by filtering (persistently dropping) or rate limiting the non-RPF traffic.

In a redundant configuration where multiple Layer 3 switches or routers connect to the same LAN segment, only one device forwards the multicast traffic from the source to the receivers on the outgoing interfaces. The following illustration shows how non-RPF traffic can occur in a common network configuration.

Figure 4: Redundant Multicast Router Configuration in a Stub Network



In this kind of topology, only Router A, the PIM designated router (PIM DR), forwards data to the common VLAN. Router B receives the forwarded multicast traffic, but must drop this traffic because it has arrived on the wrong interface and fails the RPF check. Traffic that fails the RPF check is called non-RPF traffic.

Multicast Group Transmission Scheme

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

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

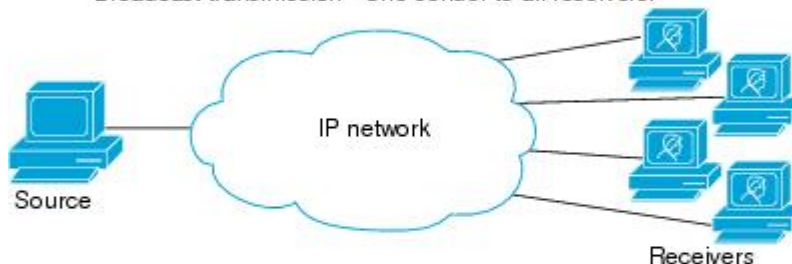
In a multicast environment, any host, regardless of whether it is a member of a group, can send to a group. However, only the members of a group can receive packets sent to that group. Multicast packets are delivered to a group using best-effort reliability, just like IP unicast packets.

Multicast Group Transmission Scheme

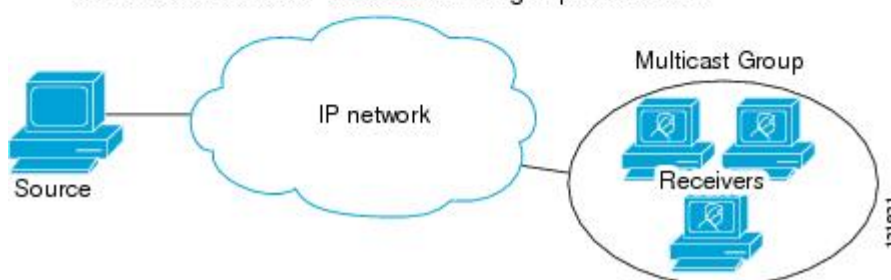
Unicast transmission—One host sends and the other receives.



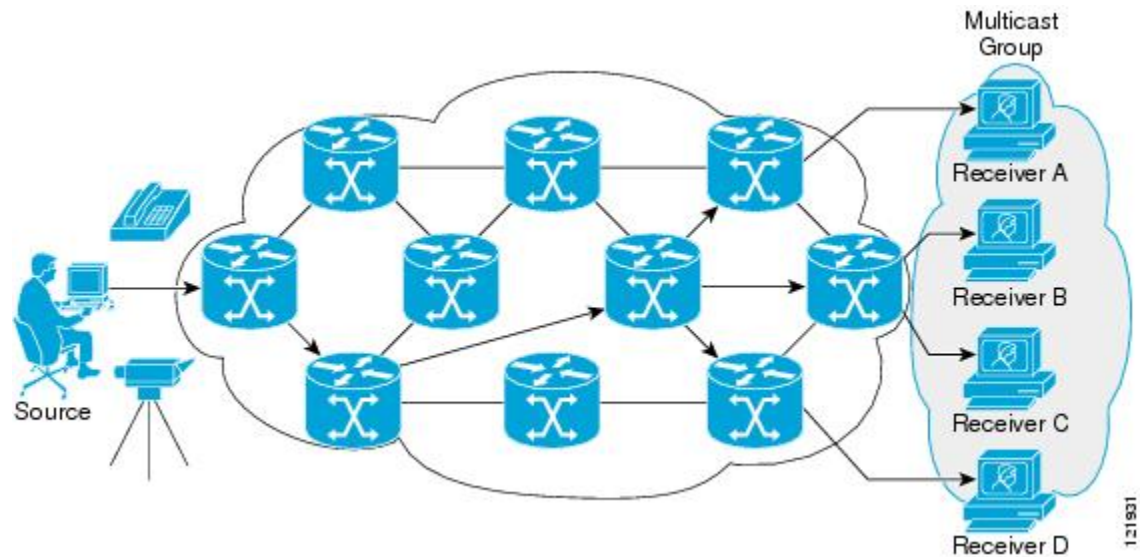
Broadcast transmission—One sender to all receivers.



Multicast transmission—One sender to a group of receivers.



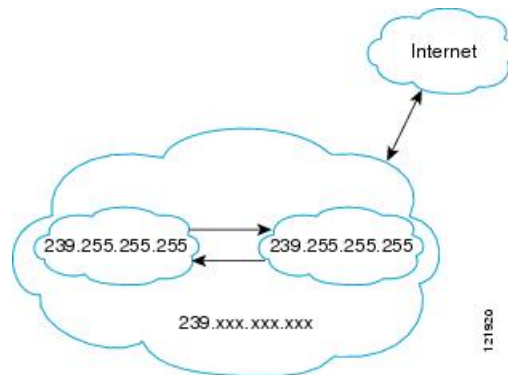
In the next figure, the receivers (the designated multicast group) are interested in receiving the video data stream from the source. The receivers indicate their interest by sending an IGMP host report to the routers in the network. The routers are then responsible for delivering the data from the source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



IP Multicast Boundary

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

Figure 5: Address Scoping at Boundaries



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

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

You can configure the **filter-autorp** keyword to examine and filter Auto-RP discovery and announcement messages at the administratively scoped boundary. Any Auto-RP group range announcements from the Auto-RP packets that are denied by the boundary access control list (ACL) are removed. An Auto-RP group

range announcement is permitted and passed by the boundary only if all addresses in the Auto-RP group range are permitted by the boundary ACL. If any address is not permitted, the entire group range is filtered and removed from the Auto-RP message before the Auto-RP message is forwarded.

IP Multicast Group Addressing

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

IP Class D Addresses

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



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

IP Multicast Address Scoping

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

Table 1: Multicast Address Range Assignments

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

Reserved Link-Local Addresses

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

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

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



Note All the packets with reserved link-local addresses are punted to CPU by default in the ASR 903 RSP2 Module.

Globally Scoped Addresses

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

Source Specific Multicast Addresses

Addresses in the range 232.0.0.0/8 are reserved for Source Specific Multicast (SSM) by IANA. In Cisco IOS software, you can use the **ip pim ssm** command to configure SSM for arbitrary IP multicast addresses also. SSM is an extension of Protocol Independent Multicast (PIM) that allows for an efficient data delivery mechanism in one-to-many communications. SSM is described in the [IP Multicast Delivery Modes, on page 14](#) section.

GLOP Addresses

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

Limited Scope Addresses

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



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

Layer 2 Multicast Addresses

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

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

Cisco Express Forwarding, MFIB, and Layer 2 Forwarding

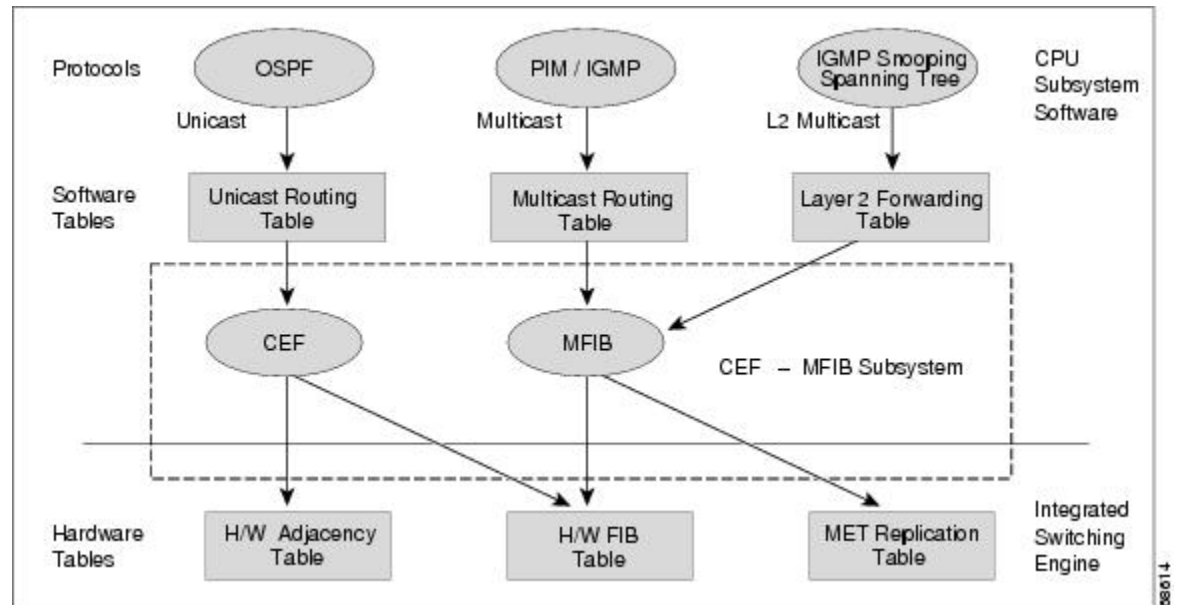
The implementation of IP multicast is an extension of centralized Cisco Express Forwarding. Cisco Express Forwarding extracts information from the unicast routing table, which is created by unicast routing protocols, such as BGP, OSPF, and EIGRP and loads it into the hardware

Forwarding Information Base (FIB). With the unicast routes in the FIB, when a route is changed in the upper-layer routing table, only one route needs to be changed in the hardware routing state. To forward unicast packets in hardware, the Integrated Switching Engine looks up source and destination routes in ternary content addressable memory (TCAM), takes the adjacency index from the hardware FIB, and gets the Layer 2 rewrite information and next-hop address from the hardware adjacency table.

The new Multicast Forwarding Information Base (MFIB) subsystem is the multicast analog of the unicast Cisco Express Forwarding. The MFIB subsystem extracts the multicast routes that PIM and IGMP create and refines them into a protocol-independent format for forwarding in hardware. The MFIB subsystem removes the protocol-specific information and leaves only the essential forwarding information. Each entry in the MFIB table consists of an (S,G) or (*,G) route, an input RPF VLAN, and a list of Layer 3 output interfaces. The MFIB subsystem, together with platform-dependent management software, loads this multicast routing information into the hardware FIB and Replica Expansion Table (RET). The device performs Layer 3 routing and Layer 2 bridging at the same time. There can be multiple Layer 2 switch ports on any VLAN interface.

The following illustration shows a functional overview of how a Cisco device combines unicast routing, multicast routing, and Layer 2 bridging information to forward in hardware:

Figure 6: Combining Cisco Express Forwarding, MFIB, and Layer 2 Forwarding Information in Hardware



Like the Cisco Express Forwarding unicast routes, the MFIB routes are Layer 3 and must be merged with the appropriate Layer 2 information. The following example shows an MFIB route:

```
(* ,203.0.113.1)
RPF interface is Vlan3
Output Interfaces are:
Vlan 1
Vlan 2
```

The route (*,203.0.113.1) is loaded in the hardware FIB table and the list of output interfaces is loaded into the MET. A pointer to the list of output interfaces, the MET index, and the RPF interface are also loaded in the hardware FIB with the (*,203.0.113.1) route. With this information loaded in hardware, merging of the Layer 2 information can begin. For the output interfaces on VLAN1, the Integrated Switching Engine must send the packet to all switch ports in VLAN1 that are in the spanning tree forwarding state. The same process applies to VLAN 2. To determine the set of switch ports in VLAN 2, the Layer 2 Forwarding Table is used.

When the hardware routes a packet, in addition to sending it to all of the switch ports on all output interfaces, the hardware also sends the packet to all switch ports (other than the one it arrived on) in the input VLAN. For example, assume that VLAN 3 has two switch ports in it, GigabitEthernet 3/1 and GigabitEthernet 3/2. If a host on GigabitEthernet 3/1 sends a multicast packet, the host on GigabitEthernet 3/2 might also need to receive the packet. To send a multicast packet to the host on GigabitEthernet 3/2, all of the switch ports in the ingress VLAN must be added to the port set that is loaded in the MET.

If VLAN 1 contains 1/1 and 1/2, VLAN 2 contains 2/1 and 2/2, and VLAN 3 contains 3/1 and 3/2, the MET chain for this route would contain these switch ports: (1/1,1/2,2/1,2/2,3/1, and 3/2).

If IGMP snooping is on, the packet should not be forwarded to all output switch ports on VLAN 2. The packet should be forwarded only to switch ports where IGMP snooping has determined that there is either a group member or router. For example, if VLAN 1 had IGMP snooping enabled, and IGMP snooping determined that only port 1/2 had a group member on it, then the MET chain would contain these switch ports: (1/1,1/2, 2/1, 2/2, 3/1, and 3/2).

IP Multicast Delivery Modes

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

Source Specific Multicast

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

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

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

Multicast Fast Drop

In IP multicast protocols, such as PIM-SM and PIM-DM, every (S,G) or (*,G) route has an incoming interface associated with it. This interface is referred to as the reverse path forwarding interface. In some cases, when a packet arrives on an interface other than the expected RPF interface, the packet must be forwarded to the CPU subsystem software to allow PIM to perform special protocol processing on the packet. One example of this special protocol processing that PIM performs is the PIM Assert protocol.

By default, the Integrated Switching Engine hardware sends all packets that arrive on a non-RPF interface to the CPU subsystem software. However, processing in software is not necessary in many cases, because these non-RPF packets are often not needed by the multicast routing protocols. The problem is that if no action is taken, the non-RPF packets that are sent to the software can overwhelm the CPU.

Instead of installing fast-drop entries, the Cisco device uses Dynamic Buffer Limiting (DBL). This flow-based congestion avoidance mechanism provides active queue management by tracking the queue length for each traffic flow. When the queue length of a flow exceeds its set limit, DBL drops packets. Rate DBL limits the non-rpf traffic to the cpu subsystem so that the CPU is not overwhelmed. The packets are rate limited per flow to the CPU. Because installing fast-drop entries in the CAM is inaccessibly, the number of fast-drop flows that can be handled by the switch need not be limited.

Protocol events, such as a link going down or a change in the unicast routing table, can impact the set of packets that can safely be fast dropped. A packet that was correctly fast dropped before might, after a topology change, need to be forwarded to the CPU subsystem software so that PIM can process it. The CPU subsystem software handles flushing fast-drop entries in response to protocol events so that the PIM code in IOS can process all the necessary RPF failures.

The use of fast-drop entries in the hardware is critical in some common topologies because you may have persistent RPF failures. Without the fast-drop entries, the CPU is exhausted by RPF failed packets that it did not need to process.

Multicast Forwarding Information Base

The Multicast Forwarding Information Base (MFIB) subsystem supports IP multicast routing in the Integrated Switching Engine hardware on Cisco devices. The MFIB logically resides between the IP multicast routing protocols in the CPU subsystem software (PIM, IGMP, MSDP, MBGP, and DVMRP) and the platform-specific code that manages IP multicast routing in hardware. The MFIB translates the routing table information created by the multicast routing protocols into a simplified format that can be efficiently processed and used for forwarding by the Integrated Switching Engine hardware.

To display the information in the multicast routing table, use the **show ip mroute** command. To display the MFIB table information, use the **show ip mfib** command.

The MFIB table contains a set of IP multicast routes. IP multicast routes include (S,G) and (*,G). Each route in the MFIB table can have one or more optional flags associated with it. The route flags indicate how a packet that matches a route should be forwarded. For example, the Internal Copy (IC) flag on an MFIB route indicates that a process on the switch needs to receive a copy of the packet. The following flags can be associated with MFIB routes:

- Internal Copy (IC) flag—Sets on a route when a process on the router needs to receive a copy of all packets matching the specified route.
- Signalling (S) flag—Sets on a route when a process needs to be notified when a packet matching the route is received; the expected behavior is that the protocol code updates the MFIB state in response to receiving a packet on a signalling interface.
- Connected (C) flag—When set on an MFIB route, has the same meaning as the Signaling (S) flag, except that the C flag indicates that only packets sent by directly connected hosts to the route should be signaled to a protocol process.

A route can also have a set of optional flags associated with one or more interfaces. For example, an (S,G) route with the flags on VLAN 1 indicates how packets arriving on VLAN 1 should be handled, and whether packets matching the route should be forwarded onto VLAN 1. The per-interface flags supported in the MFIB include the following:

- Accepting (A)—Sets on the interface that is known in multicast routing as the RPF interface. A packet that arrives on an interface that is marked as Accepting (A) is forwarded to all Forwarding (F) interfaces.
- Forwarding (F)—Used in conjunction with the Accepting (A) flag as described above. The set of Forwarding interfaces that form what is often referred to as the multicast “olist” or output interface list.
- Signaling (S)—Sets on an interface when some multicast routing protocol process in Cisco IOS needs to be notified of packets arriving on that interface.



Note When PIM-SM routing is in use, the MFIB route might include an interface as in this example:

```
PimTunnel [1.2.3.4]
```

It is a virtual interface that the MFIB subsystem creates to indicate that packets are being tunnelled to the specified destination address. A PimTunnel interface cannot be displayed with the normal **show interface** command.

S/M, 224/4

An (S/M, 224/4) entry is created in the MFIB for every multicast-enabled interface. This entry ensures that all packets sent by directly connected neighbors can be register-encapsulated to the PIM-SM RP. Typically, only a small number of packets are forwarded using the (S/M,224/4) route, until the (S,G) route is established by PIM-SM.

For example, on an interface with IP address 10.0.0.1 and netmask 255.0.0.0, a route is created matching all IP multicast packets in which the source address is anything in the class A network 10. This route can be written in conventional subnet/masklength notation as (10/8,224/4). If an interface has multiple assigned IP addresses, then one route is created for each such IP address.

Multicast High Availability

Cisco Catalyst 9400 Series Switches support multicast high availability, which ensures uninterrupted multicast traffic flow if a supervisor engine failure. MFIB states are synced to the standby supervisor engine before a switchover, ensuring NSF availability with a fast convergence upon switchover during a supervisor engine failure.

Multicast HA (SSO / NSF / ISSU) is supported for the PIM Sparse mode and SSM mode; and in Layer 2 for IGMP and MLD Snooping.

Additional References for IP Multicast

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the IP Multicast Routing Commands section of the <i>Command Reference (Catalyst 9400 Series Switches)</i>

Standards and RFCs

Standard/RFC	Title
RFC 1112	<i>Host Extensions for IP Multicasting</i>
RFC 2236	<i>Internet Group Management Protocol, Version 2</i>
RFC 4601	<i>Protocol-Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification</i>



CHAPTER 2

Configuring Basic IP Multicast Routing

- [Restrictions for Basic IP Multicast Routing, on page 17](#)
- [Information About Basic IP Multicast Routing, on page 17](#)
- [How to Configure Basic IP Multicast Routing, on page 19](#)
- [Monitoring and Maintaining Basic IP Multicast Routing, on page 27](#)
- [Configuration Examples for Basic IP Multicast Routing, on page 29](#)
- [Additional References for Basic IP Multicast Routing, on page 30](#)
- [Feature History for Basic IP Multicast Routing, on page 30](#)

Restrictions for Basic IP Multicast Routing

The following are the restrictions for IP multicast routing:

- Packets that have a multicast destination IP address and unicast MAC address are dropped.
- For some multicast groups, when more than 8K mroutes are installed in a system, the network may experience higher traffic losses upon switchover of the HA system. This is due to flushing the old multicast forwarding entries before the new entries are updated. As the number of routes increase, more time is required for the entries to be updated in the MFIB. To reduce the traffic loss in this scenario, you should increase the multicast route-flush timer (using the **ip multicast redundancy routeflush maxtime** command) to a value exceeding the default (30 seconds).

Information About Basic IP Multicast Routing

IP multicasting is an efficient way to use network resources, especially for bandwidth-intensive services such as audio and video. IP multicast routing enables a host (source) to send packets to a group of hosts (receivers) anywhere within the IP network by using a special form of IP address called the IP multicast group address.

The sending host inserts the multicast group address into the IP destination address field of the packet, and IP multicast routers and multilayer devices forward incoming IP multicast packets out all interfaces that lead to members of the multicast group. Any host, regardless of whether it is a member of a group, can send to a group. However, only the members of a group receive the message.

Multicast Forwarding Information Base Overview

The device uses the Multicast Forwarding Information Base (MFIB) architecture and the Multicast Routing Information Base (MRIB) for IP multicast.

The MFIB architecture provides both modularity and separation between the multicast control plane (Protocol Independent Multicast [PIM] and Internet Group Management Protocol [IGMP]) and the multicast forwarding plane (MFIB). This architecture is used in Cisco IOS IPv6 multicast implementations.

MFIB itself is a multicast routing protocol independent forwarding engine; that is, it does not depend on PIM or any other multicast routing protocol. It is responsible for:

- Forwarding multicast packets
- Registering with the MRIB to learn the entry and interface flags set by the control plane
- Handling data-driven events that must be sent to the control plane
- Maintaining counts, rates, and bytes of received, dropped, and forwarded multicast packets

The MRIB is the communication channel between MRIB clients. Examples of MRIB clients are PIM, IGMP, the multicast routing (mroutd) table, and the MFIB.

Default IP Multicast Routing Configuration

This table displays the default IP multicast routing configuration.

Table 2: Default IP Multicast Routing Configuration

Feature	Default Setting
Multicast routing	Disabled on all interfaces.
PIM version	Version 2.
PIM mode	No mode is defined.
PIM stub routing	None configured.
PIM RP address	None configured.
PIM domain border	Disabled.
PIM multicast boundary	None.
Candidate BSRs	Disabled.
Candidate RPs	Disabled.
Shortest-path tree threshold rate	0 kb/s.
PIM router query message interval	30 seconds.

How to Configure Basic IP Multicast Routing

This section provides information about configuring basic IP multicast routing.

Configuring Basic IP Multicast Routing

By default, multicast routing is disabled, and there is no default mode setting.

This procedure is required.

Before you begin

You must configure the PIM version and the PIM mode. The switch populates its multicast routing table and forwards multicast packets it receives from its directly connected LANs according to the mode setting.

In populating the multicast routing table, dense-mode interfaces are always added to the table. Sparse-mode interfaces are added to the table only when periodic join messages are received from downstream devices or when there is a directly connected member on the interface. When forwarding from a LAN, sparse-mode operation occurs if there is an RP known for the group. If so, the packets are encapsulated and sent toward the RP. When no RP is known, the packet is flooded in a dense-mode fashion. The multicast source address must be on the directly connected incoming interface (that is part of the same subnet) of the first-hop router for both PIM dense mode and PIM any-source multicast mode. If the multicast traffic from a specific source is sufficient, the receiver's first-hop router might send join messages toward the source to build a source-based distribution tree.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface interface-id Example: Device(config)# interface gigabitethernet 1/0/1	Specifies the Layer 3 interface on which you want to enable multicast routing, and enters interface configuration mode. The specified interface must be one of the following: <ul style="list-style-type: none">• A routed port—A physical port that has been configured as a Layer 3 port by entering the no switchport interface configuration command. You will also

	Command or Action	Purpose
		<p>need to enable IP PIM sparse-dense-mode on the interface, and join the interface as a statically connected member to an IGMP static group.</p> <ul style="list-style-type: none"> • An SVI—A VLAN interface created by using the interface vlan <i>vlan-id</i> global configuration command. You will also need to enable IP PIM sparse-dense-mode on the VLAN, join the VLAN as a statically connected member to an IGMP static group, and then enable IGMP snooping on the VLAN, the IGMP static group, and physical interface. <p>These interfaces must have IP addresses assigned to them.</p>
Step 4	<p>ip pim {dense-mode sparse-mode sparse-dense-mode}</p> <p>Example:</p> <pre>Device(config-if)# ip pim sparse-dense-mode</pre>	<p>Enables a PIM mode on the interface.</p> <p>By default, no mode is configured.</p> <p>The keywords have these meanings:</p> <ul style="list-style-type: none"> • dense-mode—Enables dense mode of operation. • sparse-mode—Enables sparse mode of operation. If you configure sparse mode, you must also configure an RP. • sparse-dense-mode—Causes the interface to be treated in the mode in which the group belongs. Sparse-dense mode is the recommended setting. <p>Note To disable PIM on an interface, use the no ip pim interface configuration command.</p>
Step 5	<p>end</p> <p>Example:</p> <pre>Device(config-if)# end</pre>	Returns to privileged EXEC mode.
Step 6	<p>show running-config</p> <p>Example:</p> <pre>Device# show running-config</pre>	Verifies your entries.

	Command or Action	Purpose
Step 7	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Configuring IP Multicast Forwarding

You can use the following procedure to configure IPv4 Multicast Forwarding Information Base (MFIB) interrupt-level IP multicast forwarding of incoming packets or outgoing packets on the device.



Note After you have enabled IP multicast routing by using the **ip multicast-routing** command, IPv4 multicast forwarding is enabled. Because IPv4 multicast forwarding is enabled by default, you can use the **no** form of the **ip mfib** command to disable IPv4 multicast forwarding.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ip mfib Example: <pre>Device(config)# ip mfib</pre>	Enables IP multicast forwarding.
Step 4	exit Example: <pre>Device(config)# exit</pre>	Returns to privileged EXEC mode.
Step 5	show running-config Example: <pre>Device# show running-config</pre>	Verifies your entries.
Step 6	copy running-config startup-config Example:	(Optional) Saves your entries in the configuration file.

	Command or Action	Purpose
	Device# copy running-config startup-config	

Configuring a Static Multicast Route (mroute)

- Static mroutes are used to calculate RPF information, not to forward traffic.
- Static mroutes cannot be redistributed.

Static mroutes are strictly local to the device on which they are defined. Because Protocol Independent Multicast (PIM) does not have its own routing protocol, there is no mechanism to distribute static mroutes throughout the network. Consequently, the administration of static mroutes tends to be more complicated than the administration of unicast static routes.

When static mroutes are configured, they are stored on the device in a separate table referred to as the static mroute table. When configured, the **ip mroute** command enters a static mroute into the static mroute table for the source address or source address range specified for the source-address and mask arguments. Sources that match the source address or that fall in the source address range specified for the source-address argument will RPF to either the interface associated with the IP address specified for the *rpf-address* argument or the local interface on the device specified for the *interface-type* and *interface-number* arguments. If an IP address is specified for the *rpf-address* argument, a recursive lookup is done from the unicast routing table on this address to find the directly connected neighbor.

If there are multiple static mroutes configured, the device performs a longest-match lookup of the mroute table. When the mroute with the longest match (of the source-address) is found, the search terminates and the information in the matching static mroute is used. The order in which the static mroutes are configured is not important.

The administrative distance of an mroute may be specified for the optional distance argument. If a value is not specified for the distance argument, the distance of the mroute defaults to zero. If the static mroute has the same distance as another RPF source, the static mroute will take precedence. There are only two exceptions to this rule: directly connected routes and the default unicast route.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip mroute [vrf vrf-name] <i>source-address mask</i> [{ fallback-lookup {global vrf vrf-name} [protocol] {rpf-address interface-type interface-number}}] [distance] Example:	The source IP address 10.1.1.1 is configured to be reachable through the interface associated with IP address 10.2.2.2.

	Command or Action	Purpose
	Device(config)# ip mroute 10.1.1.1 255.255.255.255 10.2.2.2	
Step 4	exit Example: Device(config)# exit	Returns to privileged EXEC mode.
Step 5	show running-config Example: Device# show running-config	(Optional) Verifies your entries.
Step 6	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring Optional IP Multicast Routing Features

This section provides information about configuring optional IP multicast routing features.

Defining the IP Multicast Boundary

You define a multicast boundary to prevent Auto-RP messages from entering the PIM domain. You create an access list to deny packets destined for 224.0.1.39 and 224.0.1.40, which carry Auto-RP information.

This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	access-list access-list-number deny source [source-wildcard] Example: Device(config)# access-list 12 deny 224.0.1.39 access-list 12 deny 224.0.1.40	Creates a standard access list, repeating the command as many times as necessary. • For <i>access-list-number</i> , the range is 1 to 99. • The deny keyword denies access if the conditions are matched.

	Command or Action	Purpose
		<ul style="list-style-type: none"> For <i>source</i>, enter multicast addresses 224.0.1.39 and 224.0.1.40, which carry Auto-RP information. (Optional) For <i>source-wildcard</i>, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore. <p>The access list is always terminated by an implicit deny statement for everything.</p>
Step 4	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet 1/0/1	<p>Specifies the interface to be configured, and enters interface configuration mode.</p> <p>The specified interface must be one of the following:</p> <ul style="list-style-type: none"> A routed port—A physical port that has been configured as a Layer 3 port by entering the no switchport interface configuration command. An SVI—A VLAN interface created by using the interface vlan <i>vlan-id</i> global configuration command. <p>These interfaces must have IP addresses assigned to them.</p>
Step 5	ip multicast boundary <i>access-list-number</i> Example: Device(config-if)# ip multicast boundary 12	Configures the boundary, specifying the access list you created in Step 2.
Step 6	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 7	show running-config Example: Device# show running-config	Verifies your entries.
Step 8	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring sdr Listener Support

This section provides information about configuring sdr listener support.

Enabling sdr Listener Support

By default, the device does not listen to session directory advertisements. This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet 1/0/1	Specifies the interface to be enabled for sdr, and enters interface configuration mode. The specified interface must be one of the following: <ul style="list-style-type: none"> • A routed port—A physical port that has been configured as a Layer 3 port by entering the no switchport interface configuration command. You will also need to enable IP PIM sparse-dense-mode on the interface, and join the interface as a statically connected member to an IGMP static group. For a configuration example, see Example: Interface Configuration as a Routed Port, on page 88. • An SVI—A VLAN interface created by using the interface vlan <i>vlan-id</i> global configuration command. You will also need to enable IP PIM sparse-dense-mode on the VLAN, join the VLAN as a statically connected member to an IGMP static group, and then enable IGMP snooping on the VLAN, the IGMP static group, and physical interface. For a configuration example, see Example: Interface Configuration as an SVI, on page 88. These interfaces must have IP addresses assigned to them.

Limiting How Long an sdr Cache Entry Exists

	Command or Action	Purpose
Step 4	ip sap listen Example: Device(config-if)# ip sap listen	Enables the device software to listen to session directory announcements.
Step 5	end Example: Device(config-if)# end	Returns to privileged EXEC mode.
Step 6	show running-config Example: Device# show running-config	Verifies your entries.
Step 7	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Limiting How Long an sdr Cache Entry Exists

By default, entries are never deleted from the sdr cache. You can limit how long the entry remains active so that if a source stops advertising SAP information, old advertisements are not unnecessarily kept.

This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip sap cache-timeout <i>minutes</i> Example: Device(config)# ip sap cache-timeout 30	Limits how long a Session Announcement Protocol (SAP) cache entry stays active in the cache. By default, entries are never deleted from the cache. For <i>minutes</i> , the range is 1 to 1440 minutes (24 hours).
Step 4	end Example:	Returns to privileged EXEC mode.

	Command or Action	Purpose
	Device(config) # end	
Step 5	show running-config Example: Device# show running-config	Verifies your entries.
Step 6	show ip sap Example: Device# show ip sap	Displays the SAP cache.
Step 7	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Monitoring and Maintaining Basic IP Multicast Routing

Clearing Caches, Tables, and Databases

You can remove all contents of a particular cache, table, or database. Clearing a cache, table, or database might be necessary when the contents of the particular structure are or suspected to be invalid.

You can use any of the privileged EXEC commands in the following table to clear IP multicast caches, tables, and databases.

Table 3: Commands for Clearing Caches, Tables, and Databases

Command	Purpose
clear ip igmp group {group [hostname IP address] vrf name group [hostname IP address] }	Deletes entries from the
clear ip mroute { * [hostname IP address] vrf name group [hostname IP address] }	Deletes entries from the
clear ip sap [group-address "session-name"]	Deletes the Session Di cache entry.

Displaying System and Network Statistics

You can display specific statistics, such as the contents of IP routing tables, caches, and databases.



Note This release does not support per-route statistics.

You can display information to learn resource usage and solve network problems. You can also display information about node reachability and discover the routing path that packets of your device are taking through the network.

You can use any of the privileged EXEC commands in the following table to display various routing statistics.

Table 4: Commands for Displaying System and Network Statistics

Command	Purpose
ping [<i>group-name</i> <i>group-address</i>]	Sends an ICMP Echo Request to a multicast
show ip igmp filter	Displays IGMP filter information.
show ip igmp groups [<i>group-name</i> <i>group-address</i> <i>type-number</i>]	Displays the multicast groups that are direct
show ip igmp interface [<i>type number</i>]	Displays multicast-related information about
show ip igmp profile [<i>profile_number</i>]	Displays IGMP profile information.
show ip igmp ssm-mapping [<i>hostname/IP address</i>]	Displays IGMP SSM mapping information.
show ip igmp static-group { <i>class-map</i> [<i>interface</i> [<i>type</i>]]	Displays static group information.
show ip igmp membership [<i>name/group address</i> all tracked]	Displays IGMP membership information fo
show ip igmp vrf	Displays the selected VPN Routing/Forward
show ip mfib [<i>type number</i>]	Displays the IP multicast forwarding inform
show ip mrrib { <i>client</i> <i>route</i> <i>vrf</i> }	Displays the multicast routing information b
show ip mrm { <i>interface</i> <i>manager</i> <i>status-report</i> }	Displays the IP multicast routing monitor in
show ip mroute [<i>group-name</i> <i>group-address</i>] [<i>source</i>] [<i>count</i> <i>interface</i> <i>proxy</i> <i>pruned</i> <i>summary</i> <i>verbose</i>]	Displays the contents of the IP multicast rou
show ip msdp { <i>count</i> <i>peer</i> <i>rpf-peer</i> <i>sa-cache</i> <i>summary</i> <i>vrf</i> }	Displays the Multicast Source Discovery Pr
show ip multicast [<i>interface</i> <i>limit</i> <i>mpls</i> <i>redundancy</i> <i>vrf</i>]	Displays global multicast information.
show ip pim all-vrfs { <i>tunnel</i> }	Display all VRFs.
show ip pim autorp	Display global auto-RP information.
show ip pim boundary [<i>type number</i>]	Displays boundary information.
show ip pim bsr-router	Display bootstrap router information (versio
show ip pim interface [<i>type number</i>] [<i>count</i> <i>detail</i> <i>df</i> <i>stats</i>]	Displays information about interfaces config
show ip pim neighbor [<i>type number</i>]	Lists the PIM neighbors discovered by the d
show ip pim mdt [<i>bgp</i>]	Displays multicast tunnel information.
show ip pim rp [<i>group-name</i> <i>group-address</i>]	Displays the RP routers associated with a spa images.

Command	Purpose
show ip pim rp-hash [<i>group-name</i> <i>group-address</i>]	Displays the RP to be chosen based upon
show ip pim tunnel [<i>tunnel</i> <i>verbose</i>]	Displays the registered tunnels.
show ip pim vrf <i>name</i>	Displays VPN routing and forwarding in
show ip rpf { <i>source-address</i> <i>name</i> }	Displays how the device is doing Reverse Path Forwarding (RPF) (using the routing table, or static mroutes). Command parameters include: <ul style="list-style-type: none"> • <i>Host name</i> or <i>IP address</i>—IP name • Select—Group-based VRF select in • vrf—Selects VPN Routing/Forwarding
show ip sap [<i>group</i> “ <i>session-name</i> ” detail]	Displays the Session Announcement Protocol (SAP) information. Command parameters include: <ul style="list-style-type: none"> • <i>A.B.C.D</i>—IP group address. • <i>WORD</i>—Session name (in double quotes) • detail—Session details.

Configuration Examples for Basic IP Multicast Routing

This section provides configuration examples for Basic IP Multicast Routing.

Example: Configuring an IP Multicast Boundary

This example shows how to set up a boundary for all administratively-scoped addresses:

```
Device(config)# access-list 1 deny 239.0.0.0 0.255.255.255
Device(config)# access-list 1 permit 224.0.0.0 15.255.255.255
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ip multicast boundary 1
```

Example: Responding to minfo Requests

The software answers minfo requests sent by mrouterd systems and Cisco routers and multilayer devices. The software returns information about neighbors through DVMRP tunnels and all the routed interfaces. This information includes the metric (always set to 1), the configured TTL threshold, the status of the interface, and various flags. You can also use the **minfo** privileged EXEC command to query the router or device itself, as in this example:

```
Device# minfo
```

```

171.69.214.27 (mm1-7kd.cisco.com) [version cisco 11.1] [flags: PMS]:
171.69.214.27 -> 171.69.214.26 (mm1-r7kb.cisco.com) [1/0/pim/querier]
171.69.214.27 -> 171.69.214.25 (mm1-45a.cisco.com) [1/0/pim/querier]
171.69.214.33 -> 171.69.214.34 (mm1-45c.cisco.com) [1/0/pim]
171.69.214.137 -> 0.0.0.0 [1/0/pim/querier/down/leaf]
171.69.214.203 -> 0.0.0.0 [1/0/pim/querier/down/leaf]
171.69.214.18 -> 171.69.214.20 (mm1-45e.cisco.com) [1/0/pim]
171.69.214.18 -> 171.69.214.19 (mm1-45c.cisco.com) [1/0/pim]
171.69.214.18 -> 171.69.214.17 (mm1-45a.cisco.com) [1/0/pim]

```

Additional References for Basic IP Multicast Routing

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the IP Multicast Routing Commands section of the <i>Command Reference (Catalyst 9400 Series Switches)</i>

Feature History for Basic IP Multicast Routing

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	IP Multicast Routing	IP Multicast is an efficient way to use network resources, especially for bandwidth-intensive services such as audio and video. IP multicast routing enables a host (source) to send packets to a group of hosts (receivers) anywhere within the IP network by using a special form of IP address called the IP multicast group address.

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>.



CHAPTER 3

Configuring Multicast Routing over GRE Tunnel

- [Prerequisites for Configuring Multicast Routing over GRE Tunnel, on page 31](#)
- [Restrictions for Configuring Multicast Routing over GRE Tunnel, on page 31](#)
- [Information About Multicast Routing over GRE Tunnel, on page 32](#)
- [How to Configure Multicast Routing over GRE Tunnel, on page 32](#)
- [Additional References for Multicast Routing over GRE Tunnel, on page 35](#)
- [Feature History for Multicast Routing over GRE Tunnel, on page 35](#)

Prerequisites for Configuring Multicast Routing over GRE Tunnel

Before configuring multicast routing over GRE, you should be familiar with the concepts of IP Multicast Routing Technology and GRE Tunneling.

Restrictions for Configuring Multicast Routing over GRE Tunnel

The following are the restrictions for configuring multicast routing over GRE tunnel:

- IPv6 multicast over GRE tunnel is not supported.
- The total number of supported multicast routes (mroutes) is 32000, across all tunnels.
Use the formula $8000 / (((\text{Number of tunnels}) / 4) + 1)$ to derive the number of mroutes.
- Bidirectional PIM is not supported.
- Multicast routing should be configured on the first hop router (FHR), the rendezvous point (RP) and the last hop router (LHR) to support multicast over the GRE tunnel.
- On Catalyst 9000 Series Switches, the tunnel source can be a loopback, physical, or L3 EtherChannel interface.
- No feature interactions such as IPSec, ACL, Tunnel counters, Crypto support, Fragmentation, Cisco Discovery Protocol (CDP), QoS, GRE keepalive, Multipoint GRE, etc. are supported on the GRE Tunnel.

Information About Multicast Routing over GRE Tunnel

This chapter describes how to configure a Generic Route Encapsulation (GRE) tunnel to tunnel IP multicast packets between non-IP multicast areas. The benefit is that IP multicast traffic can be sent from a source to a multicast group, over an area where IP multicast is not supported. Multicast Routing over GRE Tunnel supports sparse mode and pim-ssm mode; and supports static RP and auto-RP. See Rendezvous Point and Auto-RP for information on configuring static RP and auto-RP.

Benefits of Tunneling to Connect Non-IP Multicast Areas

- If the path between a source and a group member (destination) does not support IP multicast, a tunnel between them can transport IP multicast packets.

How to Configure Multicast Routing over GRE Tunnel

This section provides steps for configuring multicast routing over GRE tunnel.

Configuring a GRE Tunnel to Connect Non-IP Multicast Areas

You can configure a GRE tunnel to transport IP multicast packets between a source and destination that are connected by a medium that does not support multicast routing.

Procedure

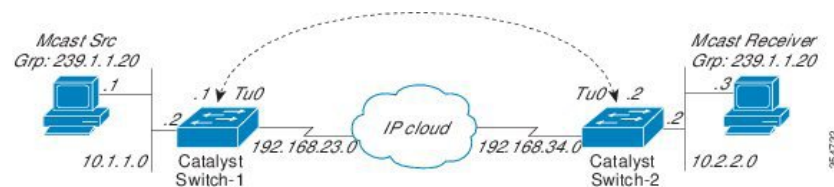
	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ip multicast-routing Example: <pre>Device(config)# ip multicast-routing</pre>	Enables IP multicast routing.
Step 4	interface tunnel <i>number</i> Example: <pre>Device(config)# interface tunnel 0</pre>	Enters tunnel interface configuration mode.

	Command or Action	Purpose
Step 5	ip address ip_address subnet_mask Example: <pre>Device(config-if)# ip address 192.168.24.1 255.255.255.252</pre>	Configures IP address and IP subnet.
Step 6	ip pim sparse-mode Example: <pre>Device(config-if)# ip pim sparse-mode</pre>	Enables sparse mode of operation of Protocol Independent Multicast (PIM) on the tunnel interface with one of the following mode of operation:
Step 7	tunnel source { ip-address interface-name } Example: <pre>Device(config-if)# tunnel source 100.1.1.1</pre>	Configures the tunnel source.
Step 8	tunnel destination { hostname ip-address } Example: <pre>Device(config-if)# tunnel destination 100.1.1.3</pre>	Configures the tunnel destination.
Step 9	end Example: <pre>Device(config-if)# end</pre>	Ends the current configuration session and returns to privileged EXEC mode.
Step 10	show interface type number Example: <pre>Device# show interface tunnel 0</pre>	Displays tunnel interface information.

Tunneling to Connect Non-IP Multicast Areas Example

The following example shows multicast-routing between a Catalyst switch through a GRE tunnel.

Figure 7: Tunnel Connecting Non-IP Multicast Areas



In the figure above, the multicast source (10.1.1.1) is connected to Catalyst Switch-1 and is configured for multicast group 239.1.1.20. The multicast receiver (10.2.2.3) is connected to Catalyst Switch-2 and is configured to receive multicast packets for group 239.1.1.20. Separating Switch-1 and Switch-2 is an IP cloud, which is not configured for multicast routing.

A GRE tunnel is configured between Switch-1 to Switch-2 sourced with their loopback interfaces. Multicast-routing is enabled on Switch-1 and Switch-2. The **ip pim sparse-mode** command is configured on tunnel interfaces to support PIM in the sparse mode. Sparse mode configuration on the tunnel interfaces allows sparse-mode packets to be forwarded over the tunnel depending on rendezvous point (RP) configuration for the group.

Switch-1 Configuration:

```
Device(config)# ip multicast-routing
Device(config)# interface Loopback0 //Tunnel source interface
Device(config-if)# ip address 2.2.2.2 255.255.255.255

Device(config)# interface Tunnel 10 //Tunnel interface configured for PIM
traffic
Device(config-if)# ip address 192.168.24.1 255.255.255.252
Device(config-if)# ip pim sparse-mode
Device(config-if)# ip nhrp map 192.168.24.3 4.4.4.4 //NHRP may optionally be
configured to dynamically discover tunnel end points.
Device(config-if)# ip nhrp map multicast 4.4.4.4
Device(config-if)# ip nhrp network-id 1
Device(config-if)# ip nhrp nhs 192.168.24.3
Device(config-if)# tunnel source Loopback0
Device(config-if)# tunnel destination 4.4.4.4

Device(config)# interface GigabitEthernet 0/0/0 //Source interface
Device(config-if)# ip address 10.1.1.2 255.255.255.0
Device(config-if)# ip pim sparse-mode
```

Switch-2 Configuration:

```
Device(config)# ip multicast-routing
Device(config)# interface Loopback0 //Tunnel source interface
Device(config-if)# ip address 4.4.4.4 255.255.255.255

Device(config)# interface Tunnel 10 //Tunnel interface configured for PIM
traffic
Device(config-if)# ip address 192.168.24.2 255.255.255.252
Device(config-if)# ip nhrp map 192.168.24.4 2.2.2.2 //NHRP may optionally be
configured to dynamically discover tunnel end points.
Device(config-if)# ip nhrp map multicast 2.2.2.2
Device(config-if)# ip nhrp network-id 1
Device(config-if)# ip nhrp nhs 192.168.24.4
Device(config-if)# ip pim sparse-mode
Device(config-if)# tunnel source Loopback0
Device(config-if)# tunnel destination 2.2.2.2

Device(config)# interface GigabitEthernet 0/0/0 //Receiver interface
Device(config-if)# ip address 10.2.2.2 255.255.255.0
Device(config-if)# ip pim sparse-mode
```


Additional References for Multicast Routing over GRE Tunnel

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the IP Multicast Routing Commands section of the <i>Command Reference (Catalyst 9400 Series Switches)</i> .

Feature History for Multicast Routing over GRE Tunnel

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	Multicast Routing over GRE Tunnel	Multicast routing over GRE tunnel allows IP multicast traffic to be sent from a source to a multicast group, over an area where IP multicast is not supported

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.



CHAPTER 4

Configuring IGMP

- [Prerequisites for IGMP and IGMP Snooping, on page 37](#)
- [Restrictions for IGMP and IGMP Snooping, on page 38](#)
- [Information about IGMP, on page 39](#)
- [How to Configure IGMP, on page 51](#)
- [How to Configure IGMP Snooping, on page 67](#)
- [Monitoring IGMP, on page 83](#)
- [Configuration Examples for IGMP, on page 86](#)
- [Additional References for IGMP, on page 91](#)
- [Feature History for IGMP, on page 91](#)

Prerequisites for IGMP and IGMP Snooping

Prerequisites for IGMP Snooping

Observe these guidelines when configuring the IGMP snooping querier:

- Configure the VLAN in global configuration mode.
- Configure an IP address on the VLAN interface. When enabled, the IGMP snooping querier uses the IP address as the query source address.
- If there is no IP address configured on the VLAN interface, the IGMP snooping querier tries to use the configured global IP address for the IGMP querier. If there is no global IP address specified, the IGMP querier tries to use the VLAN device virtual interface (SVI) IP address (if one exists). If there is no SVI IP address, the device uses the first available IP address configured on the device. The first IP address available appears in the output of the **show ip interface** privileged EXEC command. The IGMP snooping querier does not generate an IGMP general query if it cannot find an available IP address on the device.
- The IGMP snooping querier supports IGMP Versions 1 and 2.
- When administratively enabled, the IGMP snooping querier moves to the nonquerier state if it detects the presence of a multicast router in the network.
- When it is administratively enabled, the IGMP snooping querier moves to the operationally disabled state under these conditions:
 - IGMP snooping is disabled in the VLAN.

- PIM is enabled on the SVI of the corresponding VLAN.

Restrictions for IGMP and IGMP Snooping

Restrictions for Configuring IGMP

The following are the restrictions for configuring IGMP:

- The device supports IGMP Versions 1, 2, and 3.



Note For IGMP Version 3, only IGMP Version 3 BISS (Basic IGMPv3 Snooping Support) is supported.

- IGMP Version 3 uses new membership report messages that might not be correctly recognized by older IGMP snooping devices.
- IGMPv3 can operate with both ISM and SSM. In ISM, both exclude and include mode reports are applicable. In SSM, only include mode reports are accepted by the last-hop router. Exclude mode reports are ignored.
- Use ACLs to designate a specified port only as a multicast host port and not as a multicast router port. Multicast router control-packets received on this port are dropped.

Restrictions for IGMP Snooping

The following are the restrictions for IGMP snooping:

- The device supports IGMPv3 snooping based only on the destination multicast IP address. It does not support snooping based on a source IP address or proxy report.
- IGMPv3 join and leave messages are not supported on the devices running IGMP filtering or Multicast VLAN registration (MVR).
- IGMP report suppression is supported only when the multicast query has IGMPv1 and IGMPv2 reports. This feature is not supported when the query includes IGMPv3 reports.
- The IGMP configurable leave time is only supported on hosts running IGMP Version 2. IGMP version 2 is the default version for the device.

The actual leave latency in the network is usually the configured leave time. However, the leave time might vary around the configured time, depending on real-time CPU load conditions, network delays and the amount of traffic sent through the interface.

- The IGMP throttling action restriction can be applied only to Layer 2 ports. You can use **ip igmp max-groups action replace** interface configuration command on a logical EtherChannel interface but cannot use it on ports that belong to an EtherChannel port group.

When the maximum group limitation is set to the default (no maximum), entering the **ip igmp max-groups action {deny | replace}** command has no effect.

If you configure the throttling action and set the maximum group limitation after an interface has added multicast entries to the forwarding table, the forwarding-table entries are either aged out or removed, depending on the throttling action.

Information about IGMP

Role of the Internet Group Management Protocol

IGMP is used to dynamically register individual hosts in a multicast group on a particular LAN. Enabling PIM on an interface also enables IGMP. IGMP provides a means to automatically control and limit the flow of multicast traffic throughout your network with the use of special multicast queriers and hosts.

- A querier is a network device, such as a router, that sends query messages to discover which network devices are members of a given multicast group.
- A host is a receiver, including routers, that sends report messages (in response to query messages) to inform the querier of a host membership. Hosts use IGMP messages to join and leave multicast groups.

Hosts identify group memberships by sending IGMP messages to their local multicast device. Under IGMP, devices listen to IGMP messages and periodically send out queries to discover which groups are active or inactive on a particular subnet.

IGMP Multicast Addresses

IP multicast traffic uses group addresses, which are Class D IP addresses. The high-order four bits of a Class D address are 1110. Therefore, host group addresses can be in the range 224.0.0.0 to 239.255.255.255.

Multicast addresses in the range 224.0.0.0 to 224.0.0.255 are reserved for use by routing protocols and other network control traffic. The address 224.0.0.0 is guaranteed not to be assigned to any group.

IGMP packets are transmitted using IP multicast group addresses as follows:

- IGMP general queries are destined to the address 224.0.0.1 (all systems on a subnet).
- IGMP group-specific queries are destined to the group IP address for which the device is querying.
- IGMP group membership reports are destined to the group IP address for which the device is reporting.
- IGMPv2 leave-group messages are destined to the address 224.0.0.2 (all devices on a subnet).
- IGMPv3 membership reports are destined to the address 224.0.0.22; all IGMPv3-capable multicast devices must listen to this address.

IGMP Versions

The device supports IGMP version 1, IGMP version 2, and IGMP version 3. These versions are interoperable on the device. For example, if IGMP snooping is enabled and the querier's version is IGMPv2, and the device receives an IGMPv3 report from a host, then the device can forward the IGMPv3 report to the multicast router.

An IGMPv3 device can receive messages from and forward messages to a device running the Source Specific Multicast (SSM) feature.

IGMP Version 1

IGMP version 1 (IGMPv1) primarily uses a query-response model that enables the multicast router and multilayer device to find which multicast groups are active (have one or more hosts interested in a multicast group) on the local subnet. IGMPv1 has other processes that enable a host to join and leave a multicast group. For more information, see RFC 1112.

IGMP Version 2

IGMPv2 extends IGMP functionality by providing such features as the IGMP leave process to reduce leave latency, group-specific queries, and an explicit maximum query response time. IGMPv2 also adds the capability for routers to elect the IGMP querier without depending on the multicast protocol to perform this task. For more information, see RFC 2236.



Note IGMP version 2 is the default version for the device.

IGMP Version 3

The device supports IGMP version 3.

An IGMPv3 device supports Basic IGMPv3 Snooping Support (BISS), which includes support for the snooping features on IGMPv1 and IGMPv2 switches and for IGMPv3 membership report messages. BISS constrains the flooding of multicast traffic when your network includes IGMPv3 hosts. It constrains traffic to approximately the same set of ports as the IGMP snooping feature on IGMPv2 or IGMPv1 hosts.

An IGMPv3 device can receive messages from and forward messages to a device running the Source Specific Multicast (SSM) feature.

IGMPv3 Host Signaling

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

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

IGMP Versions Differences

There are three versions of IGMP, as defined by Request for Comments (RFC) documents of the Internet Engineering Task Force (IETF). IGMPv2 improves over IGMPv1 by adding the ability for a host to signal desire to leave a multicast group and IGMPv3 improves over IGMPv2 mainly by adding the ability to listen to multicast originating from a set of source IP addresses only.

Table 5: IGMP Versions

IGMP Version	Description
IGMPv1	Provides the basic query-response mechanism that allows the multicast device to determine which multicast groups are active and other processes that enable hosts to join and leave a multicast group. RFC 1112 defines the IGMPv1 host extensions for IP multicasting.
IGMPv2	Extends IGMP, allowing such capabilities as the IGMP leave process, group-specific queries, and an explicit maximum response time field. IGMPv2 also adds the capability for devices to elect the IGMP querier without dependence on the multicast protocol to perform this task. RFC 2236 defines IGMPv2.
IGMPv3	Provides for source filtering, which enables a multicast receiver host to signal to a device which groups it wants to receive multicast traffic from, and from which sources this traffic is expected. In addition, IGMPv3 supports the link local address 224.0.0.22, which is the destination IP address for IGMPv3 membership reports; all IGMPv3-capable multicast devices must listen to this address. RFC 3376 defines IGMPv3.



Note By default, enabling a PIM on an interface enables IGMPv2 on that device. IGMPv2 was designed to be as backward compatible with IGMPv1 as possible. To accomplish this backward compatibility, RFC 2236 defined special interoperability rules. If your network contains legacy IGMPv1 hosts, you should be familiar with these operability rules. For more information about IGMPv1 and IGMPv2 interoperability, see RFC 2236, Internet Group Management Protocol, Version 2 .

Devices That Run IGMPv1

IGMPv1 devices send IGMP queries to the “all-hosts” multicast address of 224.0.0.1 to solicit multicast groups with active multicast receivers. The multicast receivers also can send IGMP reports to the device to notify it that they are interested in receiving a particular multicast stream. Hosts can send the report asynchronously or in response to the IGMP queries sent by the device. If more than one multicast receiver exists for the same multicast group, only one of these hosts sends an IGMP report message; the other hosts suppress their report messages.

In IGMPv1, there is no election of an IGMP querier. If more than one device on the segment exists, all the devices send periodic IGMP queries. IGMPv1 has no special mechanism by which the hosts can leave the group. If the hosts are no longer interested in receiving multicast packets for a particular group, they simply do not reply to the IGMP query packets sent from the device. The device continues sending query packets. If the device does not hear a response in three IGMP queries, the group times out and the device stops sending multicast packets on the segment for the group. If the host later wants to receive multicast packets after the timeout period, the host simply sends a new IGMP join to the device, and the device begins to forward the multicast packet again.

If there are multiple devices on a LAN, a designated router (DR) must be elected to avoid duplicating multicast traffic for connected hosts. PIM devices follow an election process to select a DR. The PIM device with the highest IP address becomes the DR.

The DR is responsible for the following tasks:

- Sending PIM register and PIM Join and Prune messages toward the rendezvous point (RP) to inform it about host group membership.
- Sending IGMP host-query messages.
- Sending host-query messages by default every 60 seconds in order to keep the IGMP overhead on hosts and networks very low.

Devices That Run IGMPv2

IGMPv2 improves the query messaging capabilities of IGMPv1.

The query and membership report messages in IGMPv2 are identical to the IGMPv1 messages with two exceptions:

- IGMPv2 query messages are broken into two categories: general queries (identical to IGMPv1 queries) and group-specific queries.
- IGMPv1 membership reports and IGMPv2 membership reports have different IGMP type codes.

IGMPv2 also enhances IGMP by providing support for the following capabilities:

- Querier election process--Provides the capability for IGMPv2 devices to elect the IGMP querier without having to rely on the multicast routing protocol to perform the process.
- Maximum Response Time field--A new field in query messages permits the IGMP querier to specify the maximum query-response time. This field permits the tuning of the query-response process to control response burstiness and to fine-tune leave latencies.
- Group-Specific Query messages--Permits the IGMP querier to perform the query operation on a specific group instead of all groups.
- Leave-Group messages--Provides hosts with a method of notifying devices on the network that they wish to leave the group.

Unlike IGMPv1, in which the DR and the IGMP querier are typically the same device, in IGMPv2 the two functions are decoupled. The DR and the IGMP querier are selected based on different criteria and may be different devices on the same subnet. The DR is the device with the highest IP address on the subnet, whereas the IGMP querier is the device with the lowest IP address.

Query messages are used to elect the IGMP querier as follows:

1. When IGMPv2 devices start, they each multicast a general query message to the all-systems group address of 224.0.0.1 with their interface address in the source IP address field of the message.
2. When an IGMPv2 device receives a general query message, the device compares the source IP address in the message with its own interface address. The device with the lowest IP address on the subnet is elected the IGMP querier.
3. All devices (excluding the querier) start the query timer, which is reset whenever a general query message is received from the IGMP querier. If the query timer expires, it is assumed that the IGMP querier has gone down, and the election process is performed again to elect a new IGMP querier.

By default, the timer is two times the query interval.

Devices Running IGMPv3

IGMPv3 adds support for source filtering, which enables a multicast receiver host to signal to a device which groups it wants to receive multicast traffic from, and from which sources this traffic is expected. This membership information enables the software to forward traffic only from those sources from which receivers requested the traffic.

IGMPv3 supports applications that explicitly signal sources from which they want to receive traffic. With IGMPv3, receivers signal membership to a multicast group in the following two modes:

- **INCLUDE mode**--In this mode, the receiver announces membership to a group and provides a list of IP addresses (the INCLUDE list) from which it wants to receive traffic.
- **EXCLUDE mode**--In this mode, the receiver announces membership to a group and provides a list of IP addresses (the EXCLUDE list) from which it does not want to receive traffic. In other words, the host wants to receive traffic only from sources whose IP addresses are not listed in the EXCLUDE list. To receive traffic from all sources, like in the case of the Internet Standard Multicast (ISM) service model, a host expresses EXCLUDE mode membership with an empty EXCLUDE list.

IGMPv3 is the industry-designated standard protocol for hosts to signal channel subscriptions in an SSM network environment. For SSM to rely on IGMPv3, IGMPv3 must be available in the network stack portion of the operating systems running on the last hop devices and hosts and be used by the applications running on those hosts.

In IGMPv3, hosts send their membership reports to 224.0.0.22; all IGMPv3 devices, therefore, must listen to this address. Hosts, however, do not listen or respond to 224.0.0.22; they only send their reports to that address. In addition, in IGMPv3, there is no membership report suppression because IGMPv3 hosts do not listen to the reports sent by other hosts. Therefore, when a general query is sent out, all hosts on the wire respond.

IGMP Join and Leave Process

IGMP Join Process

When a host wants to join a multicast group, the host sends one or more unsolicited membership reports for the multicast group it wants to join. The IGMP join process is the same for IGMPv1 and IGMPv2 hosts.

In IGMPv3, the join process for hosts proceeds as follows:

- When a host wants to join a group, it sends an IGMPv3 membership report to 224.0.0.22 with an empty EXCLUDE list.
- When a host wants to join a specific channel, it sends an IGMPv3 membership report to 224.0.0.22 with the address of the specific source included in the INCLUDE list.
- When a host wants to join a group excluding particular sources, it sends an IGMPv3 membership report to 224.0.0.22 excluding those sources in the EXCLUDE list.



Note If some IGMPv3 hosts on a LAN wish to exclude a source and others wish to include the source, then the device will send traffic for the source on the LAN (that is, inclusion trumps exclusion in this situation).

IGMP Leave Process

The method that hosts use to leave a group varies depending on the version of IGMP in operation.

IGMPv1 Leave Process

There is no leave-group message in IGMPv1 to notify the devices on the subnet that a host no longer wants to receive the multicast traffic from a specific group. The host simply stops processing traffic for the multicast group and ceases responding to IGMP queries with IGMP membership reports for the group. As a result, the only way IGMPv1 devices know that there are no longer any active receivers for a particular multicast group on a subnet is when the devices stop receiving membership reports. To facilitate this process, IGMPv1 devices associate a countdown timer with an IGMP group on a subnet. When a membership report is received for the group on the subnet, the timer is reset. For IGMPv1 devices, this timeout interval is typically three times the query interval (3 minutes). This timeout interval means that the device may continue to forward multicast traffic onto the subnet for up to 3 minutes after all hosts have left the multicast group.

IGMPv2 Leave Process

IGMPv2 incorporates a leave-group message that provides the means for a host to indicate that it wishes to stop receiving multicast traffic for a specific group. When an IGMPv2 host leaves a multicast group, if it was the last host to respond to a query with a membership report for that group, it sends a leave-group message to the all-devices multicast group (224.0.0.2).

IGMPv3 Leave Process

IGMPv3 enhances the leave process by introducing the capability for a host to stop receiving traffic from a particular group, source, or channel in IGMP by including or excluding sources, groups, or channels in IGMPv3 membership reports.

IGMP Snooping

Layer 2 can use IGMP snooping to constrain the flooding of multicast traffic by dynamically configuring Layer 2 interfaces so that multicast traffic is forwarded to only those interfaces associated with IP multicast devices. As the name implies, IGMP snooping requires the LAN device to snoop on the IGMP transmissions between the host and the router and to keep track of multicast groups and member ports. When the device receives an IGMP report from a host for a particular multicast group, the device adds the host port number to the forwarding table entry; when it receives an IGMP Leave Group message from a host, it removes the host port from the table entry. It also periodically deletes entries if it does not receive IGMP membership reports from the multicast clients.



Note For more information on IP multicast and IGMP, see RFC 1112 and RFC 2236.

The multicast router set on the active device sends out periodic general queries to all VLANs. All hosts interested in this multicast traffic send join requests and are added to the forwarding table entry. The device creates one entry per VLAN in the IGMP snooping IP multicast forwarding table for each group from which it receives an IGMP join request.

The device supports IP multicast group-based bridging, instead of MAC-addressed based groups. With multicast MAC address-based groups, if an IP address being configured translates (aliases) to a previously

configured MAC address or to any reserved multicast MAC addresses (in the range 224.0.0.xxx), the command fails. Because the device uses IP multicast groups, there are no address aliasing issues.

The IP multicast groups learned through IGMP snooping are dynamic. However, you can statically configure multicast groups by using the **ip igmp snooping vlan *vlan-id* static *ip_address* interface *interface-id*** global configuration command. If you specify group membership for a multicast group address statically, your setting supersedes any automatic manipulation by IGMP snooping. Multicast group membership lists can consist of both user-defined and IGMP snooping-learned settings.

You can configure an IGMP snooping querier to support IGMP snooping in subnets without multicast interfaces because the multicast traffic does not need to be routed.

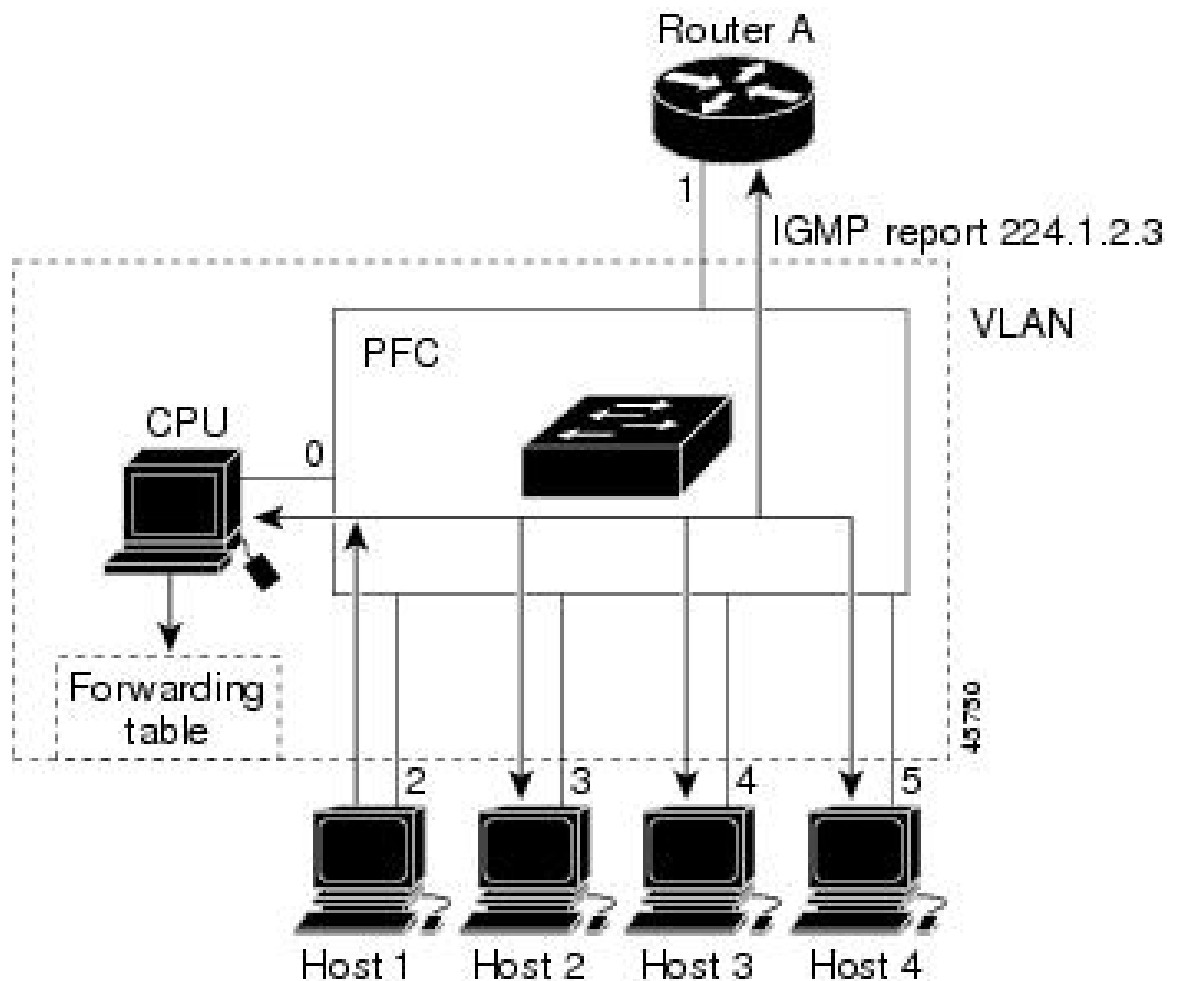
If a port spanning-tree, a port group, or a VLAN ID change occurs, the IGMP snooping-learned multicast groups from this port on the VLAN are deleted.

These sections describe IGMP snooping characteristics:

Joining a Multicast Group

Figure 8: Initial IGMP Join Message

When a host connected to the device wants to join an IP multicast group and it is an IGMP version 2 client, it sends an unsolicited IGMP join message, specifying the IP multicast group to join. Alternatively, when the device receives a general query from the router, it forwards the query to all ports in the VLAN. IGMP version 1 or version 2 hosts wanting to join the multicast group respond by sending a join message to the device. The device CPU creates a multicast forwarding-table entry for the group if it is not already present. The CPU also adds the interface where the join message was received to the forwarding-table entry. The host associated with that interface receives multicast traffic for that multicast group.



Router A sends a general query to the device, which forwards the query to ports 2 through 5, all of which are members of the same VLAN. Host 1 wants to join multicast group 224.1.2.3 and multicasts an IGMP membership report (IGMP join message) to the group. The device CPU uses the information in the IGMP report to set up a forwarding-table entry that includes the port numbers connected to Host 1 and to the router.

Table 6: IGMP Snooping Forwarding Table

Destination Address	Type of Packet	Ports
224.1.2.3	IGMP	1, 2

The device hardware can distinguish IGMP information packets from other packets for the multicast group. The information in the table tells the switching engine to send frames addressed to the 224.1.2.3 multicast IP address that are not IGMP packets to the router and to the host that has joined the group.

Figure 9: Second Host Joining a Multicast Group

If another host (for example, Host 4) sends an unsolicited IGMP join message for the same group, the CPU receives that message and adds the port number of Host 4 to the forwarding table. Because the forwarding table directs IGMP messages only to the CPU, the message is not flooded to other ports on the device. Any

Table 7: Updated IGMP Snooping Forwarding Table

Destination Address	Type of Packet	Ports
224.1.2.3	IGMP	1, 2, 5

The router sends periodic multicast general queries, and the device forwards these queries through all ports in the VLAN. Interested hosts respond to the queries. If at least one host in the VLAN wants to receive multicast traffic, the router continues forwarding the multicast traffic to the VLAN. The device forwards multicast group traffic only to those hosts listed in the forwarding table for that IP multicast group maintained by IGMP snooping.

When hosts want to leave a multicast group, they can silently leave, or they can send a leave message. When the device receives a leave message from a host, it sends a group-specific query to learn if any other devices

connected to that interface are interested in traffic for the specific multicast group. The device then updates the forwarding table for that MAC group so that only those hosts interested in receiving multicast traffic for the group are listed in the forwarding table. If the router receives no reports from a VLAN, it removes the group for the VLAN from its IGMP cache.

Immediate Leave

The device uses IGMP snooping Immediate Leave to remove from the forwarding table an interface that sends a leave message without the device sending group-specific queries to the interface. The VLAN interface is pruned from the multicast tree for the multicast group specified in the original leave message. Immediate Leave ensures optimal bandwidth management for all hosts on a switched network, even when multiple multicast groups are simultaneously in use.

Immediate Leave is only supported on IGMP version 2 hosts. IGMP version 2 is the default version for the device.



Note You should use the Immediate Leave feature only on VLANs where a single host is connected to each port. If Immediate Leave is enabled on VLANs where more than one host is connected to a port, some hosts may be dropped inadvertently.

IGMP Configurable-Leave Timer

You can configure the time that the device waits after sending a group-specific query to determine if hosts are still interested in a specific multicast group. The IGMP leave response time can be configured from 100 to 32767 milliseconds.

IGMP Report Suppression

IGMP report suppression is supported only when the multicast query has IGMPv1 and IGMPv2 reports. This feature is not supported when the query includes IGMPv3 reports.

The device uses IGMP report suppression to forward only one IGMP report per multicast router query to multicast devices. When IGMP report suppression is enabled (the default), the device sends the first IGMP report from all hosts for a group to all the multicast routers. The device does not send the remaining IGMP reports for the group to the multicast routers. This feature prevents duplicate reports from being sent to the multicast devices.

If the multicast router query includes requests only for IGMPv1 and IGMPv2 reports, the device forwards only the first IGMPv1 or IGMPv2 report from all hosts for a group to all the multicast routers.

If the multicast router query also includes requests for IGMPv3 reports, the device forwards all IGMPv1, IGMPv2, and IGMPv3 reports for a group to the multicast devices.

If you disable IGMP report suppression, all IGMP reports are forwarded to the multicast routers.

IGMP Snooping and Device Stacks

IGMP snooping functions across the device stack; that is, IGMP control information from one device is distributed to all devices in the stack. Regardless of the stack member through which IGMP multicast data enters the stack, the data reaches the hosts that have registered for that group.

If a device in the stack fails or is removed from the stack, only the members of the multicast group that are on that device will not receive the multicast data. All other members of a multicast group on other devices in

the stack continue to receive multicast data streams. However, multicast groups that are common for both Layer 2 and Layer 3 (IP multicast routing) might take longer to converge if the active device is removed.

IGMP Filtering and Throttling

In some environments, for example, metropolitan or multiple-dwelling unit (MDU) installations, you might want to control the set of multicast groups to which a user on a switch port can belong. You can control the distribution of multicast services, such as IP/TV, based on some type of subscription or service plan. You might also want to limit the number of multicast groups to which a user on a switch port can belong.

With the IGMP filtering feature, you can filter multicast joins on a per-port basis by configuring IP multicast profiles and associating them with individual switch ports. An IGMP profile can contain one or more multicast groups and specifies whether access to the group is permitted or denied. If an IGMP profile denying access to a multicast group is applied to a switch port, the IGMP join report requesting the stream of IP multicast traffic is dropped, and the port is not allowed to receive IP multicast traffic from that group. If the filtering action permits access to the multicast group, the IGMP report from the port is forwarded for normal processing. You can also set the maximum number of IGMP groups that a Layer 2 interface can join.

IGMP filtering controls only group-specific query and membership reports, including join and leave reports. It does not control general IGMP queries. IGMP filtering has no relationship with the function that directs the forwarding of IP multicast traffic. The filtering feature operates in the same manner whether CGMP or MVR is used to forward the multicast traffic.

IGMP filtering applies only to the dynamic learning of IP multicast group addresses, not static configuration.

With the IGMP throttling feature, you can set the maximum number of IGMP groups that a Layer 2 interface can join. If the maximum number of IGMP groups is set, the IGMP snooping forwarding table contains the maximum number of entries, and the interface receives an IGMP join report, you can configure an interface to drop the IGMP report or to replace the randomly selected multicast entry with the received IGMP report.



Note IGMPv3 join and leave messages are not supported on a device running IGMP filtering.

Default IGMP Configuration

This table displays the default IGMP configuration for the device.

Table 8: Default IGMP Configuration

Feature	Default Setting
Multilayer device as a member of a multicast group	No group memberships are defined.
Access to multicast groups	All groups are allowed on an interface.
IGMP version	Version 2 on all interfaces.
IGMP host-query message interval	60 seconds on all interfaces.
IGMP query timeout	60 seconds on all interfaces.
IGMP maximum query response time	10 seconds on all interfaces.

Feature	Default Setting
Multilayer device as a statically connected member	Disabled.

Default IGMP Snooping Configuration

This table displays the default IGMP snooping configuration for the device.

Table 9: Default IGMP Snooping Configuration

Feature	Default Setting
IGMP snooping	Enabled globally and per VLAN
Multicast routers	None configured
IGMP snooping Immediate Leave	Disabled
Static groups	None configured
TCN ¹ flood query count	2
TCN query solicitation	Disabled
IGMP snooping querier	Disabled
IGMP report suppression	Enabled

¹ (1) TCN = Topology Change Notification

Default IGMP Filtering and Throttling Configuration

This table displays the default IGMP filtering and throttling configuration for the device.

Table 10: Default IGMP Filtering Configuration

Feature	Default Setting
IGMP filters	None applied.
IGMP maximum number of IGMP groups	No maximum set. Note When the maximum number of groups is in the for table, the default IGMP throttling action is to deny report.
IGMP profiles	None defined.
IGMP profile action	Deny the range addresses.

How to Configure IGMP

Configuring the Device as a Member of a Group

You can configure the device as a member of a multicast group and discover multicast reachability in a network. If all the multicast-capable routers and multilayer devices that you administer are members of a multicast group, pinging that group causes all of these devices to respond. The devices respond to ICMP echo-request packets addressed to a group of which they are members. Another example is the multicast trace-route tools provided in the software.



Caution Performing this procedure might impact the CPU performance because the CPU will receive all data traffic for the group address.

This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enabled 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>interface-id</i> Example: Device (config)# interface GigabitEthernet 1/0/1	Specifies the Layer 3 interface on which you want to enable multicast routing, and enters interface configuration mode. The specified interface must be one of the following: <ul style="list-style-type: none"> • A routed port—A physical port that has been configured as a Layer 3 port by entering the no switchport interface configuration command. • An SVI—A VLAN interface created by using the interface vlan <i>vlan-id</i> global configuration command. These interfaces must have IP addresses assigned to them.
Step 4	ip igmp join-group <i>group-address</i> Example:	Configures the device to join a multicast group. By default, no group memberships are defined.

	Command or Action	Purpose
	Device(config-if)# ip igmp join-group 225.2.2.2	For <i>group-address</i> , specify the multicast IP address in dotted decimal notation.
Step 5	end Example: Device(config-if)# end	Returns to privileged EXEC mode.
Step 6	show ip igmp interface [<i>interface-id</i>] Example: Device# show ip igmp interface GigabitEthernet 1/0/1	Verifies your entries.
Step 7	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Changing the IGMP Version

By default, the switch uses IGMP Version 2, which provides features such as the IGMP query timeout and the maximum query response time.

All systems on the subnet must support the same version. The switch does not automatically detect Version 1 systems and switch to Version 1. You can mix Version 1 and Version 2 hosts on the subnet because Version 2 routers or switches always work correctly with IGMPv1 hosts.

Configure the switch for Version 1 if your hosts do not support Version 2.

This procedure is optional.

Procedure

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

	Command or Action	Purpose
Step 3	interface <i>interface-id</i> Example: <pre>Device(config)# interface gigabitethernet 1/0/1</pre>	Specifies the interface to be configured, and enters the interface configuration mode.
Step 4	ip igmp version {1 2 3} Example: <pre>Device(config-if)# ip igmp version 2</pre>	Specifies the IGMP version that the switch uses. Note If you change to Version 1, you cannot configure the ip igmp query-interval or the ip igmp query-max-response-time interface configuration commands. To return to the default setting, use the no ip igmp version interface configuration command.
Step 5	end Example: <pre>Device(config-if)# end</pre>	Returns to privileged EXEC mode.
Step 6	show ip igmp interface [<i>interface-id</i>] Example: <pre>Device# show ip igmp interface</pre>	Verifies your entries.
Step 7	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Modifying the IGMP Host-Query Message Interval

The device periodically sends IGMP host-query messages to discover which multicast groups are present on attached networks. These messages are sent to the all-hosts multicast group (224.0.0.1) with a time-to-live (TTL) of 1. The device sends host-query messages to refresh its knowledge of memberships present on the network. If, after some number of queries, the software discovers that no local hosts are members of a multicast group, the software stops forwarding multicast packets to the local network from remote origins for that group and sends a prune message upstream toward the source.

The device elects a PIM designated router (DR) for the LAN (subnet). The designated router is responsible for sending IGMP host-query messages to all hosts on the LAN. In sparse mode, the designated router also

sends PIM register and PIM join messages toward the RP router. With IGMPv2, the DR is the router or multilayer device with the highest IP address. With IGMPv1, the DR is elected according to the multicast routing protocol that runs on the LAN.

This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: <pre>Device(config)# interface gigabitethernet 1/0/1</pre>	Specifies the Layer 3 interface on which you want to enable multicast routing, and enters interface configuration mode. The specified interface must be one of the following: <ul style="list-style-type: none"> • A routed port—A physical port that has been configured as a Layer 3 port by entering the no switchport interface configuration command. • An SVI—A VLAN interface created by using the interface vlan <i>vlan-id</i> global configuration command. These interfaces must have IP addresses assigned to them.
Step 4	ip igmp query-interval <i>seconds</i> Example: <pre>Device(config-if)# ip igmp query-interval 75</pre>	Configures the frequency at which the designated router sends IGMP host-query messages. By default, the designated router sends IGMP host-query messages every 60 seconds to keep the IGMP overhead very low on hosts and networks. The range is 1 to 65535.
Step 5	end Example:	Returns to privileged EXEC mode.

	Command or Action	Purpose
	Device(config)# end	
Step 6	show ip igmp interface <i>[interface-id]</i> Example: Device# show ip igmp interface	Displays
Step 7	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Changing the Maximum Query Response Time for IGMPv2

If you are using IGMPv2, you can change the maximum query response time advertised in IGMP queries. The maximum query response time enables the device to quickly detect that there are no more directly connected group members on a LAN. Decreasing the value enables the device to prune groups faster.

This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enabled 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>interface-id</i> Example: Device(config)# interface GigabitEthernet 1/0/1	Specifies the Layer 3 interface on which you want to enable multicast routing, and enters interface configuration mode. The specified interface must be one of the following: <ul style="list-style-type: none"> • A routed port—A physical port that has been configured as a Layer 3 port by entering the no switchport interface configuration command.

	Command or Action	Purpose
		<ul style="list-style-type: none"> An SVI—A VLAN interface created by using the interface vlan <i>vlan-id</i> global configuration command. <p>These interfaces must have IP addresses assigned to them.</p>
Step 4	ip igmp query-max-response-time <i>seconds</i> Example: <pre>Device(config-if)# ip igmp query-max-response-time 15</pre>	<p>Changes the maximum query response time advertised in IGMP queries.</p> <p>The default is 10 seconds. The range is 1 to 25.</p>
Step 5	end Example: <pre>Device(config-if)# end</pre>	Returns to privileged EXEC mode.
Step 6	show ip igmp interface [<i>interface-id</i>] Example: <pre>Device# show ip igmp interface</pre>	Verifies your entries.
Step 7	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Configuring the Device as a Statically Connected Member

At various times, either there is not a group member on a network segment or a host that cannot report its group membership by using IGMP. However, you may want multicast traffic to be sent to that network segment. The following commands are used to pull multicast traffic down to a network segment:

- **ip igmp join-group**—The device accepts the multicast packets in addition to forwarding them. Accepting the multicast packets prevents the device from fast switching.
- **ip igmp static-group**—The device does not accept the packets itself, but only forwards them. This method enables fast switching. The outgoing interface appears in the IGMP cache, but the device itself is not a member, as evidenced by lack of an L (local) flag in the multicast route entry.

This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enabled 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 interface-id Example: Device(config)# interface GigabitEthernet 1/0/1	Specifies the Layer 3 interface on which you want to enable multicast routing, and enters interface configuration mode. The specified interface must be one of the following: <ul style="list-style-type: none"> • A routed port—A physical port that has been configured as a Layer 3 port by entering the no switchport interface configuration command. • An SVI—A VLAN interface created by using the interface vlan vlan-id global configuration command. These interfaces must have IP addresses assigned to them.
Step 4	ip igmp static-group group-address Example: Device(config-if)# ip igmp static-group 239.100.100.101	Configures the device as a statically connected member of a group. By default, this feature is disabled.
Step 5	end Example: Device(config-if)# end	Returns to privileged EXEC mode.
Step 6	show ip igmp interface [interface-id] Example: Device# show ip igmp interface GigabitEthernet 1/0/1	Verifies your entries.

	Command or Action	Purpose
Step 7	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Configuring IGMP Profiles

Follow these steps to create an IGMP profile:

This task is optional.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enabled privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ip igmp profile <i>profile number</i> Example: <pre>Device(config)# ip igmp profile 3</pre>	Assigns a number to the profile you are configuring, and enters IGMP profile configuration mode. The profile number range is 1 to 4294967295. When you are in IGMP profile configuration mode, you can create the profile by using these commands: <ul style="list-style-type: none"> • deny—Specifies that matching addresses are denied; this is the default. • exit—Exits from igmp-profile configuration mode. • no—Negates a command or returns to its defaults. • permit—Specifies that matching addresses are permitted. • range—Specifies a range of IP addresses for the profile. You can enter a single IP address or a range with a start and an end address.

	Command or Action	Purpose
		<p>The default for the device is to have no IGMP profiles configured.</p> <p>Note To delete a profile, use the no ip igmp profile <i>profile number</i> global configuration command.</p>
Step 4	<p>permit deny</p> <p>Example:</p> <pre>Device(config-igmp-profile)# permit</pre>	<p>(Optional) Sets the action to permit or deny access to the IP multicast address. If no action is configured, the default for the profile is to deny access.</p>
Step 5	<p>range <i>ip multicast address</i></p> <p>Example:</p> <pre>Device(config-igmp-profile)# range 229.9.9.0</pre>	<p>Enters the IP multicast address or range of IP multicast addresses to which access is being controlled. If entering a range, enter the low IP multicast address, a space, and the high IP multicast address.</p> <p>You can use the range command multiple times to enter multiple addresses or ranges of addresses.</p> <p>Note To delete an IP multicast address or range of IP multicast addresses, use the no range <i>ip multicast address</i> IGMP profile configuration command.</p>
Step 6	<p>end</p> <p>Example:</p> <pre>Device(config-if)# end</pre>	<p>Returns to privileged EXEC mode.</p>
Step 7	<p>show ip igmp profile <i>profile number</i></p> <p>Example:</p> <pre>Device# show ip igmp profile 3</pre>	<p>Verifies the profile configuration.</p>
Step 8	<p>show running-config</p> <p>Example:</p> <pre>Device# show running-config</pre>	<p>Verifies your entries.</p>
Step 9	<p>copy running-config startup-config</p> <p>Example:</p>	<p>(Optional) Saves your entries in the configuration file.</p>

	Command or Action	Purpose
	Device# copy running-config startup-config	

Applying IGMP Profiles

To control access as defined in an IGMP profile, you have to apply the profile to the appropriate interfaces. You can apply IGMP profiles only to Layer 2 access ports; you cannot apply IGMP profiles to routed ports or SVIs. You cannot apply profiles to ports that belong to an EtherChannel port group. You can apply a profile to multiple interfaces, but each interface can have only one profile applied to it.

Follow these steps to apply an IGMP profile to a switch port:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enabled privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface interface-id Example: Device(config)# interface GigabitEthernet 1/0/1	Specifies the physical interface, and enters interface configuration mode. The interface must be a Layer 2 port that does not belong to an EtherChannel port group.
Step 4	ip igmp filter profile number Example: Device(config-if)# ip igmp filter 321	Applies the specified IGMP profile to the interface. The range is 1 to 4294967295. Note To remove a profile from an interface, use the no ip igmp filter profile number interface configuration command.
Step 5	end Example: Device(config-if)# end	Returns to privileged EXEC mode.
Step 6	show running-config Example:	Verifies your entries.

	Command or Action	Purpose
	Device# <code>show running-config</code>	
Step 7	copy running-config startup-config Example: Device# <code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Setting the Maximum Number of IGMP Groups

Follow these steps to set the maximum number of IGMP groups that a Layer 2 interface can join:

Before you begin

This restriction can be applied to Layer 2 ports only; you cannot set a maximum number of IGMP groups on routed ports or SVIs. You also can use this command on a logical EtherChannel interface but cannot use it on ports that belong to an EtherChannel port group.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> <code>enable</code>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# <code>interface gigabitethernet1/0/2</code>	Specifies the interface to be configured, and enters interface configuration mode. The interface can be a Layer 2 port that does not belong to an EtherChannel group or a EtherChannel interface.
Step 4	ip igmp max-groups <i>number</i> Example: Device(config-if)# <code>ip igmp max-groups 20</code>	Sets the maximum number of IGMP groups that the interface can join. The range is 0 to 4294967294. The default is to have no maximum set.

	Command or Action	Purpose
Step 5	end Example: Device (config)# end	Returns to privileged EXEC mode.
Step 6	show running-config interface <i>interface-id</i> Example: Device# show running-config interface gigabitethernet1/0/1	Verifies your entries.
Step 7	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring the IGMP Throttling Action

After you set the maximum number of IGMP groups that a Layer 2 interface can join, you can configure an interface to replace the existing group with the new group for which the IGMP report was received.

Follow these steps to configure the throttling action when the maximum number of entries is in the forwarding table:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enabled 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>interface-id</i> Example: Device (config)# interface gigabitethernet1/0/1	Specifies the physical interface to be configured, and enters interface configuration mode. The interface can be a Layer 2 port that does not belong to an EtherChannel group or an EtherChannel interface. The interface cannot be a trunk port.

	Command or Action	Purpose
Step 4	ip igmp max-groups action {deny replace} Example: <pre>Device(config-if)# ip igmp max-groups action replace</pre>	<p>When an interface receives an IGMP report and the maximum number of entries is in the forwarding table, specifies the action that the interface takes:</p> <ul style="list-style-type: none"> • deny—Drops the report. If you configure this throttling action, the entries that were previously in the forwarding table are not removed but are aged out. After these entries are aged out and the maximum number of entries is in the forwarding table, the device drops the next IGMP report received on the interface. • replace—Replaces the existing group with the new group for which the IGMP report was received. If you configure this throttling action, the entries that were previously in the forwarding table are removed. When the maximum number of entries is in the forwarding table, the device replaces a randomly selected entry with the received IGMP report. <p>To prevent the device from removing the forwarding-table entries, you can configure the IGMP throttling action before an interface adds entries to the forwarding table.</p> <p>Note To return to the default action of dropping the report, use the no ip igmp max-groups action interface configuration command.</p>
Step 5	end Example: <pre>Device(config-if)# end</pre>	Returns to privileged EXEC mode.
Step 6	show running-config interface interface-id Example: <pre>Device# show running-config interface gigabitethernet1/0/1</pre>	Verifies your entries.
Step 7	copy running-config startup-config Example:	(Optional) Saves your entries in the configuration file.

	Command or Action	Purpose
	Device# copy running-config startup-config	

Configuring the Device to Forward Multicast Traffic in the Absence of Directly Connected IGMP Hosts

Perform this optional task to configure the device to forward multicast traffic in the absence of directly connected IGMP hosts.

Procedure

	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 gigabitethernet 1/0/1	Enters interface configuration mode. <ul style="list-style-type: none"> • For the <i>type</i> and <i>number</i> arguments, specify an interface that is connected to hosts.
Step 4	Do one of the following: <ul style="list-style-type: none"> • ip igmp join-group <i>group-address</i> • ip igmp static-group <i>{* group-address [source source-address]</i> Example: Device(config-if)# ip igmp join-group 225.2.2.2 Example: Device(config-if)# ip igmp static-group 225.2.2.2	The first sample shows how to configure an interface on the device to join the specified group. With this method, the device accepts the multicast packets in addition to forwarding them. Accepting the multicast packets prevents the device from fast switching. The second example shows how to configure static group membership entries on an interface. With this method, the device does not accept the packets itself, but only forwards them. Hence, this method allows fast switching. The outgoing interface appears in the IGMP cache, but the device itself is not a member, as evidenced by lack of an “L” (local) flag in the multicast route entry

	Command or Action	Purpose
Step 5	end Example: <pre>Device# (config-if) # end</pre>	Returns to privileged EXEC mode.
Step 6	show ip igmp interface [<i>interface-type</i> <i>interface-number</i>] Example: <pre>Device# show ip igmp interface</pre>	(Optional) Displays multicast-related information about an interface.

Controlling Access to an SSM Network Using IGMP Extended Access Lists

Perform this optional task to control access to an SSM network by using an IGMP extended access list that filters SSM traffic based on source address, group address, or both.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ip multicast-routing [distributed] Example: <pre>Device(config)# ip multicast-routing distributed</pre>	Enables IP multicast routing. <ul style="list-style-type: none"> • The distributed keyword is required for IPv4 multicast..
Step 4	ip pim ssm { default range <i>access-list</i> } Example: <pre>Device(config)# ip pim ssm default</pre>	Configures SSM service. <ul style="list-style-type: none"> • The default keyword defines the SSM range access list as 232/8. • The range keyword specifies the standard IP access list number or name that defines the SSM range.
Step 5	ip access-list extended <i>access-list -name</i> Example:	Specifies an extended named IP access list.

	Command or Action	Purpose
	Device(config)# ip access-list extended mygroup	
Step 6	<p>deny igmp <i>source source-wildcard destination destination-wildcard [igmp-type] [precedence precedence] [tos tos] [log] [time-range time-range-name] [fragments]</i></p> <p>Example:</p> <pre>Device(config-ext-nacl)# deny igmp host 10.1.2.3 any</pre>	<p>(Optional) Filters the specified source address or group address from the IGMP report, thereby restricting hosts on a subnet from membership to the (S, G) channel.</p> <ul style="list-style-type: none"> • Repeat this step to restrict hosts on a subnet membership to other (S, G) channels. (These sources should be more specific than a subsequent permit statement because any sources or groups not specifically permitted are denied.) • Remember that the access list ends in an implicit deny statement. • This example shows how to create a deny statement that filters all groups for source 10.1.2.3, which effectively denies the source.
Step 7	<p>permit igmp <i>source source-wildcard destination destination-wildcard [igmp-type] [precedence precedence] [tos tos] [log] [time-range time-range-name] [fragments]</i></p> <p>Example:</p> <pre>Device(config-ext-nacl)# permit igmp any any</pre>	<p>Allows a source address or group address in an IGMP report to pass the IP access list.</p> <ul style="list-style-type: none"> • You must have at least one permit statement in an access list. • Repeat this step to allow other sources to pass the IP access list. • This example shows how to allow group membership to sources and groups not denied by prior deny statements.
Step 8	<p>exit</p> <p>Example:</p> <pre>Device(config-ext-nacl)# exit</pre>	Exits the current configuration session and returns to global configuration mode.
Step 9	<p>interface type number</p> <p>Example:</p> <pre>Device(config)# interface ethernet 0</pre>	Selects an interface that is connected to hosts on which IGMPv3 can be enabled.
Step 10	<p>ip igmp access-group <i>access-list</i></p> <p>Example:</p> <pre>Device(config-if)# ip igmp access-group mygroup</pre>	Applies the specified access list to IGMP reports.

	Command or Action	Purpose
Step 11	ip pim sparse-mode Example: Device(config-if)# ip pim sparse-mode	Enables PIM-SM on the interface. Note You must use sparse mode.
Step 12	Repeat Steps 1 through 11 on all interfaces that require access control of SSM channel membership.	--
Step 13	ip igmp version 3 Example: Device(config-if)# ip igmp version 3	Enables IGMPv3 on this interface. The default version of IGMP is IGMP version 2. Version 3 is required by SSM.
Step 14	Repeat Step 13 on all host-facing interfaces.	--
Step 15	end Example: Device(config-if)# end	Returns to privileged EXEC mode.

How to Configure IGMP Snooping

Enabling IGMP Snooping

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip igmp snooping Example: Device(config)# ip igmp snooping	Globally enables IGMP snooping after it has been disabled.
Step 4	bridge-domain <i>bridge-id</i> Example: Device(config)# bridge-domain 100	(Optional) Enters bridge domain configuration mode.

	Command or Action	Purpose
Step 5	ip igmp snooping Example: Device(config-bdomain) # ip igmp snooping	(Optional) Enables IGMP snooping on the bridge domain interface being configured. <ul style="list-style-type: none"> Required only if IGMP snooping was previously explicitly disabled on the specified bridge domain.
Step 6	end Example: Device(config-bdomain) # end	Returns to privileged EXEC mode.

Enabling or Disabling IGMP Snooping on a VLAN Interface

Follow these steps to enable IGMP snooping on a VLAN interface:

Procedure

	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 igmp snooping vlan <i>vlan-id</i> Example: Device(config) # ip igmp snooping vlan 7	Enables IGMP snooping on the VLAN interface. The VLAN ID range is 1 to 1001 and 1006 to 4094. IGMP snooping must be globally enabled before you can enable VLAN snooping. Note To disable IGMP snooping on a VLAN interface, use the no ip igmp snooping vlan <i>vlan-id</i> global configuration command for the specified VLAN number.
Step 4	end Example: Device(config) # end	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 5	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Setting the Snooping Method

Multicast-capable router ports are added to the forwarding table for every Layer 2 multicast entry. The device learns of the ports through one of these methods:

- Snooping on IGMP queries, Protocol-Independent Multicast (PIM) packets
- Statically connecting to a multicast router port using the **ip igmp snooping mrouter** global configuration command

Beginning in privileged EXEC mode, follow these steps to alter the method in which a VLAN interface accesses a multicast router:

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ip igmp snooping vlan <i>vlan-id</i> mrouter interface {GigabitEthernet Port-Channel TenGigabitEthernet} Example: <pre>Device(config)# ip igmp snooping vlan 1 mrouter interface GigabitEthernet1/0/3</pre>	Enables IGMP snooping on a VLAN. The VLAN ID range is 1 to 1001 and 1006 to 4094.
Step 4	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 5	show ip igmp snooping Example: Device# <code>show ip igmp snooping</code>	Verifies the configuration.
Step 6	copy running-config startup-config Example: Device# <code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Configuring a Multicast Router Port

Perform these steps to add a multicast router port (enable a static connection to a multicast router) on the device.



Note Static connections to multicast routers are supported only on device ports.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> <code>enable</code>	Enabled privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 3	ip igmp snooping vlan <i>vlan-id</i> mrouter interface <i>interface-id</i> Example: Device(config)# <code>ip igmp snooping vlan 5 mrouter interface GigabitEthernet 1/0/1</code>	Specifies the multicast router VLAN ID and the interface to the multicast router. <ul style="list-style-type: none"> • The VLAN ID range is 1 to 1001 and 1006 to 4094. • The interface can be a physical interface or a port channel. The port-channel range is 1 to 128.

	Command or Action	Purpose
		Note To remove a multicast router port from the VLAN, use the no ip igmp snooping vlan <i>vlan-id</i> mrouter interface <i>interface-id</i> global configuration command.
Step 4	end Example: <pre>Device(config-if)# end</pre>	Returns to privileged EXEC mode.
Step 5	show ip igmp snooping mrouter [vlan <i>vlan-id</i>] Example: <pre>Device# show ip igmp snooping mrouter vlan 5</pre>	Verifies that IGMP snooping is enabled on the VLAN interface.
Step 6	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Configuring a Host Statically to Join a Group

Hosts or Layer 2 ports normally join multicast groups dynamically, but you can also statically configure a host on an interface.

Follow these steps to add a Layer 2 port as a member of a multicast group:

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enabled privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ip igmp snooping vlan <i>vlan-id</i> static <i>ip_address</i> interface <i>interface-id</i> Example:	Statically configures a Layer 2 port as a member of a multicast group:

	Command or Action	Purpose
	<pre>Device(config)# ip igmp snooping vlan 105 static 230.0.0.1 interface gigabitethernet1/0/1</pre>	<ul style="list-style-type: none"> • <i>vlan-id</i> is the multicast group VLAN ID. The range is 1 to 1001 and 1006 to 4094. • <i>ip-address</i> is the group IP address. • <i>interface-id</i> is the member port. It can be a physical interface or a port channel (1 to 128). <p>Note To remove the Layer 2 port from the multicast group, use the no ip igmp snooping vlan <i>vlan-id</i> static <i>mac-address</i> interface <i>interface-id</i> global configuration command.</p>
Step 4	<p>end</p> <p>Example:</p> <pre>Device(config-if)# end</pre>	Returns to privileged EXEC mode.
Step 5	<p>show ip igmp snooping groups</p> <p>Example:</p> <pre>Device# show ip igmp snooping groups</pre>	Verifies the member port and the IP address.
Step 6	<p>copy running-config startup-config</p> <p>Example:</p> <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Enabling IGMP Immediate Leave

When you enable IGMP Immediate Leave, the device immediately removes a port when it detects an IGMP Version 2 leave message on that port. You should use the Immediate-Leave feature only when there is a single receiver present on every port in the VLAN.



Note Immediate Leave is supported only on IGMP Version 2 hosts. IGMP Version 2 is the default version for the device.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ip igmp snooping vlan <i>vlan-id</i> immediate-leave Example: <pre>Device(config)# ip igmp snooping vlan 21 immediate-leave</pre>	Enables IGMP Immediate Leave on the VLAN interface. Note To disable IGMP Immediate Leave on a VLAN, use the no ip igmp snooping vlan <i>vlan-id</i> immediate-leave global configuration command.
Step 4	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 5	show ip igmp snooping vlan <i>vlan-id</i> Example: <pre>Device# show ip igmp snooping vlan 21</pre>	Verifies that Immediate Leave is enabled on the VLAN interface.
Step 6	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.

Configuring the IGMP Leave Timer

You can configure the leave time globally or on a per-VLAN basis. Follow these steps to enable the IGMP configurable-leave timer:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enabled 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 igmp snooping last-member-query-interval <i>time</i> Example: Device(config)# ip igmp snooping last-member-query-interval 1000	Configures the IGMP leave timer globally. The range is 100 to 32767 milliseconds. The default leave time is 1000 milliseconds. Note To globally reset the IGMP leave timer to the default setting, use the no ip igmp snooping last-member-query-interval global configuration command.
Step 4	ip igmp snooping vlan <i>vlan-id</i> last-member-query-interval <i>time</i> Example: Device(config)# ip igmp snooping vlan 210 last-member-query-interval 1000	(Optional) Configures the IGMP leave time on the VLAN interface. The range is 100 to 32767 milliseconds. Note Configuring the leave time on a VLAN overrides the globally configured timer. Note To remove the configured IGMP leave-time setting from the specified VLAN, use the no ip igmp snooping vlan <i>vlan-id</i> last-member-query-interval global configuration command.
Step 5	end Example: Device(config-if)# end	Returns to privileged EXEC mode.
Step 6	show ip igmp snooping Example: Device# show ip igmp snooping	(Optional) Displays the configured IGMP leave time.

	Command or Action	Purpose
Step 7	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Configuring the IGMP Robustness-Variable

Use the following procedure to configure the IGMP robustness variable on the device.

The robustness variable is the integer used by IGMP snooping during calculations for IGMP messages. The robustness variable provides fine tuning to allow for expected packet loss.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enabled privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ip igmp snooping robustness-variable <i>count</i> Example: <pre>Device(config)# ip igmp snooping robustness-variable 3</pre>	Configures the IGMP robustness variable. The range is 1 to 3 times. The recommended value for the robustness variable is 2. Use this command to change the value of the robustness variable for IGMP snooping from the default (2) to a specified value.
Step 4	ip igmp snooping vlan <i>vlan-id</i> robustness-variable <i>count</i> Example: <pre>Device(config)#ip igmp snooping vlan 100 robustness-variable 3</pre>	(Optional) Configures the IGMP robustness variable on the VLAN interface. The range is 1 to 3 times. The recommended value for the robustness variable is 2. Note Configuring the robustness variable count on a VLAN overrides the globally configured value.
Step 5	end Example:	Returns to privileged EXEC mode.

	Command or Action	Purpose
	Device (config-if) # end	
Step 6	show ip igmp snooping Example: Device# show ip igmp snooping	(Optional) Displays the configured IGMP robustness variable count.
Step 7	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring the IGMP Last Member Query Count

To configure the number of times the device sends IGMP group-specific or group-source-specific (with IGMP version 3) query messages in response to receiving a group-specific or group-source-specific leave message, use this command.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enabled 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 igmp snooping last-member-query-count count Example: Device (config)# ip igmp snooping last-member-query-count 3	Configures the IGMP last member query count. The range is 1 to 7 messages. The default is 2 messages.
Step 4	ip igmp snooping vlan <i>vlan-id</i> last-member-query-count count Example:	(Optional) Configures the IGMP last member query count on the VLAN interface. The range is 1 to 7 messages.

	Command or Action	Purpose
	Device(config)# ip igmp snooping vlan 100 last-member-query-count 3	Note Configuring the last member query count on a VLAN overrides the globally configured timer.
Step 5	end Example: Device(config-if)# end	Returns to privileged EXEC mode.
Step 6	show ip igmp snooping Example: Device# show ip igmp snooping	(Optional) Displays the configured IGMP last member query count.
Step 7	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring TCN-Related Commands

Controlling the Multicast Flooding Time After a TCN Event

You can configure the number of general queries by which multicast data traffic is flooded after a topology change notification (TCN) event. If you set the TCN flood query count to 1 the flooding stops after receiving 1 general query. If you set the count to 7, the flooding continues until 7 general queries are received. Groups are relearned based on the general queries received during the TCN event.

Some examples of TCN events are when the client location is changed and the receiver is on same port that was blocked but is now forwarding, and when a port goes down without sending a leave message.

Follow these steps to configure the TCN flood query count:

Procedure

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

	Command or Action	Purpose
Step 3	ip igmp snooping tcn flood query count <i>count</i> Example: <pre>Device(config)# ip igmp snooping tcn flood query count 3</pre>	<p>Specifies the number of IGMP general queries for which the multicast traffic is flooded.</p> <p>The range is 1 to 10. The default, the flooding query count is 2.</p> <p>Note To return to the default flooding query count, use the no ip igmp snooping tcn flood query count global configuration command.</p>
Step 4	end Example: <pre>Device(config-if)# end</pre>	Returns to privileged EXEC mode.
Step 5	show ip igmp snooping Example: <pre>Device# show ip igmp snooping</pre>	Verifies the TCN settings.
Step 6	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Recovering from Flood Mode

When a topology change occurs, the spanning-tree root sends a special IGMP leave message (also known as global leave) with the group multicast address 0.0.0.0. However, you can enable the device to send the global leave message whether it is the spanning-tree root or not. When the router receives this special leave, it immediately sends general queries, which expedite the process of recovering from the flood mode during the TCN event. Leaves are always sent if the device is the spanning-tree root regardless of this configuration.

Follow these steps to enable sending of leave messages:

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> 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	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ip igmp snooping tcn query solicit Example: <pre>Device(config)# ip igmp snooping tcn query solicit</pre>	<p>Sends an IGMP leave message (global leave) to speed the process of recovering from the flood mode caused during a TCN event. By default, query solicitation is disabled.</p> <p>Note To return to the default query solicitation, use the no ip igmp snooping tcn query solicit global configuration command.</p>
Step 4	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 5	show ip igmp snooping Example: <pre>Device# show ip igmp snooping</pre>	Verifies the TCN settings.
Step 6	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Disabling Multicast Flooding During a TCN Event

When the device receives a TCN, multicast traffic is flooded to all the ports until 2 general queries are received. If the device has many ports with attached hosts that are subscribed to different multicast groups, this flooding might exceed the capacity of the link and cause packet loss. Follow these steps to control TCN flooding:

Procedure

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

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# interface GigabitEthernet 1/0/1	Specifies the interface to be configured, and enters interface configuration mode.
Step 4	no ip igmp snooping tcn flood Example: Device(config-if)# no ip igmp snooping tcn flood	Disables the flooding of multicast traffic during a spanning-tree TCN event. By default, multicast flooding is enabled on an interface. Note To re-enable multicast flooding on an interface, use the ip igmp snooping tcn flood interface configuration command.
Step 5	end Example: Device(config-if)# end	Returns to privileged EXEC mode.
Step 6	show ip igmp snooping Example: Device# show ip igmp snooping	Verifies the TCN settings.
Step 7	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring the IGMP Snooping Querier

Follow these steps to enable the IGMP snooping querier feature in a VLAN:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enabled 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 igmp snooping querier Example: Device(config)# ip igmp snooping querier	Enables the IGMP snooping querier.
Step 4	ip igmp snooping querier address <i>ip_address</i> Example: Device(config)# ip igmp snooping querier address 172.16.24.1	(Optional) Specifies an IP address for the IGMP snooping querier. If you do not specify an IP address, the querier tries to use the global IP address configured for the IGMP querier. Note The IGMP snooping querier does not generate an IGMP general query if it cannot find an IP address on the device.
Step 5	ip igmp snooping querier query-interval <i>interval-count</i> Example: Device(config)# ip igmp snooping querier query-interval 30	(Optional) Sets the interval between IGMP queriers. The range is 1 to 18000 seconds.
Step 6	ip igmp snooping querier tcn query [count <i>count</i> interval <i>interval</i>] Example: Device(config)# ip igmp snooping querier tcn query interval 20	(Optional) Sets the time between Topology Change Notification (TCN) queries. The count range is 1 to 10. The interval range is 1 to 255 seconds.
Step 7	ip igmp snooping querier timer expiry <i>timeout</i> Example: Device(config)# ip igmp snooping querier timer expiry 180	(Optional) Sets the length of time until the IGMP querier expires. The range is 60 to 300 seconds.

	Command or Action	Purpose
Step 8	ip igmp snooping querier version <i>version</i> Example: <pre>Device(config)# ip igmp snooping querier version 2</pre>	(Optional) Selects the IGMP version number that the querier feature uses. Select 1 or 2.
Step 9	end Example: <pre>Device(config-if)# end</pre>	Returns to privileged EXEC mode.
Step 10	show ip igmp snooping vlan <i>vlan-id</i> Example: <pre>Device# show ip igmp snooping vlan 30</pre>	(Optional) Verifies that the IGMP snooping querier is enabled on the VLAN interface. The VLAN ID range is 1 to 1001 and 1006 to 4094.
Step 11	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Disabling IGMP Report Suppression

Follow these steps to disable IGMP report suppression:

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enabled privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	no ip igmp snooping report-suppression Example: <pre>Device(config)# no ip igmp snooping report-suppression</pre>	Disables IGMP report suppression. When report suppression is disabled, all IGMP reports are forwarded to the multicast routers. IGMP report suppression is enabled by default.

	Command or Action	Purpose
		When IGMP report suppression is enabled, the device forwards only one IGMP report per multicast router query. Note To re-enable IGMP report suppression, use the ip igmp snooping report-suppression global configuration command.
Step 4	end Example: Device(config-if)# end	Returns to privileged EXEC mode.
Step 5	show ip igmp snooping Example: Device# show ip igmp snooping	Verifies that IGMP report suppression is disabled.
Step 6	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Monitoring IGMP

You can display specific statistics, such as the contents of IP routing tables, caches, and databases.



Note This release does not support per-route statistics.

You can display information to learn resource usage and solve network problems. You can also display information about node reachability and discover the routing path that packets of your device are taking through the network.

You can use any of the privileged EXEC commands in the following table to display various routing statistics.

Table 11: Commands for Displaying System and Network Statistics

Command	Purpose
show ip igmp filter	Displays IGMP filter information.

Command	Purpose
show ip igmp groups [<i>type-number</i> <i>detail</i>]	Displays the multicast groups that are discovered by IGMP.
show ip igmp interface [<i>type number</i>]	Displays multicast-related information about the interface.
show ip igmp membership [<i>name/group address</i> all tracked]	Displays IGMP membership information.
show ip igmp profile [<i>profile_number</i>]	Displays IGMP profile information.
show ip igmp ssm-mapping [<i>hostname/IP address</i>]	Displays IGMP SSM mapping information.
show ip igmp static-group { class-map [interface [<i>type</i>]]	Displays static group information.
show ip igmp vrf	Displays the selected VPN routing/forwarding table. Note The show ip igmp vrf vrf-name command displays the snooping information for the specified VRF and displays the snooping information for the groups command to see the groups.

Monitoring IGMP Snooping Information

You can display IGMP snooping information for dynamically learned and statically configured router ports and VLAN interfaces. You can also display MAC address multicast entries for a VLAN configured for IGMP snooping.

Table 12: Commands for Displaying IGMP Snooping Information

Command	Purpose
show ip igmp snooping detail	Displays the operational state information.
show ip igmp snooping groups [count dynamic [count] user [count]]	Displays multicast table information for the specified interface. <ul style="list-style-type: none"> • count—Displays the total number of entries in the multicast table, including actual entries. • dynamic—Displays entries learned dynamically. • user—Displays only the user-configured entries.
show ip igmp snooping groups [count [vlan <i>vlan-id</i> [<i>A.B.C.D</i> count]]	Displays multicast table information for the specified VLAN. <ul style="list-style-type: none"> • count—Displays the total number of entries in the multicast table. • vlan—Displays group information for the specified VLAN.
show ip igmp snooping igmpv2-tracking	Displays the IGMP snooping tracking information. Note This command displays group membership for wireless LANs and not for wired LANs. Use the show ip igmp snooping command to display.

Command	Purpose
show ip igmp snooping groups vlan <i>vlan-id</i> [<i>ip_address</i> count dynamic [count] user [count]]	Displays multicast table information <ul style="list-style-type: none"> • vlan-id—The VLAN ID range. • count—Displays the total number of actual entries. • dynamic—Displays entries learned dynamically. • ip_address—Displays characteristics of the IP address. • user—Displays only the user-defined entries.
show ip igmp snooping mrouter [vlan <i>vlan-id</i>]	Displays information on dynamically learned multicast routers. <p>Note When you enable IGMP snooping, a multicast router is considered a dynamic entry.</p> <p>(Optional) Enter vlan <i>vlan-id</i> to display information for a specific VLAN.</p>
show ip igmp snooping querier [detail vlan <i>vlan-id</i>]	Displays information about the IP address of the querier and the number of query messages in the VLAN. <p>(Optional) Enter detail to display the details of the querier.</p> <p>(Optional) Enter vlan <i>vlan-id</i> to display information for a specific VLAN.</p>
show ip igmp snooping querier [vlan <i>vlan-id</i>] detail	Displays information about the IP address of the querier and the number of query messages in the VLAN and the configuration of the querier in the VLAN.
show ip igmp snooping [vlan <i>vlan-id</i> [detail]]	Displays the snooping configuration for all interfaces on the device or for a specified interface. <p>(Optional) Enter vlan <i>vlan-id</i> to display information for a specific VLAN. The range is 1001 and 1006 to 4094.</p>

Monitoring IGMP Filtering and Throttling Configuration

You can display IGMP profile characteristics, and you can display the IGMP profile and maximum group configuration for all interfaces on the device or for a specified interface. You can also display the IGMP throttling configuration for all interfaces on the device or for a specified interface.

Table 13: Commands for Displaying IGMP Filtering and Throttling Configuration

Command	Purpose
show ip igmp profile [<i>profile number</i>]	Displays the specified IGMP profile or all the profiles defined on the device.

Command	Purpose
show running-config [<i>interface interface-id</i>]	Displays the configuration of the specified interface. If no interface is specified, displays the configuration of all interfaces on the device, including (if configured) the maximum number of IGMP groups an interface can belong and the IGMP profile applied to the interface.

Configuration Examples for IGMP

Example: Configuring the Device as a Member of a Multicast Group

This example shows how to enable the device to join multicast group 255.2.2.2:

```
Device(config)# interface gigabitEthernet1/0/1
Device(config-if)# ip igmp join-group 255.2.2.2
Device(config-if)#
```

Example: Controlling Access to Multicast Groups

To limit the number of joins on the interface, configure the port for filter which associates with the IGMP profile.

```
Device# configure terminal
Device(config)# ip igmp profile 10
Device(config-igmp-profile)# ?

IGMP profile configuration commands:
deny matching addresses are denied
exit Exit from igmp profile configuration mode
no Negate a command or set its defaults
permit matching addresses are permitted
range add a range to the set

Device(config-igmp-profile)# range 172.16.5.1
Device(config-igmp-profile)# exit
Device(config)# interface gigabitEthernet 2/0/10
Device(config-if)# ip igmp filter 10
```

Examples: Configuring IGMP Snooping

This example shows how to enable a static connection to a multicast router:

```
Device# configure terminal
Device(config)# ip igmp snooping vlan 200 mrouter interface gigabitEthernet1/0/2
Device(config)# end
```

This example shows how to statically configure a host on a port:

```
Device# configure terminal
Device(config)# ip igmp snooping vlan 105 static 224.2.4.12 interface gigabitethernet1/0/1
Device(config)# end
```

This example shows how to enable IGMP Immediate Leave on VLAN 130:

```
Device# configure terminal
Device(config)# ip igmp snooping vlan 130 immediate-leave
Device(config)# end
```

This example shows how to set the IGMP snooping querier source address to 10.0.0.64:

```
Device# configure terminal
Device(config)# ip igmp snooping querier 10.0.0.64
Device(config)# end
```

This example shows how to set the IGMP snooping querier maximum response time to 25 seconds:

```
Device# configure terminal
Device(config)# ip igmp snooping querier query-interval 25
Device(config)# end
```

This example shows how to set the IGMP snooping querier timeout to 60 seconds:

```
Device# configure terminal
Device(config)# ip igmp snooping querier timer expiry 60
Device(config)# end
```

This example shows how to set the IGMP snooping querier feature to Version 2:

```
Device# configure terminal
Device(config)# no ip igmp snooping querier version 2
Device(config)# end
```

Example: Configuring IGMP Profiles

This example shows how to create IGMP profile 4 allowing access to the single IP multicast address and how to verify the configuration. If the action was to deny (the default), it would not appear in the **show ip igmp profile** output display.

```
Device(config)# ip igmp profile 4
Device(config-igmp-profile)# permit
Device(config-igmp-profile)# range 229.9.9.0
Device(config-igmp-profile)# end
Device# show ip igmp profile 4
IGMP Profile 4
    permit
    range 229.9.9.0 229.9.9.0
```

Example: Applying IGMP Profile

This example shows how to apply IGMP profile 4 to a port:

```
Device(config)# interface gigabitEthernet1/0/2
Device(config-if)# ip igmp filter 4
Device(config-if)# end
```

Example: Setting the Maximum Number of IGMP Groups

This example shows how to limit to 25 the number of IGMP groups that a port can join:

```
Device(config)# interface GigabitEthernet1/0/2
Device(config-if)# ip igmp max-groups 25
Device(config-if)# end
```

Example: Interface Configuration as a Routed Port

This example shows how to configure an interface on the device as a routed port. This configuration is required on the interface for several IP multicast routing configuration procedures that require running the **no switchport** command.

```
Device# configure terminal
Device(config)# interface GigabitEthernet1/0/9
Device(config-if)# description interface to be use as routed port
Device(config-if)# no switchport
Device(config-if)# ip address 10.20.20.1 255.255.255.0
Device(config-if)# ip pim sparse-mode
Device(config-if)# ip igmp join-group 224.1.2.3 source 15.15.15.2
Device(config-if)# end
Device# configure terminal
Device# show run interface gigabitEthernet 1/0/9

Current configuration : 166 bytes
!
interface GigabitEthernet1/0/9
 no switchport
 ip address 10.20.20.1 255.255.255.0
 ip pim sparse-mode
 ip igmp static-group 224.1.2.3 source 15.15.15.2
end
```

Example: Interface Configuration as an SVI

This example shows how to configure an interface on the device as an SVI. This configuration is required on the interface for several IP multicast routing configuration procedures that require running the **no switchport** command.

```
Device(config)# interface vlan 150
Device(config-if)# ip address 10.20.20.1 255.255.255.0
Device(config-if)# ip pim sparse-mode
Device(config-if)# ip igmp join-group 224.1.2.3 source 15.15.15.2
Device(config-if)# end
Device# configure terminal
Device(config)# ip igmp snooping vlan 20 static 224.1.2.3 interface gigabitEthernet 1/0/9
Device# show run interface vlan 150

Current configuration : 137 bytes
!
interface vlan 150
```

```
ip address 10.20.20.1 255.255.255.0
ip pim sparse-mode
ip igmp static-group 224.1.2.3 source 15.15.15.2
end
```

Example: Configuring the Device to Forward Multicast Traffic in the Absence of Directly Connected IGMP Hosts

The following example shows how to configure a device to forward multicast traffic in the absence of directly connected IGMP hosts using the **ip igmp join-group** command. With this method, the device accepts the multicast packets in addition to forwarding them. Accepting the multicast packets prevents the device from fast switching.

In this example, GigabitEthernet interface 1/0/1 on the device is configured to join the group 225.2.2.2:

```
interface GigabitEthernet1/0/1
 ip igmp join-group 225.2.2.2
```

The following example shows how to configure a device to forward multicast traffic in the absence of directly connected IGMP hosts using the **ip igmp static-group** command. With this method, the device does not accept the packets itself, but only forwards them. Hence, this method allows fast switching. The outgoing interface appears in the IGMP cache, but the device itself is not a member, as evidenced by lack of an “L” (local) flag in the multicast route entry.

In this example, static group membership entries for group 225.2.2.2 are configured on Fast Ethernet interface 0/1/0:

```
interface GigabitEthernet1/0/1
 ip igmp static-group 225.2.2.2
```

Controlling Access to an SSM Network Using IGMP Extended Access Lists

This section contains the following configuration examples for controlling access to an SSM network using IGMP extended access lists:



Note Keep in mind that access lists are very flexible: there are many combinations of permit and deny statements one could use in an access list to filter multicast traffic. The examples in this section simply provide a few examples of how it can be done.

Example: Denying All States for a Group G

The following example shows how to deny all states for a group G. In this example, Fast Ethernet interface 0/0/0 is configured to filter all sources for SSM group 232.2.2.2 in IGMPv3 reports, which effectively denies this group.

```
ip access-list extended test1
 deny igmp any host 232.2.2.2
 permit igmp any any
!
```

Example: Denying All States for a Source S

```
interface GigabitEthernet 1/0/1
 ip igmp access-group test1
```

Example: Denying All States for a Source S

The following example shows how to deny all states for a source S. In this example, Gigabit Ethernet interface 1/1/0 is configured to filter all groups for source 10.2.1.32 in IGMPv3 reports, which effectively denies this source.

```
ip access-list extended test2
 deny igmp host 10.2.1.32 any
 permit igmp any any
!
interface GigabitEthernet1/0/1
 ip igmp access-group test2
```

Example: Permitting All States for a Group G

The following example shows how to permit all states for a group G. In this example, Gigabit Ethernet interface 1/2/0 is configured to accept all sources for SSM group 232.1.1.10 in IGMPv3 reports, which effectively accepts this group altogether.

```
ip access-list extended test3
 permit igmp any host 232.1.1.10
!
interface GigabitEthernet 1/2/0
 ip igmp access-group test3
```

Example: Permitting All States for a Source S

The following example shows how to permit all states for a source S. In this example, Gigabit Ethernet interface 1/2 is configured to accept all groups for source 10.6.23.32 in IGMPv3 reports, which effectively accepts this source altogether.

```
ip access-list extended test4
 permit igmp host 10.6.23.32 any
!
interface GigabitEthernet1/2/0
 ip igmp access-group test4
```

Example: Filtering a Source S for a Group G

The following example shows how to filter a particular source S for a group G. In this example, Gigabit Ethernet interface 0/3/0 is configured to filter source 232.2.2.2 for SSM group 232.2.30.30 in IGMPv3 reports.

```
ip access-list extended test5
 deny igmp host 10.4.4.4 host 232.2.30.30
 permit igmp any any
!
interface GigabitEthernet0/3/0
 ip igmp access-group test5
```


Additional References for IGMP

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the IP Multicast Routing Commands section of the <i>Command Reference (Catalyst 9400 Series Switches)</i>

Feature History for IGMP

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	IGMP	IGMP is used to dynamically register individual hosts in a multicast group on a particular LAN. Enabling PIM on an interface also enables IGMP. IGMP provides a means to automatically control and limit the flow of multicast traffic throughout your network with the use of special multicast queriers and hosts.

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>.



CHAPTER 5

Configuring IGMP Proxy

- [Prerequisites for IGMP Proxy, on page 93](#)
- [Information About IGMP Proxy, on page 93](#)
- [How to Configure IGMP Proxy, on page 97](#)
- [Configuration Examples for IGMP Proxy, on page 103](#)
- [Additional References for IGMP Proxy, on page 104](#)
- [Feature History for IGMP Proxy, on page 105](#)

Prerequisites for IGMP Proxy

- All devices on the IGMP UDL have the same subnet address. If all devices on the UDL cannot have the same subnet address, the upstream device must be configured with secondary addresses to match all of the subnets to which the downstream devices are attached.
- IP multicast is enabled and the PIM interfaces are configured. When you are configuring PIM interfaces for IGMP proxy, use PIM sparse mode (PIM-SM) when the interface is operating in a sparse-mode region and you are running static RP, bootstrap (BSR), or Auto-RP with the Auto-RP listener capability.

Information About IGMP Proxy

IGMP Proxy

An IGMP proxy enables hosts in a unidirectional link routing (UDLR) environment that are not directly connected to a downstream router to join a multicast group sourced from an upstream network.

There are two methods of implementing IGMP Proxy:

- IGMP Proxy for a Single Upstream Interface
- IGMP Proxy for Multiple Upstream Interfaces

IGMP Proxy for a Single Upstream Interface

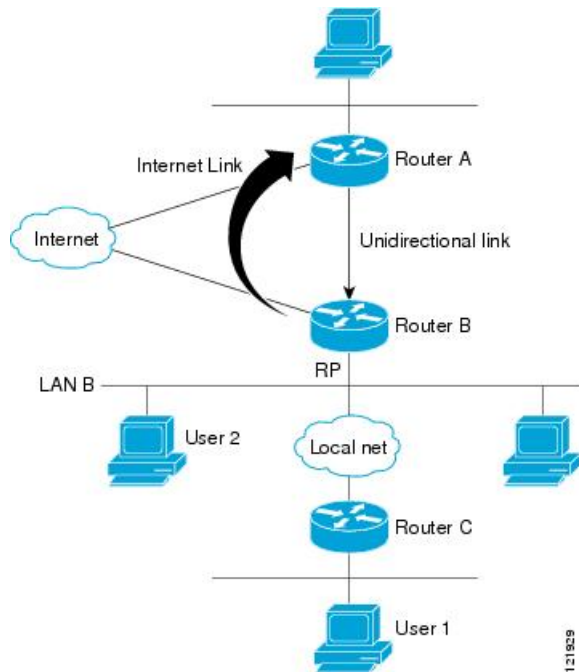
The [figure](#) below illustrates a sample topology that shows two UDLR scenarios:

- Traditional UDL routing scenario--A UDL device with directly connected receivers.
- IGMP proxy scenario--UDL device without directly connected receivers.

IGMP UDLs are needed on the upstream and downstream devices.



Note Although the following illustration and example uses routers in the configuration, any device (router or switch) can be used.



Scenario 1 - Traditional UDLR Scenario (UDL Device with Directly Connected Receivers)

For scenario 1, no IGMP proxy mechanism is needed. In this scenario, the following sequence of events occurs:

1. User 2 sends an IGMP membership report requesting interest in group G.
2. Router B receives the IGMP membership report, adds a forwarding entry for group G on LAN B, and proxies the IGMP report to Router A, which is the UDLR upstream device.
3. The IGMP report is then proxied across the Internet link.
4. Router A receives the IGMP proxy and maintains a forwarding entry on the unidirectional link.

Scenario 2 - IGMP Proxy Scenario (UDL Device without Directly Connected Receivers)

For scenario 2, the IGMP proxy mechanism is needed to enable hosts that are not directly connected to a downstream device to join a multicast group sourced from an upstream network. In this scenario, the following sequence of events occurs:

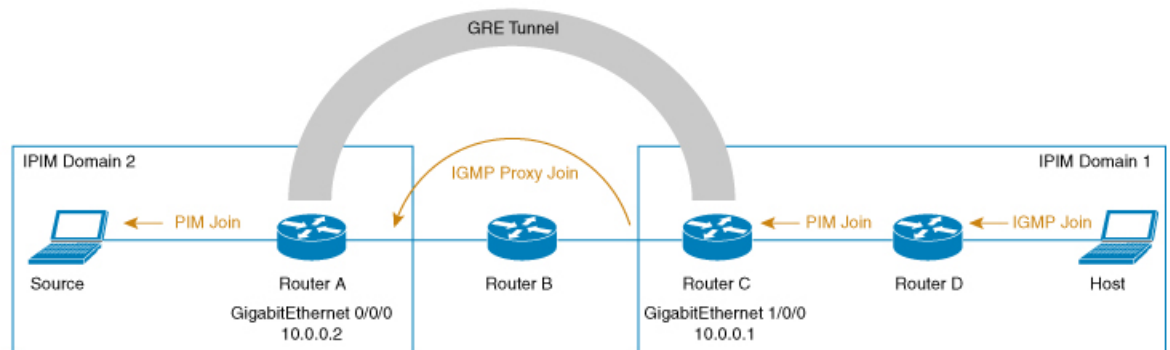
1. User 1 sends an IGMP membership report requesting interest in group G.

2. Router C sends a PIM Join message hop-by-hop to the RP (Router B).
3. Router B receives the PIM Join message and adds a forwarding entry for group G on LAN B.
4. Router B periodically checks its mroute table and proxies the IGMP membership report to its upstream UDL device across the Internet link.
5. Router A creates and maintains a forwarding entry on the unidirectional link (UDL).

In an enterprise network, it is desirable to be able to receive IP multicast traffic via satellite and forward the traffic throughout the network. With unidirectional link routing (UDLR) alone, scenario 2 would not be possible because receiving hosts must be directly connected to the downstream device, Router B. The IGMP proxy mechanism overcomes this limitation by creating an IGMP report for (*, G) entries in the multicast forwarding table. To make this scenario functional, therefore, you must enable IGMP report forwarding of proxied (*, G) multicast static route (mroute) entries (using the **ip igmp mroute-proxy** command) and enable the mroute proxy service (using the **ip igmp proxy-service** command) on interfaces leading to PIM-enabled networks with potential members.



Note Because PIM messages are not forwarded upstream, each downstream network and the upstream network have a separate domain.



Scenario 3 - IGMP Proxy Scenario without UDLR

For scenario 3, the IGMP proxy mechanism is used to enable hosts to receive traffic from an upstream network, without using a UDLR link. In this scenario, the following sequence of events occurs:

1. The host is in PIM Domain 1 and sends an IGMP membership report (a join request) to **Router D** requesting interest in group G. **Router D** converts the IGMP Join to a PIM join and sends it to **Router C**. This request should now be sent upstream, from **Router C** to **Router A**. The routers are in two different PIM domains (not PIM neighbors) and are connected through a GRE tunnel instead.
2. **Router C** converts the PIM join message to an IGMP proxy join so that it can be forwarded to the GRE tunnel endpoint, which is **Router A**.



Note IGMP proxy joins can be transferred across 1 hop only.

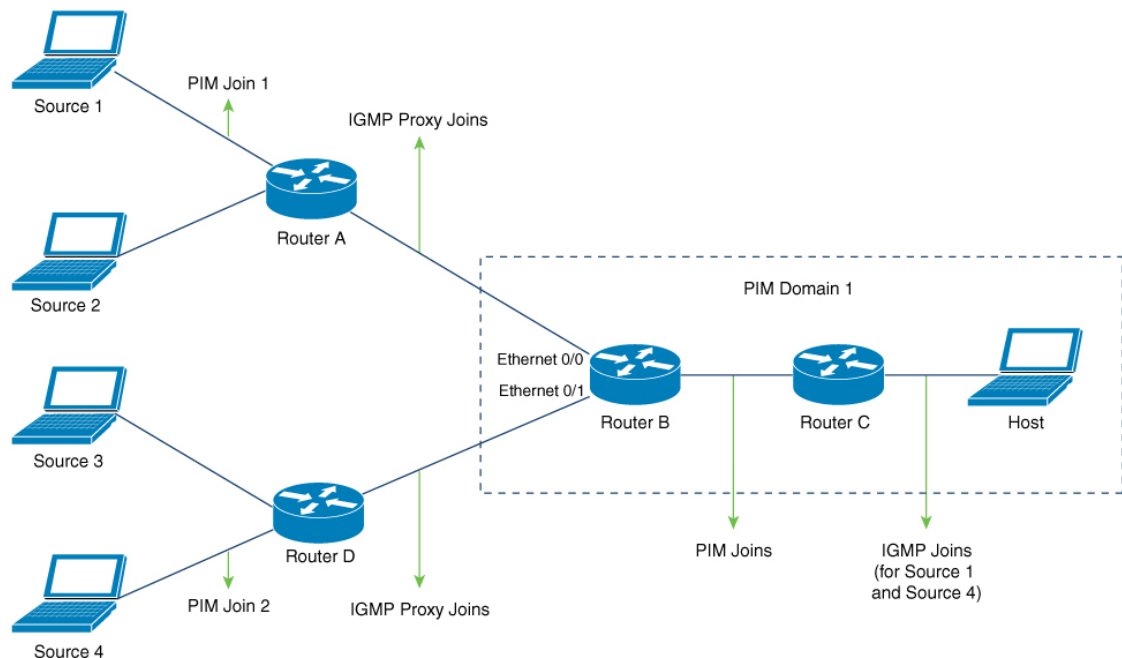
In the [figure](#) above, the GRE tunnel provides this single hop between Router C and Router A (bypassing Router B).

In the absence of a GRE tunnel, devices from different PIM domains must have directly (back-to-back) connected interfaces.

- After the IGMP proxy join reaches **Router A**, it is forwarded to the source device as a PIM join message.

IGMP Proxy for Multiple Upstream Interfaces

IGMP proxy also enables the user to request data from multiple upstream interfaces. You can implement IGMP proxy by this method if there are more number of upstream devices in the network. With this method, you can also implement IGMP proxy for a single upstream device as in any of the three scenarios described in the previous section.



In this method, IGMP proxy is used to enable the user to receive traffic from multiple upstream devices. The following sequence of events occurs:

- The host is in PIM Domain 1 and sends multiple IGMP membership reports (join requests) to **Router C** requesting interest in different groups. **Router C** converts the IGMP joins to PIM joins and sends them to **Router B**. These requests should be sent upstream, from **Router B** to **Router A**. The routers are in two different PIM domains (not PIM neighbors).
- Router B** converts the PIM join messages to IGMP proxy joins so that it can be forwarded to the upward interfaces.

3. A class-map is configured globally. This class-map describes the information about multicast groups. The IGMP proxy joins for different multicast groups are sent if the following conditions are met:
 - There is (*, G) or and (S, G) entry for that group.
 - (*, G) or the (S, G) entry has a NON-NULL OIF list.
4. During the IGMP proxy intervals, the IGMP proxy joins for different groups are sent through the respective upstream interfaces.
5. After the IGMP proxy join reaches **Router A**, it is forwarded to the different source devices as PIM join messages.

How to Configure IGMP Proxy

Configuring the Upstream UDL Device for IGMP UDLR

Perform this task to configure the upstream UDL device for IGMP UDLR.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	interface type number Example: <pre>Device(config)# interface gigabitethernet 1/0/0</pre>	Enters interface configuration mode. <ul style="list-style-type: none"> • For the <i>type</i> and <i>number</i> arguments, specify the interface to be used as the UDL on the upstream device.
Step 4	ip igmp unidirectional-link Example: <pre>Device(config-if)# ip igmp unidirectional-link</pre>	Configures IGMP on the interface to be unidirectional for IGMP UDLR.
Step 5	end Example: <pre>Device(config-if)# end</pre>	Ends the current configuration session and returns to privileged EXEC mode.

Configuring the Downstream UDL Device for IGMP UDLR with IGMP Proxy Support

Perform this task to configure the downstream UDL device for IGMP UDLR with IGMP proxy support.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: <pre>Device(config)# interface gigabitethernet 0/0/0</pre>	Enters interface configuration mode. <ul style="list-style-type: none"> • For the <i>type</i> and <i>number</i> arguments, specify the interface to be used as the UDL on the downstream device for IGMP UDLR.
Step 4	ip igmp unidirectional-link Example: <pre>Device(config-if)# ip igmp unidirectional-link</pre>	Configures IGMP on the interface to be unidirectional for IGMP UDLR.
Step 5	exit Example: <pre>Device(config-if)# exit</pre>	Exits interface configuration mode and returns to global configuration mode.
Step 6	interface <i>type number</i> Example: <pre>Device(config)# interface gigabitethernet 1/0/0</pre>	Enters interface configuration mode. <ul style="list-style-type: none"> • For the <i>type</i> and <i>number</i> arguments, select an interface that is facing the nondirectly connected hosts.
Step 7	ip igmp mroute-proxy <i>type number</i> Example: <pre>Device(config-if)# ip igmp mroute-proxy loopback 0</pre>	Enables IGMP report forwarding of proxied (*, G) multicast static route (mroute) entries. <ul style="list-style-type: none"> • This step is performed to enable the forwarding of IGMP reports to a proxy service interface for all (*, G) forwarding entries in the multicast forwarding table.

	Command or Action	Purpose
		<ul style="list-style-type: none"> In this example, the ip igmp mroute-proxy command is configured on Gigabit Ethernet interface 1/0/0 to request that IGMP reports be sent to loopback interface 0 for all groups in the mroute table that are forwarded to Gigabit Ethernet interface 1/0/0.
Step 8	exit Example: <pre>Device(config-if)# exit</pre>	Exits interface configuration mode and returns to global configuration mode.
Step 9	interface <i>type number</i> Example: <pre>Device(config)# interface loopback 0</pre>	<p>Enters interface configuration mode for the specified interface.</p> <ul style="list-style-type: none"> In this example, loopback interface 0 is specified.
Step 10	ip igmp helper-address udl <i>interface-type interface-number</i> Example: <pre>Device(config-if)# ip igmp helper-address udl gigabitethernet 0/0/0</pre>	<p>Configures IGMP helping for UDLR.</p> <ul style="list-style-type: none"> This step allows the downstream device to helper IGMP reports received from hosts to an upstream device connected to a UDL associated with the interface specified for the <i>interface-type</i> and <i>interface-number</i> arguments. In the example topology, IGMP helping is configured over loopback interface 0 on the downstream device. Loopback interface 0, thus, is configured to helper IGMP reports from hosts to an upstream device connected to Gigabit Ethernet interface 0/0/0.
Step 11	ip igmp proxy-service Example: <pre>Device(config-if)# ip igmp proxy-service</pre>	<p>Enables the mroute proxy service.</p> <ul style="list-style-type: none"> When the mroute proxy service is enabled, the device periodically checks the static mroute table for (*, G) forwarding entries that match interfaces configured with the ip igmp mroute-proxy command (see Step 7) based on the IGMP query interval. Where there is a match, one IGMP report is created and received on this interface.

	Command or Action	Purpose
		<p>Note The ip igmp proxy-service command is intended to be used with the ip igmp helper-address (UDL) command.</p> <ul style="list-style-type: none"> In this example, the ip igmp proxy-service command is configured on loopback interface 0 to enable the forwarding of IGMP reports out the interface for all groups on interfaces registered through the ip igmp mroute-proxy command (see Step 7).
Step 12	end Example: <pre>Device(config-if)# end</pre>	Ends the current configuration session and returns to privileged EXEC mode.
Step 13	show ip igmp interface Example: <pre>Device# show ip igmp interface</pre>	(Optional) Displays multicast-related information about an interface.
Step 14	show ip igmp udldr Example: <pre>Device# show ip igmp udldr</pre>	(Optional) Displays UDLR information for directly connected multicast groups on interfaces that have a UDL helper address configured.

Configuring the Downstream Device for IGMP Proxy Join without UDLR

Perform this task to configure the downstream device for IGMP Proxy without UDLR.

(Referring to the [figure](#) above, all the steps are configured on **Router C**)

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device > enable</pre>	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	interface type number	Enters interface configuration mode.

	Command or Action	Purpose
	Example: <pre>Device(config)# interface gigabitethernet 1/0/0</pre>	For the <i>type</i> and <i>number</i> arguments, specify the interface that is facing the host.
Step 4	ip igmp mroute-proxy type number Example: <pre>Device(config-if)# ip igmp mroute-proxy loopback 0</pre>	<p>Enables the forwarding of IGMP reports to the specified proxy service interface, for forwarding of all proxied (*, G) multicast static route (mroute) entries in the multicast forwarding table.</p> <p>In the step example, <i>loopback interface 0</i> is such a proxy service interface.</p>
Step 5	exit Example: <pre>Device(config-if)# exit</pre>	Exits interface configuration mode and returns to global configuration mode.
Step 6	interface type number Example: <pre>Device(config)# interface loopback 0</pre>	<p>Enters interface configuration mode for the specified proxy service interface.</p> <p>In the step example, <i>loopback interface 0</i> is specified.</p>
Step 7	ip igmp helper-address ip-address Example: <pre>Device(config-if)# ip igmp helper-address 10.0.0.2</pre>	<p>Configures IGMP helper for IGMP proxy join.</p> <p>For the <i>ip-address</i> argument, specify the ip address of the upstream device to which the IGMP proxy join should reach.</p> <p>In the example topology, the IGMP helper is configured over loopback interface 0 on the downstream device (Router C).</p> <p>This command configures loopback interface 0 to help convert the PIM joins received from Router D to IGMP proxy joins and transfer them to the upstream device (Router A).</p>
Step 8	ip igmp proxy-service Example: <pre>Device(config-if) ip igmp proxy-service</pre>	<p>Enables the mroute proxy service.</p> <p>When the mroute proxy service is enabled, the device periodically checks the static mroute table for (*, G) forwarding entries that match interfaces configured with the ip igmp mroute-proxy command (see Step 7) based on the IGMP query interval. Where there is a match, one IGMP report is created and received on this interface.</p> <p>Note The ip igmp proxy-service command is intended to be used with the ip igmp helper-address command.</p>

	Command or Action	Purpose
Step 9	end Example: Device(config-if)# end	Ends the current configuration session and returns to privileged EXEC mode.
Step 10	show ip igmp interface Example: Device# show ip igmp interface	(Optional) Displays multicast-related information about an interface.

Configuring the Downstream Device for IGMP Proxy for Multiple Upstream Interfaces

Perform this task to configure the downstream device for IGMP Proxy for multiple upstream interfaces.

(Referring to the [figure](#) earlier, all the steps are configured on the *Router B* interface facing the upstream device)

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device > enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	class-map type multicast-flows name Example: Device(config-if)# class-map type multicast-flows proxymap	Configures the interface with a class-map where the uplink interfaces for different multicast groups are defined. The range of the multicast groups is from group 225.0.0.1 to 225.0.0.10.
Step 4	interface type number Example: Device(config)# interface ethernet 0/0	Enters interface configuration mode. For the <i>type</i> and <i>number</i> arguments, specify the interface that is facing the host.
Step 5	ip igmp upstream-proxy class-map-name Example: Device(config-if)# ip igmp upstream-proxy proxymap	Enables the interface with an IGMP proxy. The IGMP proxy joins for those groups in the class map are sent if the following conditions are met: <ul style="list-style-type: none"> (*,G) or (S,G) mroute exists for the group in the same mvrf as the interface. (*,G) or (S,G) mroute has a NON-NULL OIF list.

	Command or Action	Purpose
Step 6	ip igmp iif-starg Example: Device(config-if)# ip igmp iif-starg	Alters the RPF interface of the mroutes of those groups specified in the class map to Ethernet 0/0.
Step 7	ip igmp proxy-report-interval <i>time</i> Example: Device(config-if)# ip igmp proxy-report-interval 130	Configures the interval (in seconds) at which the proxy reports will be sent. The default value is 60 seconds.
Step 8	end Example: Device(config-if)# end	Ends the current configuration session and returns to privileged EXEC mode.
Step 9	show ip igmp interface Example: Device# show ip igmp interface	(Optional) Displays multicast-related information about an interface.

Configuration Examples for IGMP Proxy

Example: Configuring the Upstream UDL Device for IGMP UDLR

The following example shows how to configure the upstream UDL device for IGMP UDLR:

```
interface gigabitethernet 0/0/0
ip address 10.1.1.1 255.255.255.0
ip pim sparse-mode
!
interface gigabitethernet 1/0/0
ip address 10.2.1.1 255.255.255.0
ip pim sparse-mode
ip igmp unidirectional-link
!
interface gigabitethernet 2/0/0
ip address 10.3.1.1 255.255.255.0
```

Example: Configuring the Downstream UDL Device for IGMP UDLR with IGMP Proxy Support

The following example shows how to configure the downstream UDL device for IGMP UDLR with IGMP proxy support:

```
ip pim rp-address 10.5.1.1 5
access-list 5 permit 239.0.0.0 0.255.255.255
!
interface loopback 0
ip address 10.7.1.1 255.255.255.0
```

```

ip pim sparse-mode
ip igmp helper-address udl ethernet 0
ip igmp proxy-service
!
interface gigabitethernet 0/0/0
ip address 10.2.1.2 255.255.255.0
ip pim sparse-mode
ip igmp unidirectional-link
!
interface gigabitethernet 1/0/0
ip address 10.5.1.1 255.255.255.0
ip pim sparse-mode
ip igmp mroute-proxy loopback 0
!
interface gigabitethernet 2/0/0
ip address 10.6.1.1 255.255.255.0

```

Example: Configuring the Downstream Device for IGMP Proxy Join without UDLR

The following example shows how to configure the downstream device for IGMP proxy without UDLR:

```

interface Loopback0
ip address 2.2.2.2 255.255.0.0
ip pim sparse-dense-mode
ip igmp helper-address 99.99.99.1
ip igmp proxy-service
ip ospf 1 area 0
!

```

Example: Configuring the Downstream Device for IGMP Proxy for Multiple Upstream Interfaces

The following example shows how to configure the downstream device for IGMP proxy for multiple upstream interfaces.

```

interface gigabitethernet0/0
ip address 99.99.99.1 255.255.255.0
ip pim passive
ip igmp upstream-proxy 12
ip igmp iif-starg
ip igmp proxy-report-interval 100
end

class-map type multicast-flows 12
group 229.0.0.1
group 228.0.0.1 to 228.0.0.10

```

Additional References for IGMP Proxy

The following sections provide references related to customizing IGMP.

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the IP Multicast Routing Commands section of the <i>Command Reference (Catalyst 9400 Series Switches)</i>

Standards and RFCs

Standard/RFC	Title
RFC 1112	<i>Host extensions for IP multicasting</i>
RFC 2236	<i>Internet Group Management Protocol, Version 2</i>
RFC 3376	<i>Internet Group Management Protocol, Version 3</i>

Feature History for IGMP Proxy

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	IGMP Proxy	An IGMP proxy enables hosts in a unidirectional link routing (UDLR) environment that are not directly connected to a downstream router to join a multicast group sourced from an upstream network.
Cisco IOS XE Gibraltar 16.12.1	IGMP Proxy without UDLR	The IGMP proxy enables hosts to receive traffic from an upstream network, without using a UDLR link.
Cisco IOS XE Amsterdam 17.1.1	IGMP Proxy for multiple upstream interfaces.	IGMP proxy enables the user to receive traffic from multiple upstream devices.

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>.



CHAPTER 6

IGMP Explicit Tracking

This module describes the explicit tracking of hosts, groups, and channels for the Internet Group Management Protocol (IGMP).

- [Restrictions for IGMP Explicit Tracking, on page 107](#)
- [Information About IGMP Explicit Tracking, on page 108](#)
- [How to Configure IGMP Explicit Tracking, on page 109](#)
- [Configuration Examples for IGMP Explicit Tracking, on page 110](#)
- [Verifying IGMP Explicit Tracking, on page 111](#)
- [Additional References for IGMP Explicit Tracking, on page 113](#)
- [Feature History for IGMP Explicit Tracking, on page 114](#)

Restrictions for IGMP Explicit Tracking

The following restrictions apply to this feature.

- If one or more hosts that supports only IGMP Version 1 or Version 2 are present on a network, the leave latencies for the multicast groups to which these hosts are joined will revert to the leave latencies of the IGMP version of the hosts—approximately 3 seconds for IGMP Version 2 and up to 180 seconds for IGMP Version 1. This condition affects only multicast groups to which these legacy hosts are actually joined at any given point in time. In addition, the membership reports for these multicast groups sent by IGMPv3 hosts may revert to IGMP Version 1 or Version 2 membership reports, thus disabling explicit tracking of those host memberships.
- Explicit tracking of IGMP Version 3 lite (IGMP v3lite) or URL Rendezvous Directory (URD) channel membership reports is not supported. Therefore, the leave latency for multicast groups sending traffic to hosts using IGMPv3 lite or URD will be determined by the leave latency of the version of IGMP configured on the hosts (for IGMPv3, the leave latency is typically 3 seconds when explicit tracking is not configured).

Information About IGMP Explicit Tracking

IGMP Explicit Tracking

The Internet Group Management Protocol (IGMP) is used by IP hosts to report their multicast group memberships to neighboring multicast devices. The IGMP Explicit Tracking feature enables a multicast device to explicitly track the membership of all multicast hosts in a particular multiaccess network. IGMP explicit tracking can be enabled globally and on Layer3 interfaces.

The explicit tracking of hosts, groups, and channels enables the device to keep track of each individual host that is joined to a particular group or channel. The main benefits of this feature are that it provides minimal leave latencies, faster channel changing, and improved diagnostics capabilities for IGMP.

Minimal Leave Latencies

The main benefit of the explicit tracking of hosts, groups, and channels in IGMP is to allow minimal leave latencies when a host leaves a multicast group or channel. The length of time between a host wanting to leave and a device stopping traffic forwarding is called the IGMP leave latency. A device configured with IGMP Version 3 (IGMPv3) and explicit tracking can immediately stop forwarding traffic if the last host to request to receive traffic from the device indicates that it no longer wants to receive traffic. The leave latency is thus bound only by the packet transmission latencies in the multiaccess network and the processing time in the device.

In IGMP Version 2, when a device receives an IGMP leave message from a host, it must first send an IGMP group-specific query to learn if other hosts on the same multiaccess network are still requesting to receive traffic. If after a specific time (the default value is approximately 3 seconds) no host replies to the query, the device will then stop forwarding the traffic. This query process is required because, in IGMP Version 1 and 2, IGMP membership reports are suppressed if the same report is already sent by another host in the network. Therefore, it is impossible for the device to reliably know how many hosts on a multiaccess network are requesting to receive traffic.

Faster Channel Changing

In networks where bandwidth is constrained between multicast devices and hosts (like in xDSL deployments), the bandwidth between devices and hosts is typically large enough to only sustain, in general, N multicast streams to be received in parallel. In these deployments, each host will typically join to only one multicast stream and the overall number of allowed hosts will be limited to N. The effective leave latency in these environments defines the channel change time of the receiver application—a single host cannot receive the new multicast stream before forwarding of the old stream has stopped. If an application tries to change the channel faster than the leave latency, the application will overload the bandwidth of the access network, resulting in a temporary degradation of traffic flow for all hosts. The explicit tracking of hosts, groups, and channels in IGMP allows for minimal leave latencies, and thus allows for fast channel changing capabilities.

Improved Diagnostic Capabilities

The explicit tracking of hosts, groups, and channels in IGMP allows network administrators to easily determine which multicast hosts are joined to other multicast groups or channels.

How to Configure IGMP Explicit Tracking

Enabling Explicit Tracking Globally

You can enable explicit-tracking globally and on Layer 3 interfaces.

Procedure

	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 igmp snooping vlan <i>vlan-id</i> explicit-tracking Example: Device(config)# ip igmp snooping vlan 1 explicit-tracking	Enables IGMP explicit host tracking.
Step 4	exit Example: Device(config)# exit	Exits global configuration mode and returns to privileged EXEC mode.

Enabling Explicit Tracking on Layer 3 Interfaces

You can enable explicit-tracking globally and on Layer 3 interfaces.

Procedure

	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
Step 3	interface <i>type number</i> Example: Device(config)# interface vlan 77	Configures an interface and enters interface configuration mode.
Step 4	ip address <i>ip-address mask</i> Example: Device(config-if)# ip address 10.1.1.1 255.255.255.254	Sets a primary or secondary IP address for an interface.
Step 5	ip pim sparse-mode Example: Device(config-if)# ip pim sparse-mode	Enables Protocol Independent Multicast (PIM) sparse mode on an interface.
Step 6	ip igmp version 3 Example: Device(config-if)# ip igmp version 3	Configure Internet Group Management Protocol (IGMP) Version 3 (IGMPv3) on the device.
Step 7	ip igmp explicit-tracking Example: Device(config-if)# ip igmp explicit-tracking	Enables IGMP explicit host tracking.
Step 8	exit Example: Device(config)# exit	Exits global configuration mode and returns to privileged EXEC mode.

Configuration Examples for IGMP Explicit Tracking

Example: Enabling Explicit Tracking

The following example shows a basic configuration to enable IGMP explicit tracking globally:

```
Device# configure terminal
Device(config)# ip multicast routing
Device(config)# ip igmp snooping vlan 1 explicit-tracking
Device(config)# end
```

The following example shows a basic configuration to enable IGMP explicit tracking on Layer 3 interfaces:

```
Device# configure terminal
Device(config)# interface vlan 77
Device(config-if)# ip address 10.1.1.1 255.255.255.254
```

```
Device(config-if)# ip pim sparse-mode
Device(config-if)# ip igmp version 3
Device(config-if)# ip igmp explicit-tracking
Device(config-if)# end
```

Verifying IGMP Explicit Tracking

Procedure

Step 1 enable

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 show ip igmp snooping vlan *vlan-ID*

Example:

```
Device# show ip igmp snooping vlan 77
```

Displays snooping information in a Catalyst VLAN.

```
Device# show ip igmp snooping vlan 77
```

Global IGMP Snooping configuration:

```
-----
IGMP snooping           : Enabled
IGMPv3 snooping         : Enabled
Report suppression      : Enabled
TCN solicit query       : Disabled
TCN flood query count   : 2
Robustness variable     : 2
Last member query count : 2
Last member query interval : 1000
```

Vlan 77:

```
-----
IGMP snooping           : Enabled
IGMPv2 immediate leave  : Disabled
Explicit host tracking   : Enabled
Multicast router learning mode : pim-dvmrp
CGMP interoperability mode : IGMP_ONLY
Robustness variable     : 2
Last member query count : 2
Last member query interval : 1000
Device#
```

Step 3 show ip igmp groups *interface-type interface-number*

Example:

```
Device# show ip igmp groups GigabitEthernet 1/0/24
```

Displays the multicast groups that are directly connected to a device, and that are learned through IGMP.

```
show ip igmp groups GigabitEthernet 1/0/24
```

```
IGMP Connected Group Membership
Group Address      Interface          Uptime    Expires    Last Reporter    Group Accounted
203.0.113.245      GigabitEthernet1/0/24  00:00:35  stopped    10.34.34.2
203.0.113.244      GigabitEthernet1/0/24  00:00:35  stopped    10.34.34.2
203.0.113.247      GigabitEthernet1/0/24  00:00:35  stopped    10.34.34.2
203.0.113.246      GigabitEthernet1/0/24  00:00:35  stopped    10.34.34.2
203.0.113.241      GigabitEthernet1/0/24  00:00:35  stopped    10.34.34.2
203.0.113.240      GigabitEthernet1/0/24  00:00:35  stopped    10.34.34.2
203.0.113.243      GigabitEthernet1/0/24  00:00:35  stopped    10.34.34.2
203.0.113.242      GigabitEthernet1/0/24  00:00:35  stopped    10.34.34.2
203.0.113.253      GigabitEthernet1/0/24  00:00:35  stopped    10.34.34.2
203.0.113.252      GigabitEthernet1/0/24  00:00:35  stopped    10.34.34.2
203.0.113.221      GigabitEthernet1/0/24  00:00:35  stopped    10.34.34.2
203.0.113.254      GigabitEthernet1/0/24  00:00:35  stopped    10.34.34.2
203.0.113.249      GigabitEthernet1/0/24  00:00:35  stopped    10.34.34.2
203.0.113.248      GigabitEthernet1/0/24  00:00:35  stopped    10.34.34.2
203.0.113.251      GigabitEthernet1/0/24  00:00:35  stopped    10.34.34.2
203.0.113.250      GigabitEthernet1/0/24  00:00:35  stopped    10.34.34.2
203.0.113.228      GigabitEthernet1/0/24  00:00:35  stopped    10.34.34.2
203.0.113.229      GigabitEthernet1/0/24  00:00:35  stopped    10.34.34.2
203.0.113.230      GigabitEthernet1/0/24  00:00:35  stopped    10.34.34.2
203.0.113.231      GigabitEthernet1/0/24  00:00:35  stopped    10.34.34.2
203.0.113.224      GigabitEthernet1/0/24  00:00:35  stopped    10.34.34.2
```

Step 4 show ip igmp membership tracked

Example:

```
Device# show ip igmp membership tracked
```

Displays the multicast groups with the explicit tracking feature enabled.

```
Device# show ip igmp membership tracked
```

```
Flags: A - aggregate, T - tracked
       L - Local, S - static, V - virtual, R - Reported through v3
       I - v3lite, U - Urd, M - SSM (S,G) channel
       1,2,3 - The version of IGMP, the group is in
Channel/Group-Flags:
/ - Filtering entry (Exclude mode (S,G), Include mode (G))
Reporter:
<mac-or-ip-address> - last reporter if group is not explicitly tracked
<n>/<m> - <n> reporter in include mode, <m> reporter in exclude

Channel/Group      Reporter      Uptime    Exp.    Flags    Interface
*,203.0.113.10     1/0          00:20:46  stop    3AT      Gi1/0/24
192.168.0.2,203.0.113.10  10.34.34.2  00:20:46  02:59  T        Gi1/0/24
*,203.0.113.11     1/0          00:20:46  stop    3AT      Gi1/0/24
192.168.0.2,203.0.113.11  10.34.34.2  00:20:46  02:59  T        Gi1/0/24
*,203.0.113.14     1/0          00:20:46  stop    3AT      Gi1/0/24
192.168.0.2,203.0.113.14  10.34.34.2  00:20:46  02:59  T        Gi1/0/24
*,203.0.113.15     1/0          00:20:46  stop    3AT      Gi1/0/24
192.168.0.2,203.0.113.15  10.34.34.2  00:20:46  02:59  T        Gi1/0/24
*,203.0.113.12     1/0          00:20:46  stop    3AT      Gi1/0/24
192.168.0.2,203.0.113.12  10.34.34.2  00:20:46  02:59  T        Gi1/0/24
*,203.0.113.13     1/0          00:20:46  stop    3AT      Gi1/0/24
192.168.0.2,203.0.113.13  10.34.34.2  00:20:46  02:59  T        Gi1/0/24
*,203.0.113.19     1/0          00:20:46  stop    3AT      Gi1/0/24
192.168.0.2,203.0.113.19  10.34.34.2  00:20:46  02:59  T        Gi1/0/24
```

```

*,203.0.113.18          1/0          00:20:46 stop 3AT   Gi1/0/24
192.168.0.2,203.0.113.18 10.34.34.2 00:20:46 02:59 T    Gi1/0/24
*,203.0.113.17          1/0          00:20:46 stop 3AT   Gi1/0/24
192.168.0.2,203.0.113.17 10.34.34.2 00:20:46 02:59 T    Gi1/0/24
*,203.0.113.16          1/0          00:20:46 stop 3AT   Gi1/0/24
192.168.0.2,203.0.113.16 10.34.34.2 00:20:46 02:59 T    Gi1/0/24
*,203.0.113.40          0/1          00:20:48 02:16 3LAT Gi1/0/24
*,209.165.201.1         10.34.34.1 00:20:48 02:16 3LT   Gi1/0/24
Device#

```

Step 5 **show ip igmp snooping vlan *vlan-ID***

Example:

```
Device# show ip igmp snooping vlan 77
```

Displays the IGMP snooping configuration on a VLAN.

```
Device# show ip igmp snooping vlan 77
```

```
Global IGMP Snooping configuration:
```

```

-----
IGMP snooping           : Enabled
IGMPv3 snooping         : Enabled
Report suppression      : Enabled
TCN solicit query       : Disabled
TCN flood query count   : 2
Robustness variable     : 2
Last member query count : 2
Last member query interval : 1000

```

```
Vlan 77:
```

```

-----
IGMP snooping           : Enabled
IGMPv2 immediate leave  : Disabled
Explicit host tracking   : Enabled
Multicast router learning mode : pim-dvmrp
CGMP interoperability mode : IGMP_ONLY
Robustness variable     : 2
Last member query count : 2
Last member query interval : 1000
Device#

```

Additional References for IGMP Explicit Tracking

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the IP Multicast Routing Commands section of the <i>Command Reference (Catalyst 9400 Series Switches)</i>

Feature History for IGMP Explicit Tracking

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Fuji 16.8.1a	IGMP Explicit Tracking	IGMP explicit tracking enables a multicast device to explicitly track the membership of all multicast hosts in a particular multiaccess network

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>.



CHAPTER 7

Constraining IP Multicast in Switched Ethernet

- [Prerequisites for Constraining IP Multicast in a Switched Ethernet Network, on page 115](#)
- [Information About IP Multicast in a Switched Ethernet Network, on page 115](#)
- [How to Constrain Multicast in a Switched Ethernet Network, on page 117](#)
- [Configuration Examples for Constraining IP Multicast in a Switched Ethernet Network, on page 119](#)
- [Additional References for Constraining IP Multicast in a Switched Ethernet Network, on page 120](#)
- [Feature History for Constraining IP Multicast in Switched Ethernet, on page 120](#)

Prerequisites for Constraining IP Multicast in a Switched Ethernet Network

Before using the tasks in this module, you should be familiar with the concepts described in the [IP Multicast Routing Technology Overview, on page 1](#) module.

Information About IP Multicast in a Switched Ethernet Network

IP Multicast Traffic and Layer 2 Switches

The default behavior for a Layer 2 switch is to forward all multicast traffic to every port that belongs to the destination LAN on the switch. This behavior reduces the efficiency of the switch, whose purpose is to limit traffic to the ports that need to receive the data. This behavior requires a constraining mechanism to reduce unnecessary multicast traffic, which improves switch performance.

Cisco Group Management Protocol (CGMP), Router Group Management Protocol (RGMP), and IGMP snooping efficiently constrain IP multicast in a Layer 2 switching environment.

- CGMP and IGMP snooping are used on subnets that include end users or receiver clients.
- RGMP is used on routed segments that contain only routers, such as in a collapsed backbone.
- RGMP and CGMP cannot interoperate. However, Internet Group Management Protocol (IGMP) can interoperate with CGMP and RGMP snooping.

CGMP on Catalyst Switches for IP Multicast

CGMP is a Cisco-developed protocol used on device connected to Catalyst switches to perform tasks similar to those performed by IGMP. CGMP is necessary for those Catalyst switches that do not distinguish between IP multicast data packets and IGMP report messages, both of which are addressed to the same group address at the MAC level. The switch can distinguish IGMP packets, but would need to use software on the switch, greatly impacting its performance.

You must configure CGMP on the multicast device and the Layer 2 switches. The result is that, with CGMP, IP multicast traffic is delivered only to those Catalyst switch ports that are attached to interested receivers. All other ports that have not explicitly requested the traffic will not receive it unless these ports are connected to a multicast router. Multicast router ports must receive every IP multicast data packet.

Using CGMP, when a host joins a multicast group, it multicasts an unsolicited IGMP membership report message to the target group. The IGMP report is passed through the switch to the router for normal IGMP processing. The router (which must have CGMP enabled on this interface) receives the IGMP report and processes it as it normally would, but also creates a CGMP Join message and sends it to the switch. The Join message includes the MAC address of the end station and the MAC address of the group it has joined.

The switch receives this CGMP Join message and then adds the port to its content-addressable memory (CAM) table for that multicast group. All subsequent traffic directed to this multicast group is then forwarded out the port for that host.

The Layer 2 switches are designed so that several destination MAC addresses could be assigned to a single physical port. This design allows switches to be connected in a hierarchy and also allows many multicast destination addresses to be forwarded out a single port.

The device port also is added to the entry for the multicast group. Multicast device must listen to all multicast traffic for every group because IGMP control messages are also sent as multicast traffic. The rest of the multicast traffic is forwarded using the CAM table with the new entries created by CGMP.

IGMP Snooping

IGMP snooping is an IP multicast constraining mechanism that runs on a Layer 2 LAN switch. IGMP snooping requires the LAN switch to examine, or “snoop,” some Layer 3 information (IGMP Join/Leave messages) in the IGMP packets sent between the hosts and the router. When the switch receives the IGMP host report from a host for a particular multicast group, the switch adds the port number of the host to the associated multicast table entry. When the switch hears the IGMP Leave group message from a host, the switch removes the table entry of the host.

Because IGMP control messages are sent as multicast packets, they are indistinguishable from multicast data at Layer 2. A switch running IGMP snooping must examine every multicast data packet to determine if it contains any pertinent IGMP control information. IGMP snooping implemented on a low-end switch with a slow CPU could have a severe performance impact when data is sent at high rates. The solution is to implement IGMP snooping on high-end switches with special application-specific integrated circuits (ASICs) that can perform the IGMP checks in hardware. CGMP is a better option for low-end switches without special hardware.

Router-Port Group Management Protocol (RGMP)

CGMP and IGMP snooping are IP multicast constraining mechanisms designed to work on routed network segments that have active receivers. They both depend on IGMP control messages that are sent between the hosts and the routers to determine which switch ports are connected to interested receivers.

Switched Ethernet backbone network segments typically consist of several routers connected to a switch without any hosts on that segment. Because routers do not generate IGMP host reports, CGMP and IGMP snooping will not be able to constrain the multicast traffic, which will be flooded to every port on the VLAN. Routers instead generate Protocol Independent Multicast (PIM) messages to Join and Prune multicast traffic flows at a Layer 3 level.

Router-Port Group Management Protocol (RGMP) is an IP multicast constraining mechanism for router-only network segments. RGMP must be enabled on the routers and on the Layer 2 switches. A multicast router indicates that it is interested in receiving a data flow by sending an RGMP Join message for a particular group. The switch then adds the appropriate port to its forwarding table for that multicast group--similar to the way it handles a CGMP Join message. IP multicast data flows will be forwarded only to the interested router ports. When the router no longer is interested in that data flow, it sends an RGMP Leave message and the switch removes the forwarding entry.

If there are any routers that are not RGMP-enabled, they will continue to receive all multicast data.

How to Constrain Multicast in a Switched Ethernet Network

Configuring Switches for IP Multicast

If you have switching in your multicast network, consult the documentation for the switch you are working with for information about how to configure IP multicast.

Configuring IGMP Snooping

No configuration is required on the router. Consult the documentation for the switch you are working with to determine how to enable IGMP snooping and follow the provided instructions.

Enabling CGMP

CGMP is a protocol used on devices connected to Catalyst switches to perform tasks similar to those performed by IGMP. CGMP is necessary because the Catalyst switch cannot distinguish between IP multicast data packets and IGMP report messages, which are both at the MAC level and are addressed to the same group address.



Note

- CGMP should be enabled only on 802 or ATM media, or LAN emulation (LANE) over ATM.
- CGMP should be enabled only on devices connected to Catalyst switches.

Procedure

	Command or Action	Purpose
Step 1	enable Example:	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	Device> enable	
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Device(config)# interface ethernet 1	Selects an interface that is connected to hosts on which IGMPv3 can be enabled.
Step 4	ip cgmp [proxy router-only] Example: Device(config-if)# ip cgmp proxy	Enables CGMP on an interface of a device connected to a Cisco Catalyst 5000 family switch. <ul style="list-style-type: none"> The proxy keyword enables the CGMP proxy function. When enabled, any device that is not CGMP-capable will be advertised by the proxy router. The proxy router advertises the existence of other non-CGMP-capable devices by sending a CGMP Join message with the MAC address of the non-CGMP-capable device and group address of 0000.0000.0000.
Step 5	end Example: Device(config-if)# end	Ends the current configuration session and returns to EXEC mode.
Step 6	clear ip cgmp [<i>interface-type interface-number</i>] Example: Device# clear ip cgmp	(Optional) Clears all group entries from the caches of Catalyst switches.

Configuring IP Multicast in a Layer 2 Switched Ethernet Network

Perform this task to configure IP multicast in a Layer 2 Switched Ethernet network using RGMP.

Procedure

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

	Command or Action	Purpose
	Device> enable	
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface type number Example: Device(config)# interface ethernet 1	Selects an interface that is connected to hosts.
Step 4	ip rgmp Example: Device(config-if)# ip rgmp	Enables RGMP on Ethernet, Fast Ethernet, and Gigabit Ethernet interfaces.
Step 5	end Example: Device(config-if)# end	Ends the current configuration session and returns to EXEC mode.
Step 6	debug ip rgmp Example: Device# debug ip rgmp	(Optional) Logs debug messages sent by an RGMP-enabled device.
Step 7	show ip igmp interface Example: Device# show ip igmp interface	(Optional) Displays multicast-related information about an interface.

Configuration Examples for Constraining IP Multicast in a Switched Ethernet Network

RGMP Configuration Example

The following example shows how to configure RGMP on a router:

```
ip multicast-routing
ip pim sparse-mode
interface ethernet 0
 ip rgmp
```

Additional References for Constraining IP Multicast in a Switched Ethernet Network

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the IP Multicast Routing Commands section of the <i>Command Reference (Catalyst 9400 Series Switches)</i> See the IP Multicast Routing Commands section of the <i>Command Reference (Catalyst 9600 Series Switches)</i>

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 History for Constraining IP Multicast in Switched Ethernet

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	Constraining IP Multicast in Switched Ethernet	IP multicast in switched ethernet provides a constraining mechanism to reduce unnecessary multicast traffic, which improves switch performance.

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>.



CHAPTER 8

Configuring Protocol Independent Multicast (PIM)

- [Prerequisites for PIM, on page 123](#)
- [Restrictions for PIM, on page 124](#)
- [Information about PIM, on page 126](#)
- [How to Configure PIM, on page 143](#)
- [Verifying PIM Operations, on page 169](#)
- [Monitoring and Troubleshooting PIM, on page 178](#)
- [Configuration Examples for PIM, on page 180](#)
- [Feature History for PIM, on page 183](#)

Prerequisites for PIM

Before you begin the PIM configuration process, decide which PIM mode to use. This is based on the applications you intend to support on your network. Use the following guidelines:

- In general, if the application is one-to-many or many-to-many in nature, then PIM-SM can be used successfully.
- For optimal one-to-many application performance, SSM is appropriate but requires IGMP version 3 support.

Before you configure PIM stub routing, check that you have met these conditions:

- You must have IP multicast routing configured on both the stub router and the central router. You must also have PIM mode configured on the uplink interface of the stub router.
- You must also configure either Enhanced Interior Gateway Routing Protocol (EIGRP) stub routing or Open Shortest Path First (OSPF) stub routing on the device.
- The PIM stub router does not route the transit traffic between the distribution routers. Unicast (EIGRP) stub routing enforces this behavior. You must configure unicast stub routing to assist the PIM stub router behavior.

Restrictions for PIM

The following are the restrictions for configuring PIM:

- Use ACLs to designate a specified port only as a multicast host port and not as a multicast router port. Multicast router control-packets received on this port are dropped.
- PIM nonbroadcast multiaccess (NBMA) mode is not supported on an ethernet interface.
- Hot Standby Router Protocol-aware (HSRP-aware) PIM is not supported.

PIMv1 and PIMv2 Interoperability

To avoid misconfiguring multicast routing on your device, review the information in this section.

The Cisco PIMv2 implementation provides interoperability and transition between Version 1 and Version 2, although there might be some minor problems.

You can upgrade to PIMv2 incrementally. PIM Versions 1 and 2 can be configured on different routers and multilayer switches within one network. Internally, all routers and multilayer switches on a shared media network must run the same PIM version. Therefore, if a PIMv2 device detects a PIMv1 device, the Version 2 device downgrades itself to Version 1 until all Version 1 devices have been shut down or upgraded.

PIMv2 uses the BSR to discover and announce RP-set information for each group prefix to all the routers and multilayer switches in a PIM domain. PIMv1, together with the Auto-RP feature, can perform the same tasks as the PIMv2 BSR. However, Auto-RP is a standalone protocol, separate from PIMv1, and is a proprietary Cisco protocol. PIMv2 is a standards track protocol in the IETF.



Note We recommend that you use PIMv2. The BSR function interoperates with Auto-RP on Cisco routers and multilayer switches.

When PIMv2 devices interoperate with PIMv1 devices, Auto-RP should have already been deployed. A PIMv2 BSR that is also an Auto-RP mapping agent automatically advertises the RP elected by Auto-RP. That is, Auto-RP sets its single RP on every router or multilayer switch in the group. Not all routers and switches in the domain use the PIMv2 hash function to select multiple RPs.

Sparse-mode groups in a mixed PIMv1 and PIMv2 region are possible because the Auto-RP feature in PIMv1 interoperates with the PIMv2 RP feature. Although all PIMv2 devices can also use PIMv1, we recommend that the RPs be upgraded to PIMv2. To ease the transition to PIMv2, we recommend:

- Using Auto-RP throughout the region.

If Auto-RP is not already configured in the PIMv1 regions, configure Auto-RP.

Restrictions for Configuring PIM Stub Routing

- Only directly connected multicast (IGMP) receivers and sources are allowed in the Layer 2 access domains. The PIM protocol is not supported in access domains.

- In a network using PIM stub routing, the only allowable route for IP traffic to the user is through a device that is configured with PIM stub routing.
- The redundant PIM stub router topology is not supported. Only the nonredundant access router topology is supported by the PIM stub feature.

Restrictions for Configuring Auto-RP and BSR

Take into consideration your network configuration, and the following restrictions when configuring Auto-RP and BSR:

Restrictions for Configuring Auto-RP

The following are restrictions for configuring Auto-RP (if used in your network configuration):

- If routed interfaces are configured in sparse mode, Auto-RP can still be used if all devices are configured with a manual RP address for the Auto-RP groups.
- If routed interfaces are configured in sparse mode and you enter the **ip pim autorp listener** global configuration command, Auto-RP can still be used even if all devices are not configured with a manual RP address for the Auto-RP groups.

Restrictions for Configuring BSR

The following are the restrictions for configuring BSR (if used in your network configuration):

- Configure the candidate BSRs as the RP-mapping agents for Auto-RP.
- For group prefixes advertised through Auto-RP, the PIMv2 BSR mechanism should not advertise a subrange of these group prefixes served by a different set of RPs. In a mixed PIMv1 and PIMv2 domain, have backup RPs serve the same group prefixes. This prevents the PIMv2 DRs from selecting a different RP from those PIMv1 DRs, due to the longest match lookup in the RP-mapping database.

Restrictions and Guidelines for Configuring Auto-RP and BSR

The following are restrictions for configuring Auto-RP and BSR (if used in your network configuration):

- If your network is all Cisco routers and multilayer switches, you can use either Auto-RP or BSR.
- If you have non-Cisco routers in your network, you must use BSR.
- If you have Cisco PIMv1 and PIMv2 routers and multilayer switches and non-Cisco routers, you must use both Auto-RP and BSR. If your network includes routers from other vendors, configure the Auto-RP mapping agent and the BSR on a Cisco PIMv2 device. Ensure that no PIMv1 device is located in the path a between the BSR and a non-Cisco PIMv2 device.



Note There are two approaches to using PIMv2. You can use Version 2 exclusively in your network or migrate to Version 2 by employing a mixed PIM version environment.

- Because bootstrap messages are sent hop-by-hop, a PIMv1 device prevents these messages from reaching all routers and multilayer switches in your network. Therefore, if your network has a PIMv1 device in it and only Cisco routers and multilayer switches, it is best to use Auto-RP.
- If you have a network that includes non-Cisco routers, configure the Auto-RP mapping agent and the BSR on a Cisco PIMv2 router or multilayer switch. Ensure that no PIMv1 device is on the path between the BSR and a non-Cisco PIMv2 router.
- If you have non-Cisco PIMv2 routers that need to interoperate with Cisco PIMv1 routers and multilayer switches, both Auto-RP and a BSR are required. We recommend that a Cisco PIMv2 device be both the Auto-RP mapping agent and the BSR.

Restrictions for Auto-RP Enhancement

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

Information about PIM

Protocol Independent Multicast Overview

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

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

PIM is defined in RFC 4601, Protocol Independent Multicast - Sparse Mode (PIM-SM)

PIM Versions

PIMv2 includes these improvements over PIMv1:

- A single, active rendezvous point (RP) exists per multicast group, with multiple backup RPs. This single RP compares to multiple active RPs for the same group in PIMv1.
- A bootstrap router (BSR) provides a fault-tolerant, automated RP discovery and distribution function that enables routers and multilayer switches to dynamically learn the group-to-RP mappings.
- PIM join and prune messages have more flexible encoding for multiple address families.
- A more flexible hello packet format replaces the query packet to encode current and future capability options.
- Register messages sent to an RP specify whether they are sent by a border router or a designated router.
- PIM packets are no longer inside IGMP packets; they are standalone packets.

Multicast Source Discovery Protocol (MSDP)

Multicast Source Discovery Protocol (MSDP) is used for inter-domain source discovery when PIM SM is used. Each PIM administrative domain has its own RP. In order for the RP in one domain to signal new sources to the RP in the other domain, MSDP is used.

When RP in a domain receives a PIM register message for a new source, with MSDP configured it sends a new source-active (SA) message to all its MSDP peers in other domains. Each intermediate MSDP peer floods this SA message away from the originating RP. The MSDP peers install this SA message in their MSDP sa-cache. If the RPs in other domains have any join requests for the group in the SA message (indicated by the presence of a (*,G) entry with non empty outgoing interface list), the domain is interested in the group, and the RP triggers an (S,G) join toward the source.

PIM Sparse Mode

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

Sparse mode interfaces are added to the multicast routing table only when periodic Join messages are received from downstream routers, or when a directly connected member is on the interface. When forwarding from a LAN, sparse mode operation occurs if an RP is known for the group. If so, the packets are encapsulated and sent toward the RP. If the multicast traffic from a specific source is sufficient, the first hop router of the receiver may send Join messages toward the source to build a source-based distribution tree.

PIM-SM distributes information about active sources by forwarding data packets on the shared tree. Because PIM-SM uses shared trees (at least, initially), it requires the use of a rendezvous point (RP). The RP must be administratively configured in the network. See the [Rendezvous Points, on page 132](#) section for more information.

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

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

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

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

Bidirectional PIM

Bidirectional PIM is a variant of the PIM suite of routing protocols for IP multicast. In PIM, packet traffic for a multicast group is routed according to the rules of the mode configured for that multicast group.

In bidirectional mode, traffic is routed only along a bidirectional shared tree that is rooted at the rendezvous point (RP) for the group. In bidir-PIM, the IP address of the RP acts as the key to having all routers establish a loop-free spanning tree topology rooted in that IP address. This IP address need not be a router, but can be any unassigned IP address on a network that is reachable throughout the PIM domain. This technique is the preferred configuration method for establishing a redundant RP configuration for bidir-PIM.

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

Bidir-PIM is designed to be used for many-to-many applications within individual PIM domains. Multicast groups in bidirectional mode can scale to an arbitrary number of sources without incurring overhead due to the number of sources.

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

Bidir-PIM is derived from the mechanisms of PIM-SM and shares many shortest-path tree (SPT) operations. Bidir-PIM also has unconditional forwarding of source traffic toward the RP upstream on the shared tree, but no registering process for sources as in PIM-SM. These modifications are necessary and sufficient to allow forwarding of traffic in all routers solely based on the (*, G) multicast routing entries. This feature eliminates any source-specific state and allows scaling capability to an arbitrary number of sources. The following figures show the difference in state created per router for a unidirectional shared tree and source tree versus a bidirectional shared tree.

Figure 10: Unidirectional Shared Tree and Source Tree

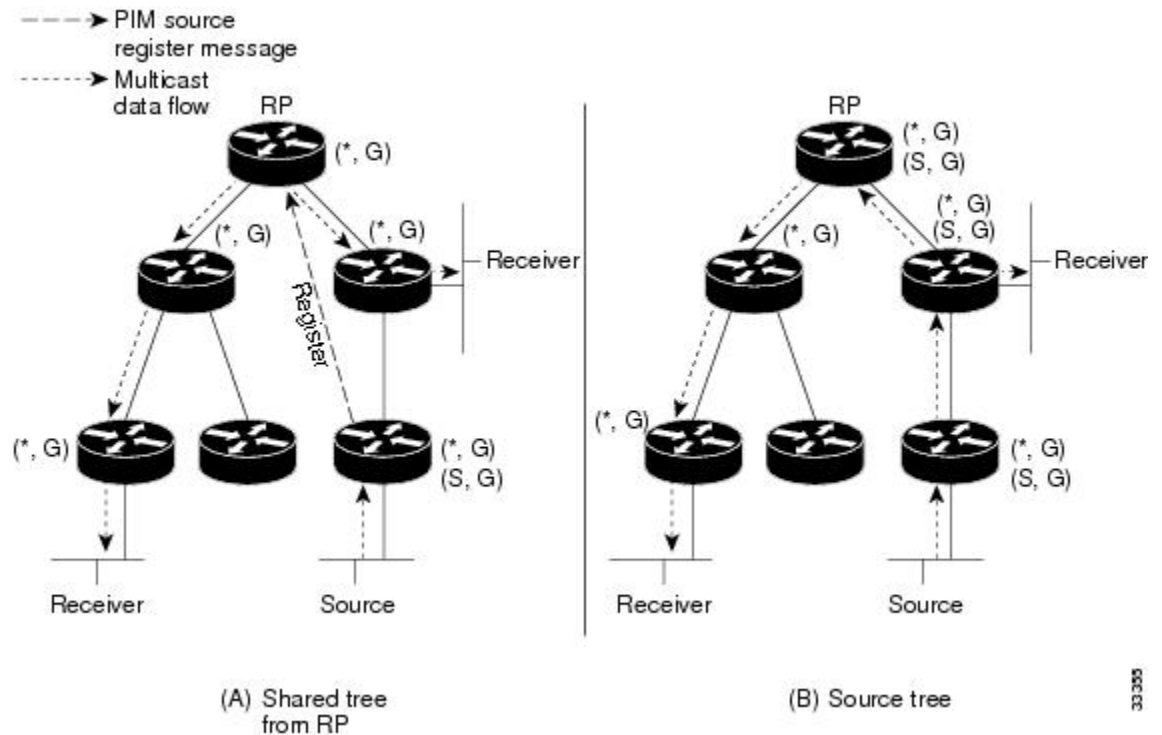
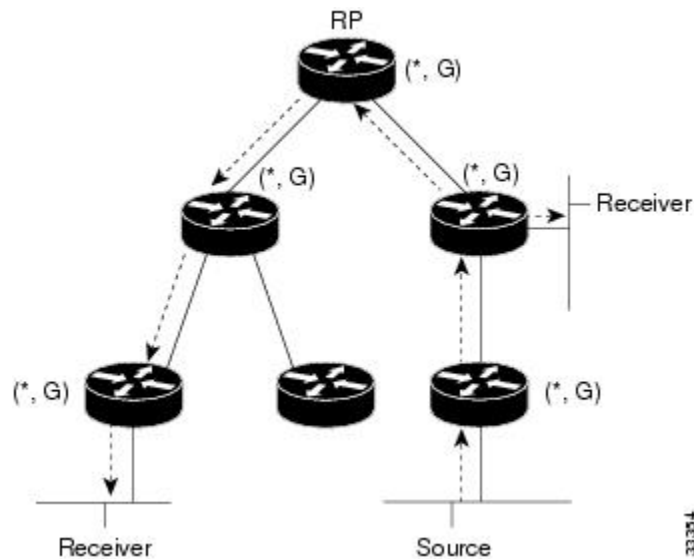


Figure 11: Bidirectional Shared Tree



When packets are forwarded downstream from the RP toward receivers, there are no fundamental differences between bidir-PIM and PIM-SM. Bidir-PIM deviates substantially from PIM-SM when passing traffic from sources upstream toward the RP.

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

Designated Forwarder Election

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

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

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

Bidirectional Group Tree Building

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

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

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

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

Packet Forwarding

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

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

IPv4 Bidirectional PIM

For Bidirectional PIM to be operational, designated forwarder is required. The designated forwarder is the router elected to forward packets to and from a segment for a IPv4 bidirectional PIM group. In DF mode, the switch accepts packets from the RPF and from the DF interfaces.

When the switch is forwarding IPv4 bidirectional PIM groups, the RPF interface is always included in the outgoing interface list of (*,G) entry, and the DF interfaces are included depending on IGMP/PIM joins.

If the route to the RP becomes unavailable, the group is changed to dense mode. Should the RPF link to the RP become unavailable, the IPv4 bidirectional PIM flow is removed from the hardware FIB.

PIM Stub Routing

The PIM stub routing feature, available in all of the device software images, reduces resource usage by moving routed traffic closer to the end user.

The PIM stub routing feature supports multicast routing between the distribution layer and the access layer. It supports two types of PIM interfaces, uplink PIM interfaces, and PIM passive interfaces. A routed interface configured with the PIM passive mode does not pass or forward PIM control traffic, it only passes and forwards IGMP traffic.

In a network using PIM stub routing, the only allowable route for IP traffic to the user is through a device that is configured with PIM stub routing. PIM passive interfaces are connected to Layer 2 access domains, such as VLANs, or to interfaces that are connected to other Layer 2 devices. Only directly connected multicast (IGMP) receivers and sources are allowed in the Layer 2 access domains. The PIM passive interfaces do not send or process any received PIM control packets.

When using PIM stub routing, you should configure the distribution and remote routers to use IP multicast routing and configure only the device as a PIM stub router. The device does not route transit traffic between distribution routers. You also need to configure a routed uplink port on the device. The device uplink port cannot be used with SVIs. If you need PIM for an SVI uplink port, you should upgrade to the Network Advantage license.

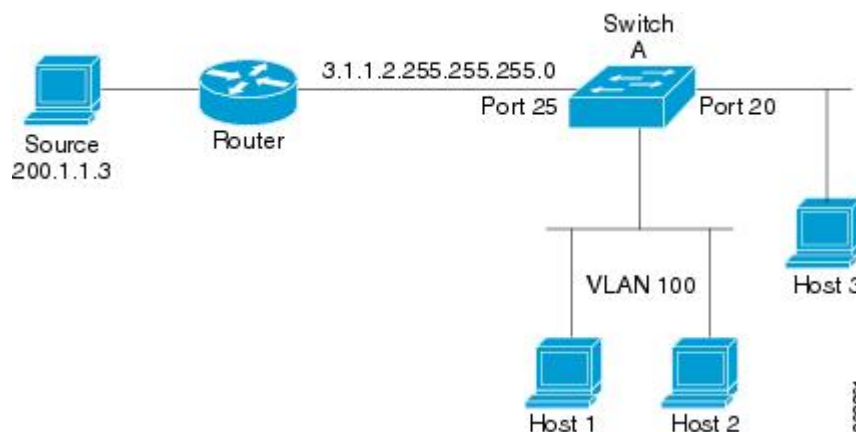


Note You must also configure EIGRP stub routing when configuring PIM stub routing on the device

The redundant PIM stub router topology is not supported. The redundant topology exists when there is more than one PIM router forwarding multicast traffic to a single access domain. PIM messages are blocked, and the PIM asset and designated router election mechanisms are not supported on the PIM passive interfaces. Only the nonredundant access router topology is supported by the PIM stub feature. By using a nonredundant topology, the PIM passive interface assumes that it is the only interface and designated router on that access domain.

Figure 12: PIM Stub Router Configuration

In the following figure, the Device A routed uplink port 25 is connected to the router and PIM stub routing is enabled on the VLAN 100 interfaces and on Host 3. This configuration allows the directly connected hosts to receive traffic from multicast source 200.1.1.3.



Rendezvous Points

A rendezvous point (RP) is a role that a device performs when operating in Protocol Independent Multicast (PIM) Sparse Mode (SM). An RP is required only in networks running PIM SM. In the PIM-SM model, only network segments with active receivers that have explicitly requested multicast data will be forwarded the traffic.

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

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

Auto-RP

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

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

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

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



Note If router interfaces are configured in sparse mode, Auto-RP can still be used if all routers are configured with a static RP address for the Auto-RP groups.

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

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

The Role of Auto-RP in a PIM Network

Auto-RP automates the distribution of group-to-rendezvous point (RP) mappings in a PIM network. To make Auto-RP work, a device must be designated as an RP mapping agent, which receives the RP announcement messages from the RPs and arbitrates conflicts.

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

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

Multicast Boundaries

Administratively-scoped boundaries can be used to limit the forwarding of multicast traffic outside of a domain or subdomain. This approach uses a special range of multicast addresses, called administratively-scoped addresses, as the boundary mechanism. If you configure an administratively-scoped boundary on a routed interface, multicast traffic whose multicast group addresses fall in this range cannot enter or exit this interface, which provides a firewall for multicast traffic in this address range.

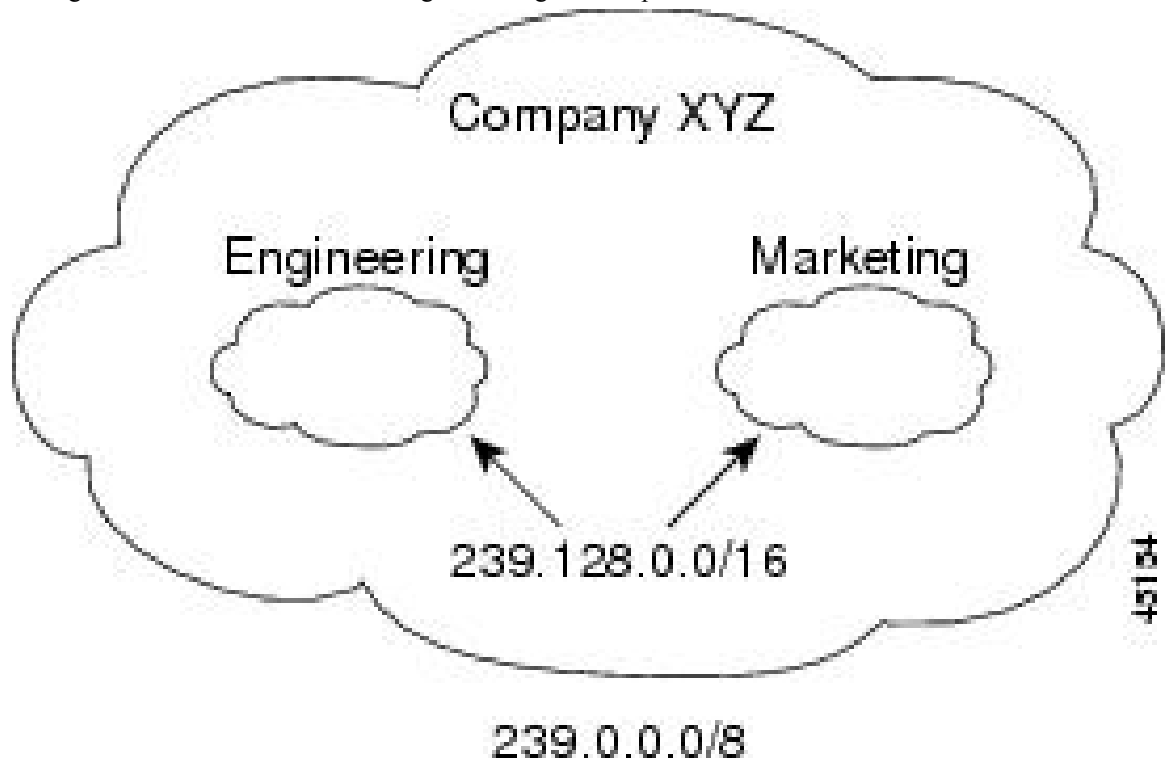


Note Multicast boundaries and TTL thresholds control the scoping of multicast domains; however, TTL thresholds are not supported by the device. You should use multicast boundaries instead of TTL thresholds to limit the forwarding of multicast traffic outside of a domain or a subdomain.

Figure 13: Administratively-Scoped Boundaries

The following figure shows that Company XYZ has an administratively-scoped boundary set for the multicast address range 239.0.0.0/8 on all routed interfaces at the perimeter of its network. This boundary prevents any multicast traffic in the range 239.0.0.0 through 239.255.255.255 from entering or leaving the network. Similarly, the engineering and marketing departments have an administratively-scoped boundary of 239.128.0.0/16 around the perimeter of their networks. This boundary prevents multicast traffic in the range of 239.128.0.0

through 239.128.255.255 from entering or leaving their respective networks.



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

The IANA has designated the multicast address range 239.0.0.0 to 239.255.255.255 as the administratively-scoped addresses. This range of addresses can then be reused in domains administered by different organizations. The addresses would be considered local, not globally unique.

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

Sparse-Dense Mode for Auto-RP

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

To successfully implement Auto-RP and prevent any groups other than 224.0.1.39 and 224.0.1.40 from operating in dense mode, we recommend configuring a “sink RP” (also known as “RP of last resort”). A sink

RP is a statically configured RP that may or may not actually exist in the network. Configuring a sink RP does not interfere with Auto-RP operation because, by default, Auto-RP messages supersede static RP configurations. We recommend configuring a sink RP for all possible multicast groups in your network, because it is possible for an unknown or unexpected source to become active. If no RP is configured to limit source registration, the group may revert to dense mode operation and be flooded with data.

Auto RP Benefits

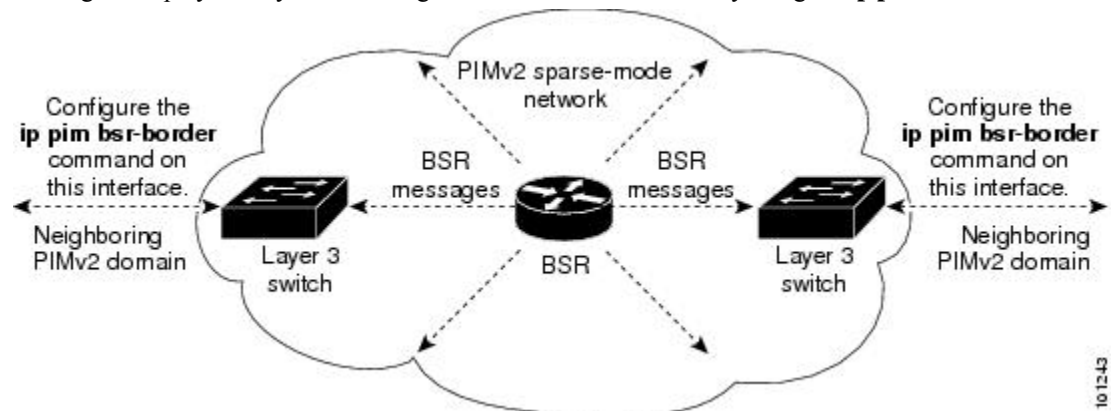
Benefits of Auto-RP in a PIM Network

- Auto-RP allows any change to the RP designation to be configured only on the devices that are RPs, not on the leaf routers.
- Auto-RP offers the ability to scope the RP address within a domain.

PIM Domain Border

As IP multicast becomes more widespread, the chance of one PIMv2 domain bordering another PIMv2 domain increases. Because two domains probably do not share the same set of RPs, BSR, candidate RPs, and candidate BSRs, you need to constrain PIMv2 BSR messages from flowing into or out of the domain. Allowing messages to leak across the domain borders could adversely affect the normal BSR election mechanism and elect a single BSR across all bordering domains and comingle candidate RP advertisements, resulting in the election of RPs in the wrong domain.

This figure displays how you can configure the PIM domain border by using the **ip pim bsr-border** command.



PIMv2 Bootstrap Router

PIMv2 Bootstrap Router (BSR) is another method to distribute group-to-RP mapping information to all PIM routers and multilayer devices in the network. It eliminates the need to manually configure RP information in every router and switch in the network. However, instead of using IP multicast to distribute group-to-RP mapping information, BSR uses hop-by-hop flooding of special BSR messages to distribute the mapping information.

The BSR is elected from a set of candidate routers and switches in the domain that have been configured to function as BSRs. The election mechanism is similar to the root-bridge election mechanism used in bridged LANs. The BSR election is based on the BSR priority of the device contained in the BSR messages that are sent hop-by-hop through the network. Each BSR device examines the message and forwards out all interfaces only the message that has either a higher BSR priority than its BSR priority or the same BSR priority, but with a higher BSR IP address. Using this method, the BSR is elected.

The elected BSR sends BSR messages with a TTL of 1. Neighboring PIMv2 routers or multilayer devices receive the BSR message and multicast it out all other interfaces (except the one on which it was received) with a TTL of 1. In this way, BSR messages travel hop-by-hop throughout the PIM domain. Because BSR messages contain the IP address of the current BSR, the flooding mechanism enables candidate RPs to automatically learn which device is the elected BSR.

Candidate RPs send candidate RP advertisements showing the group range for which they are responsible to the BSR, which stores this information in its local candidate-RP cache. The BSR periodically advertises the contents of this cache in BSR messages to all other PIM devices in the domain. These messages travel hop-by-hop through the network to all routers and switches, which store the RP information in the BSR message in their local RP cache. The routers and switches select the same RP for a given group because they all use a common RP hashing algorithm.

Multicast Forwarding

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

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

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

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

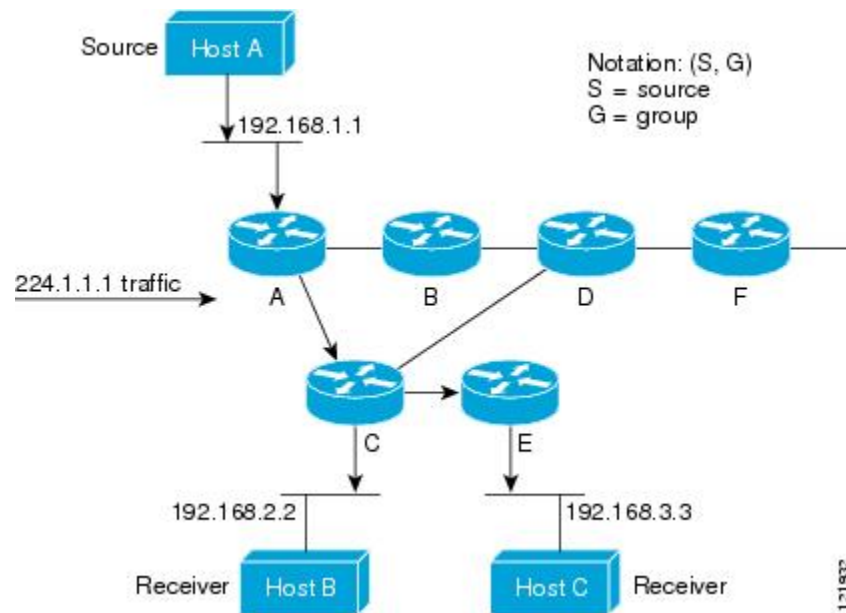
The notation of (S,G), pronounced “S comma G,” enumerates a shortest path tree where S is the IP address of the source and G is the multicast group address.

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

Multicast Distribution Source Tree

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

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



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

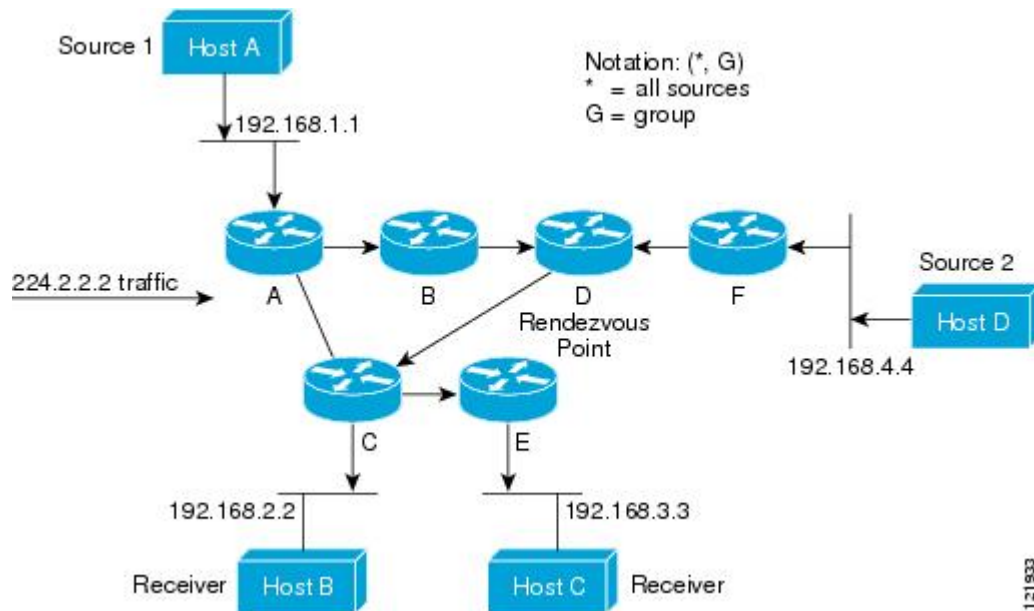
The (S,G) notation implies that a separate SPT exists for each individual source sending to each group--which is correct.

Multicast Distribution Shared Tree

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

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

Figure 14: Shared Tree



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

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

Source Tree Advantage

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

Shared Tree Advantage

Shared trees have the advantage of requiring the minimum amount of state in each router. This advantage lowers the overall memory requirements for a network that only allows shared trees. The disadvantage of shared trees is that under certain circumstances the paths between the source and receivers might not be the optimal paths, which might introduce some latency in packet delivery. For example, in the figure above the shortest path between Host A (source 1) and Host B (a receiver) would be Router A and Router C. Because we are using Router D as the root for a shared tree, the traffic must traverse Routers A, B, D and then C.

Network designers must carefully consider the placement of the rendezvous point (RP) when implementing a shared tree-only environment.

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

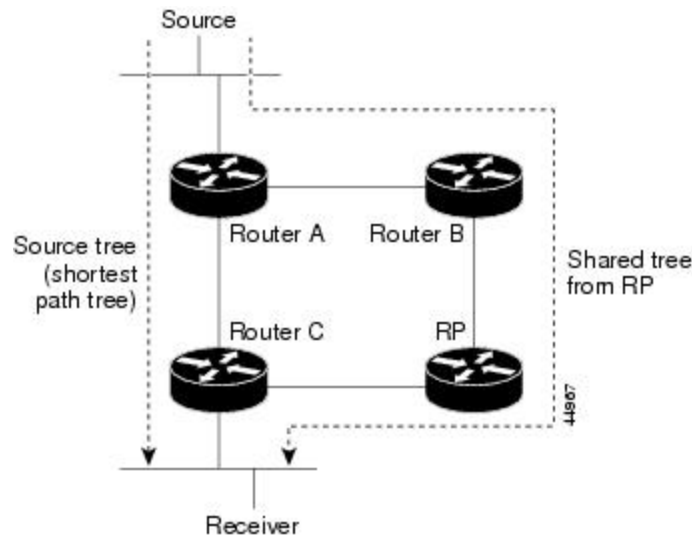
In multicast forwarding, the source is sending traffic to an arbitrary group of hosts that are represented by a multicast group address. The multicast router must determine which direction is the upstream direction (toward the source) and which one is the downstream direction (or directions) toward the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

PIM Shared Tree and Source Tree

By default, members of a group receive data from senders to the group across a single data-distribution tree rooted at the RP.

Figure 15: Shared Tree and Source Tree (Shortest-Path Tree)

The following figure shows this type of shared-distribution tree. Data from senders is delivered to the RP for distribution to group members joined to the shared tree.



If the data rate warrants, leaf routers (routers without any downstream connections) on the shared tree can use the data distribution tree rooted at the source. This type of distribution tree is called a shortest-path tree or source tree. By default, the software device to a source tree upon receiving the first data packet from a source.

This process describes the move from a shared tree to a source tree:

1. A receiver joins a group; leaf Router C sends a join message toward the RP.
2. The RP puts a link to Router C in its outgoing interface list.
3. A source sends data; Router A encapsulates the data in a register message and sends it to the RP.

4. The RP forwards the data down the shared tree to Router C and sends a join message toward the source. At this point, data might arrive twice at Router C, once encapsulated and once natively.
5. When data arrives natively (unencapsulated) at the RP, it 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 device 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. You can configure the PIM device to stay on the shared tree.

The change from shared to source tree happens when the first data packet arrives at the last-hop router. This change depends upon the threshold that is configured by using the **ip pim spt-threshold** global configuration command.

The shortest-path tree requires more memory than the shared tree but reduces delay. You may want to postpone its use. Instead of allowing the leaf router to immediately move to the shortest-path tree, you can specify that the traffic must first reach a threshold.

You can configure when a PIM leaf router should join the shortest-path tree for a specified group. If a source sends at a rate greater than or equal to the specified kbps rate, the multilayer switch triggers a PIM join message toward the source to construct a source tree (shortest-path tree). If the traffic rate from the source drops below the threshold value, the leaf router switches back to the shared tree and sends a prune message toward the source.

You can specify to which groups the shortest-path tree threshold applies by using a group list (a standard access list). If a value of 0 is specified or if the group list is not used, the threshold applies to all groups.

Reverse Path Forwarding

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

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

Protocol Independent Multicast (PIM) uses the unicast routing information to create a distribution tree along the reverse path from the receivers towards the source. The multicast routers then forward packets along the distribution tree from the source to the receivers. RPF is a key concept in multicast forwarding. It enables

routers to correctly forward multicast traffic down the distribution tree. RPF makes use of the existing unicast routing table to determine the upstream and downstream neighbors. A router will forward a multicast packet only if it is received on the upstream interface. This RPF check helps to guarantee that the distribution tree will be loop-free.

RPF Check

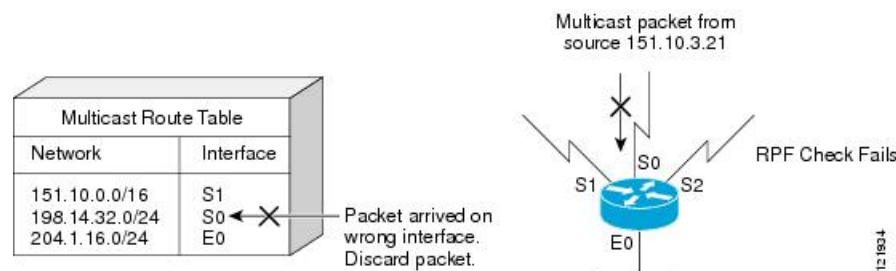
When a multicast packet arrives at a router, the router performs an RPF check on the packet. If the RPF check succeeds, the packet is forwarded. Otherwise, it is dropped.

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

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

The figure shows an example of an unsuccessful RPF check.

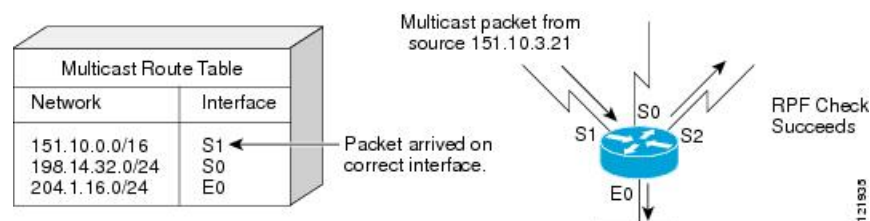
Figure 16: RPF Check Fails



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

The figure shows an example of a successful RPF check.

Figure 17: RPF Check Succeeds



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

PIM uses both source trees and RP-rooted shared trees to forward datagrams. The RPF check is performed differently for each:

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



Note DVMRP is not supported on the switch.

Sparse-mode PIM uses the RPF lookup function to decide where it needs to send joins and prunes:

- (S, G) joins (which are source-tree states) are sent toward the source.
- (*,G) joins (which are shared-tree states) are sent toward the RP.

Default PIM Routing Configuration

This table displays the default PIM routing configuration for the device.

Table 14: Default Multicast Routing Configuration

Feature	Default Setting
Multicast routing	Disabled on all interfaces.
PIM version	Version 2.
PIM mode	No mode is defined.
PIM stub routing	None configured.
PIM RP address	None configured.
PIM domain border	Disabled.
PIM multicast boundary	None.
Candidate BSRs	Disabled.
Candidate RPs	Disabled.
Shortest-path tree threshold rate	0 kb/s.
PIM router query message interval	30 seconds.

How to Configure PIM

Enabling PIM Stub Routing

This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: <pre>Device(config)# interface gigabitethernet 1/0/1</pre>	Specifies the interface on which you want to enable PIM stub routing, and enters interface configuration mode. The specified interface must be one of the following: These interfaces must have IP addresses assigned to them. <ul style="list-style-type: none"> • A routed port—A physical port that has been configured as a Layer 3 port by entering the no switchport interface configuration command. • An SVI—A VLAN interface created by using the interface vlan <i>vlan-id</i> global configuration command.
Step 4	ip pim passive Example: <pre>Device(config-if)# ip pim passive</pre>	Configures the PIM stub feature on the interface.
Step 5	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 6	show ip pim interface Example: <pre>Device# show ip pim interface</pre>	(Optional) Displays the PIM stub that is enabled on each interface.
Step 7	show ip igmp groups detail Example: <pre>Device# show ip igmp groups detail</pre>	(Optional) Displays the interested clients that have joined the specific multicast source group.
Step 8	show ip mroute Example: <pre>Device# show ip mroute</pre>	(Optional) Displays the IP multicast routing table.
Step 9	show running-config Example: <pre>Device# show running-config</pre>	Verifies your entries.
Step 10	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Configuring a Rendezvous Point

You must have a rendezvous point (RP), if the interface is in sparse-dense mode and if you want to handle the group as a sparse group. You can use these methods:

- By manually assigning an RP to multicast groups.
- As a standalone, Cisco-proprietary protocol separate from PIMv1, which includes:
- By using a standards track protocol in the Internet Engineering Task Force (IETF), which includes configuring PIMv2 BSR .



Note You can use Auto-RP, BSR, or a combination of both, depending on the PIM version that you are running and the types of routers in your network. For information about working with different PIM versions in your network, see [PIMv1 and PIMv2 Interoperability](#), on page 124.

Manually Assigning an RP to Multicast Groups

If the rendezvous point (RP) for a group is learned through a dynamic mechanism (such as Auto-RP or BSR), you need not perform this task for that RP.

Senders of multicast traffic announce their existence through register messages received from the source first-hop router (designated router) and forwarded to the RP. Receivers of multicast packets use RPs to join a multicast group by using explicit join messages.



Note RPs are not members of the multicast group; they serve as a *meeting place* for multicast sources and group members.

You can configure a single RP for multiple groups defined by an access list. If there is no RP configured for a group, the multilayer switch responds to the group as dense and uses the dense-mode PIM techniques.

This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ip pim rp-address <i>ip-address</i> [<i>access-list-number</i>] [<i>override</i>] Example: <pre>Device(config)# ip pim rp-address 10.1.1.1 20 override</pre>	Configures the address of a PIM RP. By default, no PIM RP address is configured. You must configure the IP address of RPs on all routers and multilayer switches (including the RP). Note If there is no RP configured for a group, the device treats the group as dense, using the dense-mode PIM techniques. A PIM device can be an RP for more than one group. Only one RP address can be used at a time within a PIM domain. The access list conditions specify for which groups the device is an RP. <ul style="list-style-type: none"> For <i>ip-address</i>, enter the unicast address of the RP in dotted-decimal notation.

	Command or Action	Purpose
		<ul style="list-style-type: none"> • (Optional) For <i>access-list-number</i>, enter an IP standard access list number from 1 to 99. If no access list is configured, the RP is used for all groups. • (Optional) The override keyword indicates that if there is a conflict between the RP configured with this command and one learned by Auto-RP or BSR, the RP configured with this command prevails.
Step 4	access-list <i>access-list-number</i> { deny permit } <i>source</i> [<i>source-wildcard</i>] Example: <pre>Device(config)# access-list 25 permit 10.5.0.1 255.224.0.0</pre>	<p>Creates a standard access list, repeating the command as many times as necessary.</p> <ul style="list-style-type: none"> • For <i>access-list-number</i>, enter the access list number specified in Step 2. • The deny keyword denies access if the conditions are matched. • The permit keyword permits access if the conditions are matched. • For <i>source</i>, enter the multicast group address for which the RP should be used. • (Optional) For <i>source-wildcard</i>, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore. <p>The access list is always terminated by an implicit deny statement for everything.</p>
Step 5	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 6	show running-config Example: <pre>Device# show running-config</pre>	Verifies your entries.
Step 7	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Setting Up Auto-RP in a New Internetwork



Note Omit Step 3 in the following procedure, if you want to configure a PIM router as the RP for the local group.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	show running-config Example: <pre>Device# show running-config</pre>	Verifies that a default RP is already configured on all PIM devices and the RP in the sparse-mode network. It was previously configured with the ip pim rp-address global configuration command. Note This step is not required for sparse-dense-mode environments. The selected RP should have good connectivity and be available across the network. Use this RP for the global groups (for example, 224.x.x.x and other global groups). Do not reconfigure the group address range that this RP serves. RPs dynamically discovered through Auto-RP take precedence over statically configured RPs. Assume that it is desirable to use a second RP for the local groups.
Step 3	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 4	ip pim send-rp-announce interface-id scope ttl group-list access-list-number interval seconds Example: <pre>Device(config)# ip pim send-rp-announce gigabitethernet 1/0/5 scope 20 group-list 10 interval 120</pre>	Configures another PIM device to be the candidate RP for local groups. <ul style="list-style-type: none"> For <i>interface-id</i>, enter the interface type and number that identifies the RP address. Valid interfaces include physical ports, port channels, and VLANs. For scope <i>ttl</i>, specify the time-to-live value in hops. Enter a hop count that is

	Command or Action	Purpose
		<p>high enough so that the RP-announce messages reach all mapping agents in the network. There is no default setting. The range is 1 to 255.</p> <ul style="list-style-type: none"> For group-list <i>access-list-number</i>, enter an IP standard access list number from 1 to 99. If no access list is configured, the RP is used for all groups. For interval <i>seconds</i>, specify how often the announcement messages must be sent. The default is 60 seconds. The range is 1 to 16383.
Step 5	<p>access-list <i>access-list-number</i> {deny permit} <i>source</i> [<i>source-wildcard</i>]</p> <p>Example:</p> <pre>Device(config)# access-list 10 permit 10.10.0.0</pre>	<p>Creates a standard access list, repeating the command as many times as necessary.</p> <ul style="list-style-type: none"> For <i>access-list-number</i>, enter the access list number specified in Step 3. The deny keyword denies access if the conditions are matched. The permit keyword permits access if the conditions are matched. For <i>source</i>, enter the multicast group address range for which the RP should be used. (Optional) For <i>source-wildcard</i>, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore. <p>Note Recall that the access list is always terminated by an implicit deny statement for everything.</p>
Step 6	<p>ip pim send-rp-discovery scope <i>tvl</i></p> <p>Example:</p> <pre>Device(config)# ip pim send-rp-discovery scope 50</pre>	<p>Finds a device whose connectivity is not likely to be interrupted, and assign it the role of RP-mapping agent.</p> <p>For scope <i>tvl</i>, specify the time-to-live value in hops to limit the RP discovery packets. All devices within the hop count from the source device receive the Auto-RP discovery messages. These messages tell other devices which group-to-RP mapping to use to avoid conflicts (such as overlapping group-to-RP</p>

	Command or Action	Purpose
		ranges). There is no default setting. The range is 1 to 255.
Step 7	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 8	show running-config Example: <pre>Device# show running-config</pre>	Verifies your entries.
Step 9	show ip pim rp mapping Example: <pre>Device# show ip pim rp mapping</pre>	Displays active RPs that are cached with associated multicast routing entries.
Step 10	show ip pim rp Example: <pre>Device# show ip pim rp</pre>	Displays the information cached in the routing table.
Step 11	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Adding Auto-RP to an Existing Sparse-Mode Cloud

This section contains suggestions for the initial deployment of Auto-RP into an existing sparse-mode cloud to minimize disruption of the existing multicast infrastructure.

This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
Step 2	show running-config Example: <pre>Device# show running-config</pre>	<p>Verifies that a default RP is already configured on all PIM devices and the RP in the sparse-mode network. It was previously configured with the ip pim rp-address global configuration command.</p> <p>Note This step is not required for sparse-dense-mode environments.</p> <p>The selected RP should have good connectivity and be available across the network. Use this RP for the global groups (for example, 224.x.x.x and other global groups). Do not reconfigure the group address range that this RP serves. RPs dynamically discovered through Auto-RP take precedence over statically configured RPs. Assume that it is desirable to use a second RP for the local groups.</p>
Step 3	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 4	ip pim send-rp-announce interface-id scope ttl group-list access-list-number interval seconds Example: <pre>Device(config)# ip pim send-rp-announce gigabitethernet 1/0/5 scope 20 group-list 10 interval 120</pre>	<p>Configures another PIM device to be the candidate RP for local groups.</p> <ul style="list-style-type: none"> For <i>interface-id</i>, enter the interface type and number that identifies the RP address. Valid interfaces include physical ports, port channels, and VLANs. For scope <i>ttl</i>, specify the time-to-live value in hops. Enter a hop count that is high enough so that the RP-announce messages reach all mapping agents in the network. There is no default setting. The range is 1 to 255. For group-list <i>access-list-number</i>, enter an IP standard access list number from 1 to 99. If no access list is configured, the RP is used for all groups. For interval <i>seconds</i>, specify how often the announcement messages must be sent. The default is 60 seconds. The range is 1 to 16383.

	Command or Action	Purpose
Step 5	<p>access-list <i>access-list-number</i> {deny permit} <i>source</i> [<i>source-wildcard</i>]</p> <p>Example:</p> <pre>Device(config)# access-list 10 permit 224.0.0.0 15.255.255.255</pre>	<p>Creates a standard access list, repeating the command as many times as necessary.</p> <ul style="list-style-type: none"> For <i>access-list-number</i>, enter the access list number specified in Step 3. The deny keyword denies access if the conditions are matched. The permit keyword permits access if the conditions are matched. For <i>source</i>, enter the multicast group address range for which the RP should be used. (Optional) For <i>source-wildcard</i>, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore. <p>Recall that the access list is always terminated by an implicit deny statement for everything.</p>
Step 6	<p>ip pim send-rp-discovery scope <i>ttl</i></p> <p>Example:</p> <pre>Device(config)# ip pim send-rp-discovery scope 50</pre>	<p>Finds a device whose connectivity is not likely to be interrupted, and assigns it the role of RP-mapping agent.</p> <p>For scope <i>ttl</i>, specify the time-to-live value in hops to limit the RP discovery packets. All devices within the hop count from the source device receive the Auto-RP discovery messages. These messages tell other devices which group-to-RP mapping to use to avoid conflicts (such as overlapping group-to-RP ranges). There is no default setting. The range is 1 to 255.</p> <p>Note To remove the device as the RP-mapping agent, use the no ip pim send-rp-discovery global configuration command.</p>
Step 7	<p>end</p> <p>Example:</p> <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 8	<p>show running-config</p> <p>Example:</p>	Verifies your entries.

	Command or Action	Purpose
	Device# show running-config	
Step 9	show ip pim rp mapping Example: Device# show ip pim rp mapping	Displays active RPs that are cached with associated multicast routing entries.
Step 10	show ip pim rp Example: Device# show ip pim rp	Displays the information cached in the routing table.
Step 11	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Preventing Join Messages to False RPs

Determine whether the **ip pim accept-rp** command was previously configured throughout the network by using the **show running-config** privileged EXEC command. If the **ip pim accept-rp** command is not configured on any device, this problem can be addressed later. In those routers or multilayer switches already configured with the **ip pim accept-rp** command, you must enter the command again to accept the newly advertised RP.

Filtering Incoming RP Announcement Messages

You can add configuration commands to the mapping agents to prevent a maliciously configured router from masquerading as a candidate RP and causing problems.

This procedure is optional.

Procedure

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

	Command or Action	Purpose
	Device# <code>configure terminal</code>	
Step 3	<p>ip pim rp-announce-filter rp-list access-list-number group-list access-list-number</p> <p>Example:</p> <pre>Device(config)# ip pim rp-announce-filter rp-list 10 group-list 14</pre>	<p>Filters incoming RP announcement messages.</p> <p>Enter this command on each mapping agent in the network. Without this command, all incoming RP-announce messages are accepted by default.</p> <p>For rp-list access-list-number, configure an access list of candidate RP addresses that, if permitted, is accepted for the group ranges supplied in the group-list access-list-number variable. If this variable is omitted, the filter applies to all multicast groups.</p> <p>If more than one mapping agent is used, the filters must be consistent across all mapping agents to ensure that no conflicts occur in the group-to-RP mapping information.</p>
Step 4	<p>access-list access-list-number {deny permit} source [source-wildcard]</p> <p>Example:</p> <pre>Device(config)# access-list 10 permit 10.8.1.0 255.255.224.0</pre>	<p>Creates a standard access list, repeating the command as many times as necessary.</p> <ul style="list-style-type: none"> • For <i>access-list-number</i>, enter the access list number specified in Step 2. • The deny keyword denies access if the conditions are matched. • The permit keyword permits access if the conditions are matched. • Create an access list that specifies from which routers and multilayer switches the mapping agent accepts candidate RP announcements (rp-list ACL). • Create an access list that specifies the range of multicast groups from which to accept or deny (group-list ACL). • For <i>source</i>, enter the multicast group address range for which the RP should be used. • (Optional) For <i>source-wildcard</i>, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore. <p>The access list is always terminated by an implicit deny statement for everything.</p>

	Command or Action	Purpose
Step 5	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 6	show running-config Example: <pre>Device# show running-config</pre>	Verifies your entries.
Step 7	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Configuring PIMv2 BSR

The process for configuring PIMv2 BSR may involve the following optional tasks:

- Defining the PIM domain border
- Defining the IP multicast boundary
- Configuring candidate BSRs
- Configuring candidate RPs

Defining the PIM Domain Border

Perform the following steps to configure the PIM domain border. This procedure is optional.

Procedure

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

	Command or Action	Purpose
	Device# <code>configure terminal</code>	
Step 3	interface <i>interface-id</i> Example: Device(config)# interface <code>gigabitethernet 1/0/1</code>	<p>Specifies the interface to be configured, and enters interface configuration mode.</p> <p>The specified interface must be one of the following:</p> <ul style="list-style-type: none"> • A routed port—A physical port that has been configured as a Layer 3 port by entering the no switchport interface configuration command. • An SVI—A VLAN interface created by using the interface vlan <i>vlan-id</i> global configuration command. <p>These interfaces must have IP addresses assigned to them.</p>
Step 4	ip pim bsr-border Example: Device(config-if)# ip pim bsr-border	<p>Defines a PIM bootstrap message boundary for the PIM domain.</p> <p>Enter this command on each interface that connects to other bordering PIM domains. This command instructs the device to neither send nor receive PIMv2 BSR messages on this interface.</p> <p>Note To remove the PIM border, use the no ip pim bsr-border interface configuration command.</p>
Step 5	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 6	show running-config Example: Device# show running-config	Verifies your entries.
Step 7	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Defining the IP Multicast Boundary

You define a multicast boundary to prevent Auto-RP messages from entering the PIM domain. You create an access list to deny packets destined for 224.0.1.39 and 224.0.1.40, which carry Auto-RP information.

This procedure is optional.

Procedure

	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: # configure terminal	Enters global configuration mode.
Step 3	access-list access-list-number deny source <i>[source-wildcard]</i> Example: Device(config)# access-list 12 deny 224.0.1.39 access-list 12 deny 224.0.1.40	Creates a standard access list, repeating the command as many times as necessary. <ul style="list-style-type: none"> • For <i>access-list-number</i>, the range is 1 to 99. • The deny keyword denies access if the conditions are matched. • For <i>source</i>, enter multicast addresses 224.0.1.39 and 224.0.1.40, which carry Auto-RP information. • (Optional) For <i>source-wildcard</i>, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore. <p>The access list is always terminated by an implicit deny statement for everything.</p>
Step 4	interface interface-id Example: Device(config)# interface gigabitethernet 1/0/1	Specifies the interface to be configured, and enters interface configuration mode. The specified interface must be one of the following: <ul style="list-style-type: none"> • A routed port—A physical port that has been configured as a Layer 3 port by entering the no switchport interface configuration command.

	Command or Action	Purpose
		<ul style="list-style-type: none"> An SVI—A VLAN interface created by using the interface vlan <i>vlan-id</i> global configuration command. <p>These interfaces must have IP addresses assigned to them.</p>
Step 5	ip multicast boundary <i>access-list-number</i> Example: <pre>Device(config-if) # ip multicast boundary 12</pre>	Configures the boundary, specifying the access list you created in Step 2.
Step 6	end Example: <pre>Device(config) # end</pre>	Returns to privileged EXEC mode.
Step 7	show running-config Example: <pre>Device# show running-config</pre>	Verifies your entries.
Step 8	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Configuring Candidate BSRs

You can configure one or more candidate BSRs. The devices serving as candidate BSRs should have good connectivity to other devices and be in the backbone portion of the network.

This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	<p>Enables privileged EXEC mode.</p> <p>Enter your password if prompted.</p>

	Command or Action	Purpose
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ip pim bsr-candidate interface-id hash-mask-length [priority] Example: <pre>Device(config)# ip pim bsr-candidate gigabitethernet 1/0/3 28 100</pre>	Configures your device to be a candidate BSR. <ul style="list-style-type: none"> For <i>interface-id</i>, enter the interface on this device from which the BSR address is derived to make it a candidate. This interface must be enabled with PIM. Valid interfaces include physical ports, port channels, and VLANs. For <i>hash-mask-length</i>, specify the mask length (32 bits maximum) that is to be ANDed with the group address before the hash function is called. All groups with the same seed hash correspond to the same RP. For example, if this value is 24, only the first 24 bits of the group addresses matter. (Optional) For <i>priority</i>, enter a number from 0 to 255. The BSR with the larger priority is preferred. If the priority values are the same, the device with the highest IP address is selected as the BSR. The default is 0.
Step 4	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 5	show running-config Example: <pre>Device# show running-config</pre>	Verifies your entries.
Step 6	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Configuring the Candidate RPs

You can configure one or more candidate RPs. Similar to BSRs, the RPs should also have good connectivity to other devices and be in the backbone portion of the network. An RP can serve the entire IP multicast address space or a portion of it. Candidate RPs send candidate RP advertisements to the BSR.

This procedure is optional.

Before you begin

When deciding which devices should be RPs, consider these options:

- In a network of Cisco routers and multilayer switches where only Auto-RP is used, any device can be configured as an RP.
- In a network that includes only Cisco PIMv2 routers and multilayer switches and with routers from other vendors, any device can be used as an RP.
- In a network of Cisco PIMv1 routers, Cisco PIMv2 routers, and routers from other vendors, configure only Cisco PIMv2 routers and multilayer switches as RPs.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ip pim rp-candidate <i>interface-id</i> [group-list access-list-number] Example: <pre>Device(config)# ip pim rp-candidate gigabitethernet 1/0/5 group-list 10</pre>	Configures your device to be a candidate RP. <ul style="list-style-type: none"> • For <i>interface-id</i>, specify the interface whose associated IP address is advertised as a candidate RP address. Valid interfaces include physical ports, port channels, and VLANs. • (Optional) For group-list access-list-number, enter an IP standard access list number from 1 to 99. If no group-list is specified, the device is a candidate RP for all groups.
Step 4	access-list access-list-number {deny permit} source [source-wildcard]	Creates a standard access list, repeating the command as many times as necessary.

	Command or Action	Purpose
	Example: <pre>Device(config)# access-list 10 permit 239.0.0.0 0.255.255.255</pre>	<ul style="list-style-type: none"> For <i>access-list-number</i>, enter the access list number specified in Step 2. The deny keyword denies access if the conditions are matched. The permit keyword permits access if the conditions are matched. For <i>source</i>, enter the number of the network or host from which the packet is being sent. (Optional) For <i>source-wildcard</i>, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore. <p>The access list is always terminated by an implicit deny statement for everything.</p>
Step 5	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 6	show running-config Example: <pre>Device# show running-config</pre>	Verifies your entries.
Step 7	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Configuring Sparse Mode with Auto-RP

Before you begin

- All access lists that are needed when Auto-RP is configured should be configured prior to beginning the configuration task.

**Note**

- If a group has no known RP and the interface is configured to be sparse-dense mode, the interface is treated as if it were in dense mode, and data is flooded over the interface. To avoid this data flooding, configure the Auto-RP listener and then configure the interface as sparse mode.
- When configuring Auto-RP, you must either configure the Auto-RP listener feature (Step 5) and specify sparse mode (Step 7).
- When you configure sparse-dense mode, dense mode failover may result in a network dense-mode flood. To avoid this condition, use PIM sparse mode with the Auto-RP listener feature.

Follow this procedure to configure auto-rendezvous point (Auto-RP). Auto-RP can also be optionally used with anycast RP.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ip multicast-routing Example: <pre>Device(config)# ip multicast-routing</pre>	Enables IP multicast routing.
Step 4	Either perform Steps 5 through 7 or perform Steps 6 and 8.	--
Step 5	interface <i>type number</i> Example: <pre>Device(config)# interface GigabitEthernet 1/0/0</pre>	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 6	ip pim sparse-mode Example: <pre>Device(config-if)# ip pim sparse-mode</pre>	Enables PIM sparse mode on an interface. When configuring Auto-RP in sparse mode, you must also configure the Auto-RP listener in the next step. <ul style="list-style-type: none"> • Skip this step if you are configuring sparse-dense mode in Step 8.

	Command or Action	Purpose
Step 7	exit Example: <pre>Device(config-if)# exit</pre>	Exits interface configuration mode and returns to global configuration mode.
Step 8	Repeat Steps 1 through 9 on all PIM interfaces.	--
Step 9	ip pim send-rp-announce <i>{interface-type interface-number ip-address}</i> scope <i>ttl-value</i> [group-list <i>access-list</i>] [interval <i>seconds</i>] Example: <pre>Device(config)# ip pim send-rp-announce loopback0 scope 31 group-list 5</pre>	<p>Sends RP announcements out all PIM-enabled interfaces.</p> <ul style="list-style-type: none"> • Perform this step on the RP device only. • Use the <i>interface-type</i> and <i>interface-number</i> arguments to define which IP address is to be used as the RP address. • Use the <i>ip-address</i> argument to specify a directly connected IP address as the RP address. <p>Note If the <i>ip-address</i> argument is configured for this command, the RP-announce message will be sourced by the interface to which this IP address is connected (that is, the source address in the IP header of the RP-announce message is the IP address of that interface).</p> <ul style="list-style-type: none"> • This example shows that the interface is enabled with a maximum of 31 hops. The IP address by which the device wants to be identified as RP is the IP address associated with loopback interface 0. Access list 5 describes the groups for which this device serves as RP.
Step 10	ip pim send-rp-discovery [<i>interface-type interface-number</i>] scope <i>ttl-value</i> [interval <i>seconds</i>] Example: <pre>Device(config)# ip pim send-rp-discovery loopback 1 scope 31</pre>	<p>Configures the device to be an RP mapping agent.</p> <ul style="list-style-type: none"> • Perform this step on RP mapping agent devices or on combined RP/RP mapping agent devices.

	Command or Action	Purpose
		<p>Note Auto-RP allows the RP function to run separately on one device and the RP mapping agent to run on one or multiple devices. It is possible to deploy the RP and the RP mapping agent on a combined RP/RP mapping agent device.</p> <ul style="list-style-type: none"> • Use the optional <i>interface-type</i> and <i>interface-number</i> arguments to define which IP address is to be used as the source address of the RP mapping agent. • Use the scope keyword and <i>tvl-value</i> argument to specify the Time-to-Live (TTL) value in the IP header of Auto-RP discovery messages. • Use the optional interval keyword and <i>seconds</i> argument to specify the interval at which Auto-RP discovery messages are sent. <p>Note Lowering the interval at which Auto-RP discovery messages are sent from the default value of 60 seconds results in more frequent floodings of the group-to-RP mappings. In some network environments, the disadvantages of lowering the interval (more control packet overhead) may outweigh the advantages (more frequent group-to-RP mapping updates).</p> <ul style="list-style-type: none"> • The example shows limiting the Auto-RP discovery messages to 31 hops on loopback interface 1.
Step 11	<p>ip pim rp-announce-filter rp-list <i>access-list</i> group-list <i>access-list</i></p> <p>Example:</p> <pre>Device(config)# ip pim rp-announce-filter rp-list 1 group-list 2</pre>	<p>Filters incoming RP announcement messages sent from candidate RPs (C-RPs) to the RP mapping agent.</p> <ul style="list-style-type: none"> • Perform this step on the RP mapping agent only.
Step 12	<p>interface <i>type number</i></p> <p>Example:</p>	<p>Selects an interface that is connected to hosts on which PIM can be enabled.</p>

	Command or Action	Purpose
	Device (config)# interface gigabitethernet 1/0/0	
Step 13	ip multicast boundary <i>access-list</i> [filter-autorp] Example: Device (config-if)# ip multicast boundary 10 filter-autorp	Configures an administratively scoped boundary. <ul style="list-style-type: none"> • Perform this step on the interfaces that are boundaries to other devices. • The access list is not shown in this task. • An access list entry that uses the deny keyword creates a multicast boundary for packets that match that entry.
Step 14	end Example: Device (config-if)# end	Returns to global configuration mode.
Step 15	show ip pim autorp Example: Device# show ip pim autorp	(Optional) Displays the Auto-RP information.
Step 16	show ip pim rp [mapping] [rp-address] Example: Device# show ip pim rp mapping	(Optional) Displays RPs known in the network and shows how the device learned about each RP.
Step 17	show ip igmp groups [<i>group-name</i> <i>group-address</i> <i>interface-type</i> <i>interface-number</i>] [detail] Example: Device# show ip igmp groups	(Optional) Displays the multicast groups having receivers that are directly connected to the device and that were learned through Internet Group Management Protocol (IGMP). <ul style="list-style-type: none"> • A receiver must be active on the network at the time that this command is issued in order for receiver information to be present on the resulting display.
Step 18	show ip mroute [<i>group-address</i> <i>group-name</i>] [<i>source-address</i> <i>source-name</i>] [<i>interface-type</i> <i>interface-number</i>] [summary] [count] [active <i>kpps</i>] Example: Device# show ip mroute cbone-audio	(Optional) Displays the contents of the IP multicast routing (mroute) table.

Configuring IPv4 Bidirectional PIM

This section provides information about configuring Bidirectional PIM.

Enabling Bidirectional PIM Globally

To enable IPv4 bidirectional PIM, perform this task:

Before you begin

Before configuring bidirectional PIM, ensure that the feature is supported on all IP multicast-enabled routers in that domain. It is not possible to enable groups for bidirectional PIM operation in a partially upgraded network. Packet loops will occur immediately in networks that are only partially upgraded to support bidirectional PIM.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip pim bidir-enable Example: Device(config)# ip pim bidir-enable	Enables IPv4 bidirectional PIM globally on the device.

Configuring the Rendezvous Point for IPv4 Bidirectional PIM Groups

To statically configure the rendezvous point for an IPv4 bidirectional PIM group, perform this task:

Before you begin

Before configuring rendezvous points for IPv4 Bidirectional PIM groups, ensure that you have enabled bidirectional PIM globally.

Procedure

	Command or Action	Purpose
Step 1	ip pim [vrf vrf-name] rp-address ip-address [access-list] [override] bidir Example: Device(config)# ip pim rp-address 10.0.0.1 10 override bidir	Statically configures the IP address of the rendezvous point for the group. When you specify the override option, the static rendezvous point is used.

	Command or Action	Purpose
Step 2	access-list <i>access-list</i> [permit deny] <i>ip-address</i> Example: Device(config)# access-list 10 permit 224.1.0.0 0.0.255.255	Configures an access list.
Step 3	ip pim [<i>vrf vrf-name</i>] send-rp-announce <i>interface-type interface-number</i> scope <i>ttl-value</i> [group-list <i>access-list</i>] [interval <i>seconds</i>] [bidir] Example: Device(config)# ip pim send-rp-announce Loopback0 scope 16 group-list c21-rp-list-0 bidir	Configures the system to use auto-RP to configure groups for which the router will act as a rendezvous point (RP).
Step 4	ip access-list standard <i>access-list-name</i> [permit deny] <i>ip-address</i> Example: Device(config)# ip access-list standard c21-rp-list-0 permit 230.31.31.1 0.0.255.255	Configures a standard IP access list.

Delaying the Use of PIM Shortest-Path Tree

Perform these steps to configure a traffic rate threshold that must be reached before multicast routing is switched from the source tree to the shortest-path tree.

This procedure is optional.

Procedure

	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	access-list <i>access-list-number</i> { deny permit } <i>source</i> [<i>source-wildcard</i>] Example:	Creates a standard access list. <ul style="list-style-type: none"> • For <i>access-list-number</i>, the range is 1 to 99.

	Command or Action	Purpose
	<pre>Device(config)# access-list 16 permit 225.0.0.0 0.255.255.255</pre>	<ul style="list-style-type: none"> • The deny keyword denies access if the conditions are matched. • The permit keyword permits access if the conditions are matched. • For <i>source</i>, specify the multicast group to which the threshold will apply. • (Optional) For <i>source-wildcard</i>, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore. <p>The access list is always terminated by an implicit deny statement for everything.</p>
Step 4	<p>ip pim spt-threshold {<i>kbits</i> infinity} [group-list <i>access-list-number</i>]</p> <p>Example:</p> <pre>Device(config)# ip pim spt-threshold infinity group-list 16</pre>	<p>Specifies the threshold that must be reached before moving to shortest-path tree (spt).</p> <ul style="list-style-type: none"> • For <i>kbits</i>, specify the traffic rate in kilobits per second. The default is 0 kbps. <p>Note Because of device hardware limitations, 0 kbps is the only valid entry even though the range is 0 to 4294967.</p> <ul style="list-style-type: none"> • Specify infinity if you want all sources for the specified group to use the shared tree, never switching to the source tree. • (Optional) For group-list <i>access-list-number</i>, specify the access list created in Step 2. If the value is 0 or if the group list is not used, the threshold applies to all groups.
Step 5	<p>end</p> <p>Example:</p> <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 6	<p>show running-config</p> <p>Example:</p> <pre>Device# show running-config</pre>	Verifies your entries.

	Command or Action	Purpose
Step 7	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Modifying the PIM Router-Query Message Interval

PIM routers and multilayer switches send PIM router-query messages to find which device will be the designated router (DR) for each LAN segment (subnet). The DR is responsible for sending IGMP host-query messages to all hosts on the directly connected LAN.

With PIM DM operation, the DR has meaning only if IGMPv1 is in use. IGMPv1 does not have an IGMP querier election process, so the elected DR functions as the IGMP querier. With PIM-SM operation, the DR is the device that is directly connected to the multicast source. It sends PIM register messages to notify the RP that multicast traffic from a source needs to be forwarded down the shared tree. In this case, the DR is the device with the highest IP address.

This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	interface interface-id Example: <pre>Device(config)# interface gigabitethernet 1/0/1</pre>	Specifies the interface to be configured, and enters interface configuration mode. The specified interface must be one of the following: <ul style="list-style-type: none"> • A routed port—A physical port that has been configured as a Layer 3 port by entering the no switchport interface configuration command. • An SVI—A VLAN interface created by using the interface vlan vlan-id global configuration command.

	Command or Action	Purpose
		These interfaces must have IP addresses assigned to them.
Step 4	ip pim query-interval <i>seconds</i> Example: <pre>Device(config-if) # ip pim query-interval 45</pre>	Configures the frequency at which the device sends PIM router-query messages. The default is 30 seconds. The range is 1 to 65535.
Step 5	end Example: <pre>Device(config) # end</pre>	Returns to privileged EXEC mode.
Step 6	show ip igmp interface [<i>interface-id</i>] Example: <pre>Device# show ip igmp interface</pre>	Verifies your entries.
Step 7	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Verifying PIM Operations

Verifying IP Multicast Operation in a PIM-SM or a PIM-SSM Network

When you verify the operation of IP multicast in a PIM-SM network environment or in an PIM-SSM network environment, a useful approach is to begin the verification process on the last hop router, and then continue the verification process on the routers along the SPT until the first hop router has been reached. The goal of the verification is to ensure that IP multicast traffic is being routed properly through an IP multicast network.

Perform the following optional tasks to verify IP multicast operation in a PIM-SM or a PIM-SSM network. The steps in these tasks help to locate a faulty hop when sources and receivers are not operating as expected.



Note

If packets are not reaching their expected destinations, you might want consider disabling IP multicast fast switching, which would place the router in process switching mode. If packets begin reaching their proper destinations after IP multicast fast switching has been disabled, then the issue most likely was related to IP multicast fast switching.

Verifying IP Multicast on the First Hop Router

Enter these commands on the first hop router to verify IP multicast operations on the first hop router:

Procedure

	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	show ip mroute [<i>group-address</i>] Example: Device# show ip mroute 239.1.2.3 (*, 239.1.2.3), 00:18:10/stopped, RP 172.16.0.1, flags: SPF Incoming interface: Serial1/0, RPF nbr 172.31.200.2 Outgoing interface list: Null (10.0.0.1, 239.1.2.3), 00:18:10/00:03:22, flags: FT Incoming interface: GigabitEthernet0/0/0, RPF nbr 0.0.0.0 Outgoing interface list: Serial1/0, Forward/Sparse, 00:18:10/00:03:19	Confirms that the F flag has been set for mroutes on the first hop router.
Step 3	show ip mroute active [<i>kb/s</i>] Example: Device# show ip mroute active Active IP Multicast Sources - sending >= 4 kbps	Displays information about active multicast sources sending to groups. The output of this command provides information about the multicast packet rate for active sources.

	Command or Action	Purpose
	<pre> Group: 239.1.2.3, (?) Source: 10.0.0.1 (?) Rate: 20 pps/4 kbps(1sec), 4 kbps(last 30 secs), 4 kbps(life avg) </pre>	<p>Note By default, the output of the show ip mroute command with the active keyword displays information about active sources sending traffic to groups at a rate greater than or equal to 4 kb/s. To display information about active sources sending low-rate traffic to groups (that is, traffic less than 4 kb/s), specify a value of 1 for the <i>kb/s</i> argument. Specifying a value of 1 for this argument displays information about active sources sending traffic to groups at a rate equal to or greater than 1 kb/s, which effectively displays information about all possible active source traffic.</p>

Verifying IP Multicast on Routers Along the SPT

Enter these commands on routers along the SPT to verify IP multicast operations on routers along the SPT in a PIM-SM or PIM-SSM network:

Procedure

	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>show ip mroute <i>[group-address]</i></p> <p>Example:</p> <pre> Device# show ip mroute 239.1.2.3 (*, 239.1.2.3), 00:17:56/00:03:02, RP 172.16.0.1, flags: S Incoming interface: Null, RPF nbr 0.0.0.0 Outgoing interface list: GigabitEthernet0/0/0, Forward/Sparse, 00:17:56/00:03:02 (10.0.0.1, 239.1.2.3), 00:15:34/00:03:28, flags: T Incoming interface: Serial1/0, RPF nbr 172.31.200.1 Outgoing interface list: </pre>	<p>Confirms the RPF neighbor towards the source for a particular group or groups.</p>

	Command or Action	Purpose
	GigabitEthernet0/0/0, Forward/Sparse, 00:15:34/00:03:02	
Step 3	show ip mroute active Example: <pre>Device# show ip mroute active Active IP Multicast Sources - sending >= 4 kbps Group: 239.1.2.3, (?) Source: 10.0.0.1 (?) Rate: 20 pps/4 kbps(1sec), 4 kbps(last 30 secs), 4 kbps(life avg)</pre>	<p>Displays information about active multicast sources sending to groups. The output of this command provides information about the multicast packet rate for active sources.</p> <p>Note By default, the output of the show ip mroute command with the active keyword displays information about active sources sending traffic to groups at a rate greater than or equal to 4 kb/s. To display information about active sources sending low-rate traffic to groups (that is, traffic less than 4 kb/s), specify a value of 1 for the <i>kb/s</i> argument. Specifying a value of 1 for this argument displays information about active sources sending traffic to groups at a rate equal to or greater than 1 kb/s, which effectively displays information about all possible active source traffic.</p>

Verifying IP Multicast Operation on the Last Hop Router

Enter these commands on the last hop router to verify IP multicast operations on the last hop router:

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	show ip igmp groups Example: <pre>Device# show ip igmp groups IGMP Connected Group Membership Group Address Interface Uptime Expires Last Reporter 239.1.2.3 GigabitEthernet1/0/0 00:05:14 00:02:14 10.1.0.6</pre>	<p>Verifies IGMP memberships on the last hop router. This information will confirm the multicast groups with receivers that are directly connected to the last hop router and that are learned through IGMP.</p>

	Command or Action	Purpose
	<pre>224.0.1.39 GigabitEthernet0/0/0 00:09:11 00:02:08 172.31.100.1</pre>	
Step 3	<p>show ip pim rp mapping</p> <p>Example:</p> <pre>Device# show ip pim rp mapping PIM Group-to-RP Mappings Group(s) 224.0.0.0/4 RP 172.16.0.1 (?), v2v1 Info source: 172.16.0.1 (?), elected via Auto-RP Uptime: 00:09:11, expires: 00:02:47</pre>	<p>Confirms that the group-to-RP mappings are being populated correctly on the last hop router.</p> <p>Note Ignore this step if you are verifying a last hop router in a PIM-SSM network. The show ip pim rp mapping command does not work with routers in a PIM-SSM network because PIM-SSM does not use RPs. In addition, if configured correctly, PIM-SSM groups do not appear in the output of the show ip pim rp mapping command.</p>
Step 4	<p>show ip mroute</p> <p>Example:</p> <pre>Device# show ip mroute (*, 239.1.2.3), 00:05:14/00:03:04, RP 172.16.0.1, flags: SJC Incoming interface: GigabitEthernet0/0/0, RPF nbr 172.31.100.1 Outgoing interface list: GigabitEthernet1/0, Forward/Sparse, 00:05:10/00:03:04 (10.0.0.1, 239.1.2.3), 00:02:49/00:03:29, flags: T Incoming interface: GigabitEthernet0/0/0, RPF nbr 172.31.100.1 Outgoing interface list: GigabitEthernet1/0, Forward/Sparse, 00:02:49/00:03:04 (*, 224.0.1.39), 00:10:05/stopped, RP 0.0.0.0, flags: DC Incoming interface: Null, RPF nbr 0.0.0.0 Outgoing interface list: GigabitEthernet1/0, Forward/Sparse, 00:05:15/00:00:00 GigabitEthernet0/0, Forward/Sparse, 00:10:05/00:00:00 (172.16.0.1, 224.0.1.39), 00:02:00/00:01:33, flags: PTX Incoming interface: GigabitEthernet0/0/0, RPF nbr 172.31.100.1</pre>	<p>Verifies that the mroute table is being populated properly on the last hop router.</p>

	Command or Action	Purpose
Step 5	<p>show ip interface <i>[type number]</i></p> <p>Example:</p> <pre>Device# show ip interface GigabitEthernet 0/0/0 GigabitEthernet0/0/0 is up, line protocol is up Internet address is 172.31.100.2/24 Broadcast address is 255.255.255.255 Address determined by setup command MTU is 1500 bytes Helper address is not set Directed broadcast forwarding is disabled Multicast reserved groups joined: 224.0.0.1 224.0.0.22 224.0.0.13 224.0.0.5 224.0.0.6 Outgoing access list is not set Inbound access list is not set Proxy ARP is enabled Local Proxy ARP is disabled Security level is default Split horizon is enabled ICMP redirects are always sent ICMP unreachable are always sent ICMP mask replies are never sent IP fast switching is enabled IP fast switching on the same interface is disabled IP Flow switching is disabled IP CEF switching is disabled IP Fast switching turbo vector IP multicast fast switching is enabled IP route-cache flags are Fast Router Discovery is disabled IP output packet accounting is disabled IP access violation accounting is disabled TCP/IP header compression is disabled RTP/IP header compression is disabled Policy routing is disabled Network address translation is disabled WCCP Redirect outbound is disabled WCCP Redirect inbound is disabled WCCP Redirect exclude is disabled BGP Policy Mapping is disabled</pre>	<p>Verifies that multicast fast switching is enabled for optimal performance on the outgoing interface on the last hop router.</p> <p>Note Using the no ip mroute-cache interface command disables IP multicast fast-switching. When IP multicast fast switching is disabled, packets are forwarded through the process-switched path.</p>
Step 6	<p>show ip mfib</p> <p>Example:</p> <pre>Device# show ip mfib</pre>	Displays the forwarding entries and interfaces in the IP Multicast Forwarding Information Base (MFIB).
Step 7	<p>show ip pim interface count</p> <p>Example:</p> <pre>Device# show ip pim interface count</pre>	Confirms that multicast traffic is being forwarded on the last hop router.

	Command or Action	Purpose
	<pre> State: * - Fast Switched, H - Hardware Switching Enabled Address Interface FS Mpackets In/Out 172.31.100.2 GigabitEthernet0/0/0 * 4122/0 10.1.0.1 GigabitEthernet1/0/0 * 0/3193 </pre>	
Step 8	<p>show ip mroute count</p> <p>Example:</p> <pre> Device# show ip mroute count IP Multicast Statistics 6 routes using 4008 bytes of memory 3 groups, 1.00 average sources per group Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per second Other counts: Total/RPF failed/Other drops(OIF-null, rate-limit etc) Group: 239.1.2.3, Source count: 1, Packets forwarded: 3165, Packets received: 3165 RP-tree: Forwarding: 0/0/0/0, Other: 0/0/0 Source: 10.0.0.1/32, Forwarding: 3165/20/28/4, Other: 0/0/0 Group: 224.0.1.39, Source count: 1, Packets forwarded: 21, Packets received: 120 Source: 172.16.0.1/32, Forwarding: 21/1/48/0, Other: 120/0/99 Group: 224.0.1.40, Source count: 1, Packets forwarded: 10, Packets received: 10 Source: 172.16.0.1/32, Forwarding: 10/1/48/0, Other: 10/0/0 </pre>	Confirms that multicast traffic is being forwarded on the last hop router.
Step 9	<p>show ip mroute active [kb/s]</p> <p>Example:</p> <pre> Device# show ip mroute active Active IP Multicast Sources - sending >= 4 kbps </pre>	Displays information about active multicast sources sending traffic to groups on the last hop router. The output of this command provides information about the multicast packet rate for active sources.

	Command or Action	Purpose
	Group: 239.1.2.3, (?) Source: 10.0.0.1 (?) Rate: 20 pps/4 kbps(1sec), 4 kbps(last 50 secs), 4 kbps(life avg)	Note By default, the output of the show ip mroute command with the active keyword displays information about active sources sending traffic to groups at a rate greater than or equal to 4 kb/s. To display information about active sources sending low-rate traffic to groups (that is, traffic less than 4 kb/s), specify a value of 1 for the <i>kb/s</i> argument. Specifying a value of 1 for this argument displays information about active sources sending traffic to groups at a rate equal to or greater than 1 kb/s, which effectively displays information about all possible active source traffic.

Using PIM-Enabled Routers to Test IP Multicast Reachability

If all the PIM-enabled routers and access servers that you administer are members of a multicast group, pinging that group causes all routers to respond, which can be a useful administrative and debugging tool.

To use PIM-enabled routers to test IP multicast reachability, perform the following tasks:

Configuring Routers to Respond to Multicast Pings

Follow these steps to configure a router to respond to multicast pings. Perform the task on all the interfaces of a router and on all the routers participating in the multicast network:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface type number Example:	Enters interface configuration mode.

	Command or Action	Purpose
	Device(config)# interface <i>gigabitethernet</i> 1/0/0	For the <i>type</i> and <i>number</i> arguments, specify an interface that is directly connected to hosts or is facing hosts.
Step 4	ip igmp join-group <i>group-address</i> Example: Device(config-if)# ip igmp join-group 225.2.2.2	(Optional) Configures an interface on the router to join the specified group. For the purpose of this task, configure the same group address for the <i>group-address</i> argument on all interfaces on the router participating in the multicast network. Note With this method, the router accepts the multicast packets in addition to forwarding them. Accepting the multicast packets prevents the router from fast switching.
Step 5	Repeat Step 3 and Step 4 for each interface on the router participating in the multicast network.	--
Step 6	end Example: Device(config-if)# end	Ends the current configuration session and returns to privileged EXEC mode.

Pinging Routers Configured to Respond to Multicast Pings

Follow these steps on a router to initiate a ping test to the routers configured to respond to multicast pings. This task is used to test IP multicast reachability in a network.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	ping <i>group-address</i> Example: Device# ping 225.2.2.2	Pings an IP multicast group address. A successful response indicates that the group address is functioning.

Monitoring and Troubleshooting PIM

Monitoring PIM Information

Use the privileged EXEC commands in the following table to monitor your PIM configurations.

Table 15: PIM Monitoring Commands

Command	Purpose
show ip pim all-vrfs tunnel [tunnel <i>tunnel_number</i> verbose]	Displays all VRFs.
show ip pim autorp	Displays global auto-RP information.
show ip pim boundary	Displays information about mroutes filtered by administratively scoped IPv4 multicast boundaries configured on an interface.
show ip pim interface	Displays information about interfaces configured for Protocol Independent Multicast (PIM).
show ip pim neighbor	Displays the PIM neighbor information.
show ip pim rp [<i>group-name</i> <i>group-address</i>]	Displays RP routers associated with a sparse-mode multicast group. This command is available in all software images.
show ip pim tunnel [tunnel verbose]	Displays information about Protocol Independent Multicast (PIM) tunnel interfaces
show ip pim vrf { word { all-vrfs autorp boundary bsr-router interface mdt neighbor rp rp-hash tunnel } }	Displays the VPN routing/forwarding instance.
show ip igmp groups detail	Displays the interested clients that have joined the specific multicast source group.

Monitoring the RP Mapping and BSR Information

Use the privileged EXEC mode in the following table to verify the consistency of group-to-RP mappings:

Table 16: RP Mapping Monitoring Commands

Command	Purpose
show ip pim rp [<i>hostname</i> or <i>IP address</i> mapping [<i>hostname</i> or <i>IP address</i> elected in-use] metric [<i>hostname</i> or <i>IP address</i>]]	<p>Displays all available RP mappings and metrics. This tells you how the device learns of the RP (through the BSR or the Auto-RP mechanism).</p> <ul style="list-style-type: none"> • (Optional) For the <i>hostname</i>, specify the IP name of the group about which to display RPs. • (Optional) For the <i>IP address</i>, specify the IP address of the group about which to display RPs. • (Optional) Use the mapping keyword to display all group-to-RP mappings of which the Cisco device is aware (either configured or learned from Auto-RP). • (Optional) Use the metric keyword to display the RP RPF metric.
show ip pim rp-hash <i>group</i>	<p>Displays the RP that was selected for the specified group. That is, on a PIMv2 router or multilayer switch, confirms that the same RP is the one that a PIMv1 system chooses. For <i>group</i>, enter the group address for which to display RP information.</p>

Use the privileged EXEC commands in the following table to monitor BSR information:

Table 17: BSR Monitoring Commands

Command	Purpose
show ip pim bsr	Displays information about the elected BSR.
show ip pim bsr-router	Displays information about the BSRv2.

Troubleshooting PIMv1 and PIMv2 Interoperability Problems

When debugging interoperability problems between PIMv1 and PIMv2, check these in the order shown:

1. Verify RP mapping with the **show ip pim rp-hash** privileged EXEC command, making sure that all systems agree on the same RP for the same group.
2. Verify interoperability between different versions of DRs and RPs. Make sure that the RPs are interacting with the DRs properly (by responding with register-stops and forwarding decapsulated data packets from registers).

Monitoring IPv4 Bidirectional PIM Information

Use the privileged EXEC commands in the following table to monitor your Bidirectional PIM configurations.

Command	Purpose
show ip mfib	Displays MFIB information for bidirectional PIM.
show platform software fed {active standby } ip multicast groups	Displays platform-dependent IP multicast tables and other information.
show ip pim [vrf vrf-name] interface interface-type interface-number df [rp-address]	Displays information about interfaces configured for PIM.
show ip pim [vrf vrf-name] rp [mapping metric] [rp-address]	Displays active rendezvous points (RPs) that are cached with associated multicast routing entries.
show platform software fed ip multicast df [vrf-id vrf-id vrf-name vrf-name] [df-index]	Displays information about IP multicast designated forwarders (DF).

Configuration Examples for PIM

Example: Enabling PIM Stub Routing

In this example, IP multicast routing is enabled, Switch A PIM uplink port 25 is configured as a routed uplink port with **sparse-dense-mode** enabled. PIM stub routing is enabled on the VLAN 100 interfaces and on Gigabit Ethernet port 20.

```

Device(config)# ip multicast-routing
Device(config)# interface GigabitEthernet3/0/25
Device(config-if)# no switchport
Device(config-if)# ip address 3.1.1.2 255.255.255.0
Device(config-if)# ip pim sparse-dense-mode
Device(config-if)# exit
Device(config)# interface vlan100
Device(config-if)# ip pim passive
Device(config-if)# exit
Device(config)# interface GigabitEthernet3/0/20
Device(config-if)# ip pim passive
Device(config-if)# exit
Device(config)# interface vlan100
Device(config-if)# ip address 100.1.1.1 255.255.255.0
Device(config-if)# ip pim passive
Device(config-if)# exit
Device(config)# interface GigabitEthernet3/0/20
Device(config-if)# no switchport
Device(config-if)# ip address 10.1.1.1 255.255.255.0
Device(config-if)# ip pim passive
Device(config-if)# end

```

Example: Verifying PIM Stub Routing

To verify that PIM stub is enabled for each interface, use the **show ip pim interface** command in privileged EXEC mode:

```
Device# show ip pim interface
Address Interface Ver/ Nbr Query DR DR
Mode Count Intvl Prior
3.1.1.2 GigabitEthernet3/0/25 v2/SD 1 30 1 3.1.1.2
100.1.1.1 Vlan100 v2/P 0 30 1 100.1.1.1
10.1.1.1 GigabitEthernet3/0/20 v2/P 0 30 1 10.1.1.1
```

Example: Manually Assigning an RP to Multicast Groups

This example shows how to configure the address of the RP to 147.106.6.22 for multicast group 225.2.2.2 only:

```
Device(config)# access-list 1 permit 225.2.2.2 0.0.0.0
Device(config)# ip pim rp-address 147.106.6.22 1
```

Example: Configuring Auto-RP

This example shows how to send RP announcements out all PIM-enabled interfaces for a maximum of 31 hops. The IP address of port 1 is the RP. Access list 5 describes the group for which this device serves as RP:

```
Device(config)# ip pim send-rp-announce gigabitethernet1/0/1 scope 31 group-list 5
Device(config)# access-list 5 permit 224.0.0.0 15.255.255.255
```

Example: Sparse Mode with Auto-RP

The following example configures sparse mode with Auto-RP:

```
Device(config)# ip multicast-routing
Device(config)# ip pim autorp listener
Device(config)# ip pim send-rp-announce Loopback0 scope 16 group-list 1
Device(config)# ip pim send-rp-discovery Loopback1 scope 16
Device(config)# no ip pim dm-fallback
Device(config)# access-list 1 permit 239.254.2.0 0.0.0.255
Device(config)# access-list 1 permit 239.254.3.0 0.0.0.255
.
.
.
Device(config)# access-list 10 permit 224.0.1.39
Device(config)# access-list 10 permit 224.0.1.40
Device(config)# access-list 10 permit 239.254.2.0 0.0.0.255
Device(config)# access-list 10 permit 239.254.3.0 0.0.0.255
```

Example: Defining the IP Multicast Boundary to Deny Auto-RP Information

This example shows a portion of an IP multicast boundary configuration that denies Auto-RP information:

```
Device(config)# access-list 1 deny 224.0.1.39
Device(config)# access-list 1 deny 224.0.1.40
Device(config)# access-list 1 permit all
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ip multicast boundary 1
```

Example: Filtering Incoming RP Announcement Messages

This example shows a sample configuration on an Auto-RP mapping agent that is used to prevent candidate RP announcements from being accepted from unauthorized candidate RPs:

```
Device(config)# ip pim rp-announce-filter rp-list 10 group-list 20
Device(config)# access-list 10 permit host 172.16.5.1
Device(config)# access-list 10 permit host 172.16.2.1
Device(config)# access-list 20 deny 239.0.0.0 0.0.255.255
Device(config)# access-list 20 permit 224.0.0.0 15.255.255.255
```

The mapping agent accepts candidate RP announcements from only two devices, 172.16.5.1 and 172.16.2.1. The mapping agent accepts candidate RP announcements from these two devices only for multicast groups that fall in the group range of 224.0.0.0 to 239.255.255.255. The mapping agent does not accept candidate RP announcements from any other devices in the network. Furthermore, the mapping agent does not accept candidate RP announcements from 172.16.5.1 or 172.16.2.1 if the announcements are for any groups in the 239.0.0.0 through 239.255.255.255 range. This range is the administratively scoped address range.

Example: Preventing Join Messages to False RPs

If all interfaces are in sparse mode, use a default-configured RP to support the two well-known groups 224.0.1.39 and 224.0.1.40. Auto-RP uses these two well-known groups to collect and distribute RP-mapping information. When this is the case and the **ip pim accept-rp auto-rp** command is configured, another **ip pim accept-rp** command accepting the RP must be configured as follows:

```
Device(config)# ip pim accept-rp 172.10.20.1 1
Device(config)# access-list 1 permit 224.0.1.39
Device(config)# access-list 1 permit 224.0.1.40
```

Example: Configuring Candidate BSRs

This example shows how to configure a candidate BSR, which uses the IP address 172.21.24.18 on a port as the advertised BSR address, uses 30 bits as the hash-mask-length, and has a priority of 10.

```
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# ip address 172.21.24.18 255.255.255.0
Device(config-if)# ip pim sparse-mode
Device(config-if)# ip pim bsr-candidate gigabitethernet1/0/2 30 10
```

Example: Configuring Candidate RPs

This example shows how to configure the device to advertise itself as a candidate RP to the BSR in its PIM domain. Standard access list number 4 specifies the group prefix associated with the RP that has the address identified by a port. That RP is responsible for the groups with the prefix 239.

```
Device(config)# ip pim rp-candidate gigabitethernet1/0/2 group-list 4
Device(config)# access-list 4 permit 239.0.0.0 0.255.255.255
```

Feature History for PIM

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	PIM	The Protocol Independent Multicast (PIM) protocol maintains the current IP multicast service mode of receiver-initiated membership. PIM is not dependent on a specific unicast routing protocol; it is IP routing protocol independent and can leverage whichever unicast routing protocols are used to populate the unicast routing table, including Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First (OSPF), Border Gateway Protocol (BGP), and static routes. PIM uses unicast routing information to perform the multicast forwarding function.
Cisco IOS XE Gibraltar 16.12.1	Bidirectional PIM	Bidirectional PIM is a variant of the PIM suite of routing protocols for IP multicast.

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>.



CHAPTER 9

Configuring PIM MIB Extension for IP Multicast

- [Information About PIM MIB Extension for IP Multicast, on page 185](#)
- [How to Configure PIM MIB Extension for IP Multicast, on page 186](#)
- [Configuration Examples for PIM MIB Extensions, on page 187](#)
- [Additional References for PIM MIB Extension for IP Multicast, on page 188](#)
- [Feature History for PIM MIB Extension for IP Multicast, on page 188](#)

Information About PIM MIB Extension for IP Multicast

PIM MIB Extensions for SNMP Traps for IP Multicast

Protocol Independent Multicast (PIM) is an IP multicast routing protocol used for routing multicast data packets to multicast groups. RFC 2934 defines the PIM MIB for IPv4, which describes managed objects that enable users to remotely monitor and configure PIM using Simple Network Management Protocol (SNMP).

PIM MIB extensions introduce the following new classes of PIM notifications:

- neighbor-change--This notification results from the following conditions:
 - A router's PIM interface is disabled or enabled (using the **ip pim** command in interface configuration mode)
 - A router's PIM neighbor adjacency expires (defined in RFC 2934)
- rp-mapping-change--This notification results from a change in the rendezvous point (RP) mapping information due to either Auto-RP messages or bootstrap router (BSR) messages.
- invalid-pim-message--This notification results from the following conditions:
 - An invalid (*, G) Join or Prune message is received by the device (for example, when a router receives a Join or Prune message for which the RP specified in the packet is not the RP for the multicast group)
 - An invalid PIM register message is received by the device (for example, when a router receives a register message from a multicast group for which it is not the RP)

Benefits of PIM MIB Extensions

PIM MIB extensions:

- Allow users to identify changes in the multicast topology of their network by detecting changes in the RP mapping.
- Provide traps to monitor the PIM protocol on PIM-enabled interfaces.
- Help users identify routing issues when multicast neighbor adjacencies expire on a multicast interface.
- Enable users to monitor RP configuration errors (for example, errors due to flapping in dynamic RP allocation protocols like Auto-RP).

How to Configure PIM MIB Extension for IP Multicast

Enabling PIM MIB Extensions for IP Multicast

Perform this task to enable PIM MIB extensions for IP multicast.



Note

- The pimInterfaceVersion object was removed from RFC 2934 and, therefore, is no longer supported in software.
- The following MIB tables are not supported in Cisco software:
 - pimIpMRouteTable
 - pimIpMRouteNextHopTable

Procedure

	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	snmp-server enable traps pim [neighbor-change rp-mapping-change invalid-pim-message] Example:	Enables a device to send PIM notifications. <ul style="list-style-type: none"> • neighbor-change --This keyword enables notifications indicating when a device's PIM interface is disabled or enabled, or

	Command or Action	Purpose
	<pre>Device(config)# snmp-server enable traps pim neighbor-change</pre>	<p>when a device's PIM neighbor adjacency expires.</p> <ul style="list-style-type: none"> • rp-mapping-change --This keyword enables notifications indicating a change in RP mapping information due to either Auto-RP messages or BSR messages. • invalid-pim-message --This keyword enables notifications for monitoring invalid PIM protocol operations (for example, when a device receives a join or prune message for which the RP specified in the packet is not the RP for the multicast group or when a device receives a register message from a multicast group for which it is not the RP).
Step 4	<p>snmp-server host <i>host-address</i> [traps informs] <i>community-string</i> pim</p> <p>Example:</p> <pre>Device(config)# snmp-server host 10.10.10.10 traps public pim</pre>	Specifies the recipient of a PIM SNMP notification operation.

Configuration Examples for PIM MIB Extensions

Example Enabling PIM MIB Extensions for IP Multicast

The following example shows how to configure a router to generate notifications indicating that a PIM interface of the router has been enabled. The first line configures PIM traps to be sent as SNMP v2c traps to the host with IP address 10.0.0.1. The second line configures the router to send the neighbor-change class of trap notification to the host.

```
snmp-server host 10.0.0.1 traps version 2c public pim
snmp-server enable traps pim neighbor-change
interface ethernet0/0
 ip pim sparse-mode
```

Additional References for PIM MIB Extension for IP Multicast

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the IP Multicast Routing Commands section of the <i>Command Reference (Catalyst 9400 Series Switches)</i>

Standards and RFCs

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

Feature History for PIM MIB Extension for IP Multicast

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	PIM MIB Extension for IP Multicast	Protocol Independent Multicast (PIM) is an IP multicast routing protocol used for routing multicast data packets to multicast groups. RFC 2934 defines the PIM MIB for IPv4, which describes managed objects that enable users to remotely monitor and configure PIM using Simple Network Management Protocol (SNMP).

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>.



CHAPTER 10

Configuring MSDP

- [Prerequisites for Using MSDP to Interconnect Multiple PIM-SM Domains, on page 189](#)
- [Information About Using MSDP to Interconnect Multiple PIM-SM Domains, on page 189](#)
- [How to Use MSDP to Interconnect Multiple PIM-SM Domains, on page 203](#)
- [Configuration Examples for Using MSDP to Interconnect Multiple PIM-SM Domains, on page 221](#)
- [Additional References Multicast Source Discovery Protocol, on page 224](#)
- [Feature History for Multicast Source Discovery Protocol, on page 224](#)

Prerequisites for Using MSDP to Interconnect Multiple PIM-SM Domains

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

Information About Using MSDP to Interconnect Multiple PIM-SM Domains

This section provides information about using MSDP to interconnect multiple PIM-SM domains.

Benefits of Using MSDP to Interconnect Multiple PIM-SM Domains

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

Use of MSDP to Interconnect Multiple PIM-SM Domains

MSDP is a mechanism to connect multiple PIM-SM domains. The purpose of MSDP is to discover multicast sources in other PIM domains. The main advantage of MSDP is that it reduces the complexity of interconnecting multiple PIM-SM domains by allowing PIM-SM domains to use an interdomain source tree (rather than a common shared tree). When MSDP is configured in a network, RPs exchange source information with RPs

in other domains. An RP can join the interdomain source tree for sources that are sending to groups for which it has receivers. The RP can do that because it is the root of the shared tree within its domain, which has branches to all points in the domain where there are active receivers. When a last-hop device learns of a new source outside the PIM-SM domain (through the arrival of a multicast packet from the source down the shared tree), it then can send a join toward the source and join the interdomain source tree.



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

When MSDP is enabled, an RP in a PIM-SM domain maintains MSDP peering relationships with MSDP-enabled devices in other domains. This peering relationship occurs over a TCP connection, where primarily a list of sources sending to multicast groups is exchanged. MSDP uses TCP (port 639) for its peering connections. As with BGP, using point-to-point TCP peering means that each peer must be explicitly configured. The TCP connections between RPs, moreover, are achieved by the underlying routing system. The receiving RP uses the source lists to establish a source path. If the multicast sources are of interest to a domain that has receivers, multicast data is delivered over the normal, source-tree building mechanism provided by PIM-SM. MSDP is also used to announce sources sending to a group. These announcements must originate at the RP of the domain.



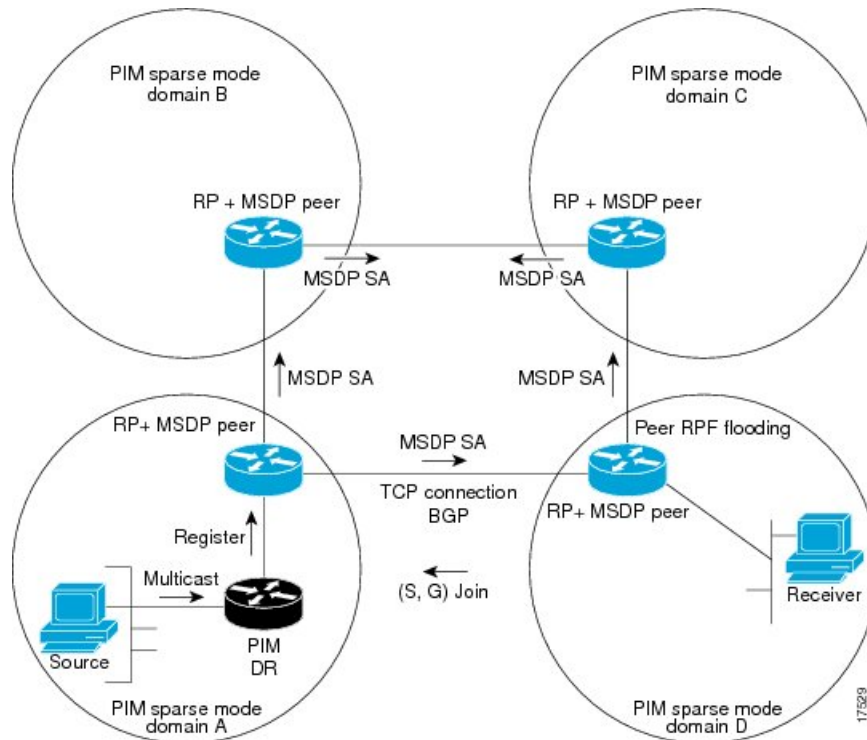
Note MSDP depends on BGP or multiprotocol BGP (MBGP) for interdomain operation. We recommended that you run MSDP on RPs sending to global multicast groups.

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



Note Although the following illustration and example uses routers in the configuration, any device (router or switch) can be used.

Figure 18: MSDP Running Between RP Peers



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

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



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

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

**Note**

- (M)BGP is not required in MSDP mesh group scenarios. For more information about MSDP mesh groups, see the [Configuring an MSDP Mesh Group, on page 210](#) section.
- (M)BGP is not required in default MSDP peer scenarios or in scenarios where only one MSDP peer is configured. For more information, see the [Configuring a Default MSDP Peer, on page 209](#) section.

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

**Note**

In all current and supported software releases, caching of MSDP SA messages is mandatory and cannot be manually enabled or disabled. By default, when an MSDP peer is configured, the **ip multicast cache-sa-state** command will automatically be added to the running configuration.

MSDP Message Types

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

SA Messages

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

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

**Note**

For more information about SA messages, see the [SA Message Origination Receipt and Processing, on page 193](#) section.

SA Request Messages

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



Note For more information about SA request messages, see the [Requesting Source Information from MSDP Peers, on page 214](#) section.

SA Response Messages

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



Note For more information about SA response messages, see the [Controlling the Response to Outgoing SA Request Messages from MSDP Peers Using SA Request Filters, on page 215](#) section.

Keepalive Messages

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



Note For more information about keepalive messages, see the [Adjusting the MSDP Keepalive and Hold-Time Intervals, on page 208](#) section.

SA Message Origination Receipt and Processing

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

SA Message Origination

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



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

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

SA Message Receipt

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

How RPF Check Rules Are Applied to SA Messages

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

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

RPF checks are not performed in the following cases:

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

How the Software Determines the Rule to Apply to RPF Checks

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

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

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

Rule 1 of RPF Checking of SA Messages in MSDP

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

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



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

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

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

Implications of Rule 1 of RPF Checking on MSDP

The MSDP topology must mirror the (M)BGP topology. In general, wherever there is an i(M)BGP peer connection between two devices, an MSDP peer connection should be configured. More specifically, the IP address of the far-end MSDP peer connection must be the same as the far-end i(M)BGP peer connection. The addresses must be the same because the BGP topology between i(M)BGP peers inside an autonomous system is not described by the AS path. If it were always the case that i(M)BGP peers updated the next-hop address in the path when sending an update to another i(M)BGP peer, then the peer could rely on the next-hop address to describe the i(M)BGP topology (and hence the MSDP topology). However, because the default behavior for i(M)BGP peers is to not update the next-hop address, the peer cannot rely on the next-hop address to describe the (M)BGP topology (MSDP topology). Instead, the i(M)BGP peer uses the address of the i(M)BGP peer that sent the path to describe the i(M)BGP topology (MSDP topology) inside the autonomous system.



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

Rule 2 of RPF Checking of SA Messages in MSDP

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

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

Implications of Rule 2 of RPF Checking on MSDP

The MSDP topology must mirror the (M)BGP topology. In general, wherever there is an e(M)BGP peer connection between two devices, an MSDP peer connection should be configured. As opposed to Rule 1, the

IP address of the far-end MSDP peer connection does not have to be the same as the far-end e(M)BGP peer connection. The reason that the addresses do not have to be identical is that BGP topology between two e(M)BGP peers is not described by the AS path.

Rule 3 of RPF Checking of SA Messages in MSDP

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

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



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

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

SA Message Processing

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

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



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

MSDP Peers

Like BGP, MSDP establishes neighbor relationships with other MSDP peers. MSDP peers connect using TCP port 639. The lower IP address peer takes the active role of opening the TCP connection. The higher IP address

peer waits in LISTEN state for the other to make the connection. MSDP peers send keepalive messages every 60 seconds. The arrival of data performs the same function as the keepalive message and keeps the session from timing out. If no keepalive messages or data is received for 75 seconds, the TCP connection is reset.

MSDP MD5 Password Authentication

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

How MSDP MD5 Password Authentication Works

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

Benefits of MSDP MD5 Password Authentication

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

SA Message Limits

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

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

MSDP Keepalive and Hold-Time Intervals

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

Once an MSDP peering session is established, each side of the connection sends a keepalive message and sets a keepalive timer. If the keepalive timer expires, the local MSDP peer sends a keepalive message and restarts its keepalive timer; this interval is referred to as the keepalive interval. The *keepalive-interval* argument

is used to adjust the interval for which keepalive messages will be sent. The keepalive timer is set to the value specified for the *keepalive-interval* argument when the peer comes up. The keepalive timer is reset to the value of the *keepalive-interval* argument whenever an MSDP keepalive message is sent to the peer and reset when the timer expires. The keepalive timer is deleted when an MSDP peering session is closed. By default, the keepalive timer is set to 60 seconds.



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

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

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

MSDP Connection-Retry Interval

You can adjust the interval at which all MSDP peers will wait after peering sessions are reset before attempting to reestablish the peering sessions. This interval is referred to as the connection-retry interval. By default, MSDP peers will wait 30 seconds after the session is reset before attempting to reestablish sessions with other peers. The modified configured connection-retry interval applies to all MSDP peering sessions on the device.

Default MSDP Peers

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

If your switch does not support BGP and MBGP, you cannot configure an MSDP peer on the local switch by using the **ip msdp peer** global configuration command. Instead, you define a default MSDP peer (by using the **ip msdp default-peer** global configuration command) which can accept all SA messages for the switch. The default MSDP peer must be a previously configured MSDP peer. Configure a default MSDP peer when the switch is not BGP- or MBGP-peering with an MSDP peer. If a single MSDP peer is configured, the switch always accepts all SA messages from that peer.

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

The figure illustrates a scenario where default MSDP peers might be used. In the figure, a customer that owns Device B is connected to the Internet through two Internet service providers (ISPs), one that owns Device A

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

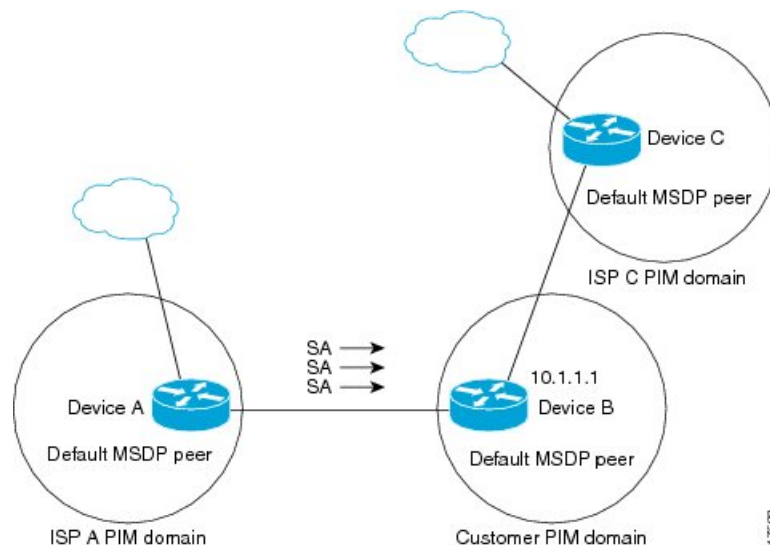
The ISP will also likely use a prefix list to define which prefixes it will accept from the customer device. The customer will define multiple default peers, each having one or more prefixes associated with it.

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



Note Although the following illustration and example uses routers in the configuration, any device (router or switch) can be used.

Figure 19: Default MSDP Peer Scenario



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

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

MSDP Mesh Groups

An MSDP mesh group is a group of MSDP speakers that have fully meshed MSDP connectivity between one another. In other words, each of the MSDP peers in the group must have an MSDP peering relationship (MSDP

connection) to every other MSDP peer in the group. When an MSDP mesh group is configured between a group of MSDP peers, SA message flooding is reduced. Because when an MSDP peer in the group receives an SA message from another MSDP peer in the group, it assumes that this SA message was sent to all the other MSDP peers in the group. As a result, it is not necessary for the receiving MSDP peer to flood the SA message to the other MSDP peers in the group.

Benefits of MSDP Mesh Groups

- Optimizes SA flooding--MSDP mesh groups are particularly useful for optimizing SA flooding when two or more peers are in a group.
- Reduces the amount of SA traffic across the Internet--When MSDP mesh groups are used, SA messages are not flooded to other mesh group peers.
- Eliminates RPF checks on arriving SA messages--When an MSDP mesh group is configured, SA messages are always accepted from mesh group peers.

SA Origination Filters

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

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

- You can configure an RP to prevent the device from advertising local sources in SA messages. The device will still forward SA messages from other MSDP peers in the normal fashion; it will just not originate any SA messages for local sources.
- You can configure the device to only originate SA messages for local sources sending to specific groups that match (S, G) pairs defined in the extended access list. All other local sources will not be advertised in SA messages.
- You can configure the device to only originate SA messages for local sources sending to specific groups that the match AS paths defined in an AS-path access list. All other local sources will not be advertised in SA messages.
- You can configure the device to only originate SA messages for local sources that match the criteria defined in the route map. All other local sources will not be advertised in SA messages.
- You configure an SA origination filter that includes an extended access list, an AS-path access list, and route map, or a combination thereof. In this case, all conditions must be true before any local sources are advertised in SA messages.

Use of Outgoing Filter Lists in MSDP

By default, an MSDP-enabled device forwards all SA messages it receives to all of its MSDP peers. However, you can prevent SA messages from being forwarded to MSDP peers by creating outgoing filter lists. Outgoing filter lists apply to all SA messages, whether locally originated or received from another MSDP peer, whereas

SA origination filters apply only to locally originated SA messages. For more information about enabling a filter for MSDP SA messages originated by the local device, see the [Controlling SA Messages Originated by an RP for Local Sources, on page 211](#) section.

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

- You can filter all outgoing SA messages forwarded to a specified MSDP peer by configuring the device to stop forwarding its SA messages to the MSDP peer.
- You can filter a subset of outgoing SA messages forwarded to a specified MSDP peer based on (S, G) pairs defined in an extended access list by configuring the device to only forward SA messages to the MSDP peer that match the (S, G) pairs permitted in an extended access list. The forwarding of all other SA messages to the MSDP peer will be stopped.
- You can filter a subset of outgoing SA messages forwarded to a specified MSDP peer based on match criteria defined in a route map by configuring the device to only forward SA messages that match the criteria defined in the route map. The forwarding of all other SA messages to the MSDP peer will be stopped.
- You can filter a subset of outgoing SA messages from a specified peer based on the announcing RP address contained in the SA message by configuring the device to filter outgoing SA messages based on their origin, even after an SA message has been transmitted across one or more MSDP peers. The forwarding of all other SA messages to the MSDP peer will be stopped.
- You can configure an outgoing filter list that includes an extended access list, a route map, and either an RP access list or an RP route map. In this case, all conditions must be true for the MSDP peer to forward the outgoing SA message.

**Caution**

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

Use of Incoming Filter Lists in MSDP

By default, an MSDP-enabled device receives all SA messages sent to it from its MSDP peers. However, you can control the source information that a device receives from its MSDP peers by creating incoming filter lists.

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

- You can filter all incoming SA messages from a specified MSDP peer by configuring the device to ignore all SA messages sent to it from the specified MSDP peer.
- You can filter a subset of incoming SA messages from a specified peer based on (S, G) pairs defined in an extended access list by configuring the device to only receive SA messages from the MSDP peer that match the (S, G) pairs defined in the extended access list. All other incoming SA messages from the MSDP peer will be ignored.
- You can filter a subset of incoming SA request messages from a specified peer based on match criteria defined in a route map by configuring the device to only receive SA messages that match the criteria defined in the route map. All other incoming SA messages from the MSDP peer will be ignored.

- You can filter a subset of incoming SA messages from a specified peer based on both (S, G) pairs defined in an extended access list and on match criteria defined in a route map by configuring the device to only receive incoming SA messages that both match the (S, G) pairs defined in the extended access list and match the criteria defined in the route map. All other incoming SA messages from the MSDP peer will be ignored.
- You can filter a subset of incoming SA messages from a specified peer based on the announcing RP address contained in the SA message by configuring the device to filter incoming SA messages based on their origin, even after the SA message may have already been transmitted across one or more MSDP peers.
- You can configure an incoming filter list that includes an extended access list, a route map, and either an RP access list or an RP route map. In this case, all conditions must be true for the MSDP peer to receive the incoming SA message.

**Caution**

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

TTL Thresholds in MSDP

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

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

SA Request Messages

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



Note In all current and supported software releases, caching of MSDP SA messages is mandatory and cannot be manually enabled or disabled. By default, when an MSDP peer is configured, the configured commands are automatically added to the running configuration.

SA Request Filters

By default, a device honors all outgoing SA request messages from its MSDP peers; that is, it sends cached source information to requesting MSDP peers in SA response messages. You can control the outgoing SA request messages that a device will honor from specified peers by creating an SA request filter. An SA request filter controls the outgoing SA requests that the device will honor from MSDP peers as follows:

- You can filter all SA request messages from a specified peer by configuring the device to ignore all SA requests from the specified MSDP peer.
- You can filter a subset of SA request messages from a specified peer based on groups defined in a standard access list by configuring the device to honor only SA request messages from the MSDP peer that match the groups defined in a standard access list. SA request messages from the specified peer for other groups will be ignored.

How to Use MSDP to Interconnect Multiple PIM-SM Domains

The first task is required; all other tasks are optional.

Configuring an MSDP Peer



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

Before you begin

- IP multicast routing must be enabled and PIM-SM must be configured.
- With the exception of a single MSDP peer, default MSDP peer, and MSDP mesh group scenarios, all MSDP peers must be configured to run BGP prior to being configured for MSDP.

Procedure

	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.

	Command or Action	Purpose
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ip msdp peer <i>{peer-name peer-address}</i> [<i>connect-source type number</i>] [remote-as <i>as-number</i>] Example: <pre>Device(config)# ip msdp peer 192.168.1.2 connect-source loopback0</pre>	<p>Enables MSDP and configures an MSDP peer as specified by the DNS name or IP address.</p> <p>Note The device that is selected to be configured as an MSDP peer is also usually a BGP neighbor. If it is not, see the Configuring a Default MSDP Peer, on page 209 section or the Configuring an MSDP Mesh Group, on page 210 section.</p> <ul style="list-style-type: none"> If you specify the connect-source keyword, the primary address of the specified local interface <i>type</i> and <i>number</i> values are used as the source IP address for the TCP connection. The connect-source keyword is recommended, especially for MSDP peers on a border that peer with a device inside of a remote domain.
Step 4	ip msdp description <i>{peer-name peer-address}</i> <i>text</i> Example: <pre>Device(config)# ip msdp description 192.168.1.2 router at customer a</pre>	(Optional) Configures a description for a specified peer to make it easier to identify in a configuration or in show command output.
Step 5	end Example: <pre>Device(config)# end</pre>	Exits global configuration mode and returns to privileged EXEC mode.

Shutting Down an MSDP Peer

Perform this optional task to shut down an MSDP peer.

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



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

Before you begin

MSDP is running and the MSDP peers must be configured.

Procedure

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

Configuring MSDP MD5 Password Authentication Between MSDP Peers

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

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ip msdp password peer {peer-name peer-address} [encryption-type] string Example: <pre>Device(config)# ip msdp password peer 10.32.43.144 0 test</pre>	<p>Enables MD5 password encryption for a TCP connection between two MSDP peers.</p> <p>Note MD5 authentication must be configured with the same password on both MSDP peers; otherwise, the connection between them will not be made.</p> <ul style="list-style-type: none"> • If you configure or change the password or key, which is used for MD5 authentication between two MSDP peers, the local device does not disconnect the existing session after you configure the password. You must manually disconnect the session to activate the new or changed password.
Step 4	exit Example: <pre>Device(config)# exit</pre>	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	show ip msdp peer [peer-address peer-name] Example: <pre>Device# show ip msdp peer</pre>	<p>(Optional) Displays detailed information about MSDP peers.</p> <p>Note Use this command to verify whether MD5 password authentication is enabled on an MSDP peer.</p>

Troubleshooting Tips

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

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

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

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

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

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

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



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

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ip msdp sa-limit <i>{peer-address peer-name}</i> <i>sa-limit</i> Example: <pre>Device(config)# ip msdp sa-limit 192.168.10.1 100</pre>	Limits the number of SA messages allowed in the SA cache from the specified MSDP.
Step 4	Repeat Step 3 to configure SA limits for additional MSDP peers.	--
Step 5	exit Example: <pre>Device(config)# exit</pre>	Exits global configuration mode and returns to privileged EXEC mode.
Step 6	show ip msdp count <i>[as-number]</i> Example: <pre>Device# show ip msdp count</pre>	(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.

	Command or Action	Purpose
Step 7	show ip msdp peer [<i>peer-address</i> <i>peer-name</i>] Example: <pre>Device# show ip msdp peer</pre>	(Optional) Displays detailed information about MSDP peers. Note The output of this command displays the number of SA messages received from MSDP peers that are stored in the cache.
Step 8	show ip msdp summary Example: <pre>Device# show ip msdp summary</pre>	(Optional) Displays MSDP peer status. Note The output of this command displays a per-peer “SA Count” field that displays the number of SAs stored in the cache.

Adjusting the MSDP Keepalive and Hold-Time Intervals

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



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

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ip msdp keepalive { <i>peer-address</i> <i>peer-name</i> } <i>keepalive-interval hold-time-interval</i> Example:	Configures the interval at which an MSDP peer will send keepalive messages and the interval at which the MSDP peer will wait for keepalive

	Command or Action	Purpose
	Device(config)# ip msdp keepalive 10.1.1.3 40 55	messages from other peers before declaring them down.
Step 4	Repeat Step 3 to adjust the keepalive message interval for additional MSDP peers.	--
Step 5	exit Example: Device(config)# exit	Exits global configuration mode and returns to privileged EXEC mode.

Adjusting the MSDP Connection-Retry Interval

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

Procedure

	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 msdp timer connection-retry-interval Example: Device# ip msdp timer 45	Configures the interval at which MSDP peers will wait after peering sessions are reset before attempting to reestablish the peering sessions.
Step 4	exit Example: Device(config)# exit	Exits global configuration mode and returns to privileged EXEC mode.

Configuring a Default MSDP Peer

Perform this optional task to configure a default MSDP peer.

Before you begin

An MSDP default peer must be a previously configured MSDP peer. Before configuring a default MSDP peer, you must first configure an MSDP peer.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ip msdp default-peer <i>{peer-address peer-name}</i> [prefix-list <i>list</i>] Example: <pre>Device(config)# ip msdp default-peer 192.168.1.3</pre>	Configures a default peer from which to accept all MSDP SA messages
Step 4	exit Example: <pre>Device(config)# exit</pre>	Exits global configuration mode and returns to privileged EXEC mode.

Configuring an MSDP Mesh Group

Perform this optional task to configure an MSDP mesh group.



Note You can configure multiple mesh groups per device.

Procedure

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

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip msdp mesh-group <i>mesh-name</i> <i>{peer-address peer-name}</i> Example: Device(config)# ip msdp mesh-group peermesh	Configures an MSDP mesh group and indicates that an MSDP peer belongs to that mesh group. Note All MSDP peers on a device that participate in a mesh group must be fully meshed with all other MSDP peers in the group. Each MSDP peer on each device must be configured as a peer using the ip msdp peer command and also as a member of the mesh group using the ip msdp mesh-group command.
Step 4	Repeat Step 3 to add MSDP peers as members of the mesh group.	--
Step 5	exit Example: Device(config)# exit	Exits global configuration mode and returns to privileged EXEC mode.

Controlling SA Messages Originated by an RP for Local Sources

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



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

Procedure

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

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

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

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



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

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> <code>enable</code>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 3	ip msdp sa-filter out { <i>peer-address</i> <i>peer-name</i> } [<i>list access-list</i>] [route-map	Enables a filter for outgoing MSDP messages.

	Command or Action	Purpose
	<code>map-name] [rp-list access-list rp-route-map map-name]</code> Example: <pre>Device(config)# ip msdp sa-filter out 192.168.1.5 peerone</pre>	
Step 4	Repeat Step 3 to configure outgoing filter lists for additional MSDP peers.	--
Step 5	exit Example: <pre>Device(config)# exit</pre>	Exits global configuration mode and returns to privileged EXEC mode.

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

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



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

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ip msdp sa-filter in {peer-address peer-name} [list access-list] [route-map map-name] [rp-list access-list rp-route-map map-name] Example: <pre>Device(config)# ip msdp sa-filter in 192.168.1.3</pre>	Enables a filter for incoming MSDP SA messages.

	Command or Action	Purpose
Step 4	Repeat Step 3 to configure incoming filter lists for additional MSDP peers.	--
Step 5	exit Example: <pre>Device(config)# exit</pre>	Exits global configuration mode and returns to privileged EXEC mode.

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

Perform this optional task to establish a time to live (TTL) threshold to limit the multicast data sent in SA messages.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ip msdp ttl-threshold <i>{peer-address peer-name} ttl-value</i> Example: Example: <pre>Device(config)# ip msdp ttl-threshold 192.168.1.5 8</pre>	Sets a TTL value for MSDP messages originated by the local device. <ul style="list-style-type: none"> • By default, multicast data packets in SA messages are sent to an MSDP peer, provided the TTL value of the packet is greater than 0, which is standard TTL behavior.
Step 4	exit Example: <pre>Device(config)# exit</pre>	Exits global configuration mode and returns to privileged EXEC mode.

Requesting Source Information from MSDP Peers

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



Note Because SA caching is enabled by default and cannot be explicitly enabled or disabled in earlier Cisco software releases, performing this task is seldom needed.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ip msdp sa-request <i>{peer-address peer-name}</i> Example: <pre>Device(config)# ip msdp sa-request 192.168.10.1</pre>	Specifies that the device send SA request messages to the specified MSDP peer.
Step 4	Repeat Step 3 to specify that the device send SA request messages to additional MSDP caching peers.	--
Step 5	exit Example: <pre>Device(config)# exit</pre>	Exits global configuration mode and returns to privileged EXEC mode.

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

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

Procedure

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

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

Configuring an Originating Address Other Than the RP Address

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

You can also change the originator ID for any one of the following reasons:

- If you configure multiple devices in an MSDP mesh group for Anycast RP.
- If you have a device that borders a PIM-SM domain and a PIM-DM domain. If a device borders a PIM-SM domain and a PIM-DM domain and you want to advertise active sources within the PIM-DM domain, configure the RP address in SA messages to be the address of the originating device's interface.

Before you begin

MSDP is enabled and the MSDP peers are configured. For more information about configuring MSDP peers, see the [Configuring an MSDP Peer, on page 203](#) section.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip msdp originator-id <i>type number</i> Example: Device(config)# ip msdp originator-id ethernet 1	Configures the RP address in SA messages to be the address of the originating device's interface.
Step 4	exit Example: Device(config)# exit	Exits global configuration mode and returns to privileged EXEC mode.

Monitoring MSDP

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

Procedure

Step 1

enable

Example:

```
Device# enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2

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

Use this command to debug MSDP activity.

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

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

Example:

```
Device# debug ip msdp
MSDP debugging is on
Device#
MSDP: 224.150.44.254: Received 1388-byte message from peer
MSDP: 224.150.44.254: SA TLV, len: 1388, ec: 115, RP: 172.31.3.92
MSDP: 224.150.44.254: Peer RPF check passed for 172.31.3.92, used EMBGP peer
MSDP: 224.150.44.250: Forward 1388-byte SA to peer
MSDP: 224.150.44.254: Received 1028-byte message from peer
MSDP: 224.150.44.254: SA TLV, len: 1028, ec: 85, RP: 172.31.3.92
```

```

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

```

Step 3 debug ip msdp resets

Use this command to debug MSDP peer reset reasons.

Example:

```
Device# debug ip msdp resets
```

Step 4 show ip msdp count [as-number]

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

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

Example:

```

Device# show ip msdp count
SA State per Peer Counters, <Peer>: <# SA learned>
    192.168.4.4: 8
SA State per ASN Counters, <asn>: <# sources>/<# groups>
    Total entries: 8
    ?: 8/8

```

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

Use this command to display detailed information about MSDP peers.

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

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

Example:

```

Device# show ip msdp peer 192.168.4.4
MSDP Peer 192.168.4.4 (?), AS 64512 (configured AS)
Connection status:
    State: Up, Resets: 0, Connection source: Loopback0 (2.2.2.2)
    Uptime(Downtime): 00:07:55, Messages sent/received: 8/18
    Output messages discarded: 0
    Connection and counters cleared 00:08:55 ago
SA Filtering:
    Input (S,G) filter: none, route-map: none

```



```

Input RP filter: none, route-map: none
Output (S,G) filter: none, route-map: none
Output RP filter: none, route-map: none
SA-Requests:
  Input filter: none
Peer ttl threshold: 0
SAs learned from this peer: 8
Input queue size: 0, Output queue size: 0
MD5 signature protection on MSDP TCP connection: not enabled

```

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

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

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

Example:

```

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

```

Step 7 **show ip msdp summary**

Use this command to display MSDP peer status.

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

Example:

```

Device# show ip msdp summary
MSDP Peer Status Summary
Peer Address      AS      State      Uptime/  Reset SA      Peer Name
                  Downtime Count Count
192.168.4.4       4       Up         00:08:05 0       8       ?

```

Clearing MSDP Connections Statistics and SA Cache Entries

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

Procedure

	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.

	Command or Action	Purpose
Step 2	clear ip msdp peer [<i>peer-address</i> <i>peer-name</i>] Example: Device# clear ip msdp peer	Clears the TCP connection to the specified MSDP peer and resets all MSDP message counters.
Step 3	clear ip msdp statistics [<i>peer-address</i> <i>peer-name</i>] Example: Device# clear ip msdp statistics	Clears the statistics counters for the specified MSDP peer and resets all MSDP message counters.
Step 4	clear ip msdp sa-cache [<i>group-address</i>] Example: Device# clear ip msdp sa-cache	Clears SA cache entries. <ul style="list-style-type: none"> • If the clear ip msdp sa-cache is specified with the optional <i>group-address</i> argument or <i>source-address</i> argument, all SA cache entries are cleared. • Use the optional <i>group-address</i> argument to clear all SA cache entries associated with a specific group.

Enabling SNMP Monitoring of MSDP

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

Before you begin

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



Note

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

Procedure

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

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

Troubleshooting Tips

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

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

This section provides configuration examples of using MSDP to interconnect multiple PIM-SM domains.

Example: Configuring an MSDP Peer

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

Device A

```
!
interface Loopback 0
 ip address 10.220.8.1 255.255.255.255
!
```

```
ip msdp peer 10.220.16.1 connect-source Loopback0
ip msdp peer 10.220.32.1 connect-source Loopback0
!
```

Device B

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

Device C

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

Example: Configuring MSDP MD5 Password Authentication

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

Device A

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

Device B

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

Example: Configuring a Default MSDP Peer

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

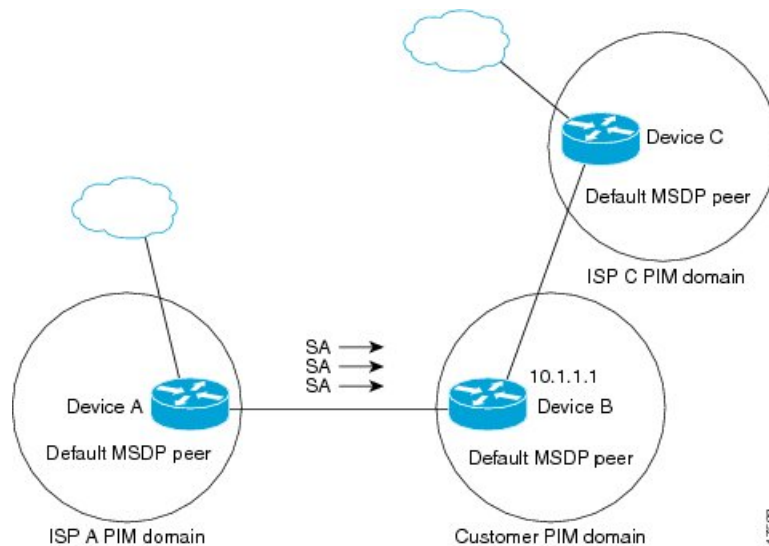
The ISP will also likely use a prefix list to define which prefixes it will accept from the customer device. The customer will define multiple default peers, each having one or more prefixes associated with it.

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



Note Although the following illustration and example uses routers in the configuration, any device (router or switch) can be used.

Figure 20: Default MSDP Peer Scenario



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

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

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

Device A Configuration

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

Device C Configuration

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

Example: Configuring MSDP Mesh Groups

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

Device A Configuration

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

Device B Configuration

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

Device C Configuration

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

Additional References Multicast Source Discovery Protocol

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the IP Multicast Routing Commands section of the <i>Command Reference (Catalyst 9400 Series Switches)</i> .

Feature History for Multicast Source Discovery Protocol

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	Multicast Source Discovery Protocol	MSDP is a mechanism to connect multiple PIM-SM domains. The purpose of MSDP is to discover multicast sources in other PIM domains. The main advantage of MSDP is that it reduces the complexity of interconnecting multiple PIM-SM domains by allowing PIM-SM domains to use an interdomain source tree (rather than a common shared tree).

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>.



CHAPTER 11

Configuring SSM

- [Prerequisites for Configuring SSM, on page 227](#)
- [Restrictions for Configuring SSM, on page 227](#)
- [Information About SSM, on page 229](#)
- [How to Configure SSM, on page 232](#)
- [Monitoring SSM, on page 238](#)
- [Where to Go Next for SSM, on page 239](#)
- [Additional References for SSM, on page 239](#)
- [Feature History for SSM, on page 239](#)

Prerequisites for Configuring SSM

The following are the prerequisites for configuring source-specific multicast (SSM) and SSM mapping:

- Before you configure SSM mapping, you must perform the following tasks:
 - Enable IP multicast routing.
 - Enable PIM sparse mode.
 - Configure SSM.
- Before you configure static SSM mapping, you must configure access control lists (ACLs) that define the group ranges to be mapped to source addresses.
- Before you can configure and use SSM mapping with DNS lookups, you need to add records to a running DNS server. If you do not already have a DNS server running, you need to install one.



Note You can use a product such as *Cisco Network Registrar* to add records to a running DNS server.

Restrictions for Configuring SSM

The following are the restrictions for configuring SSM:

- To run SSM with IGMPv3, SSM must be supported in the Cisco IOS router, the host where the application is running, and the application itself.
- Existing applications in a network predating SSM will not work within the SSM range unless they are modified to support (S, G) channel subscriptions. Therefore, enabling SSM in a network may cause problems for existing applications if they use addresses within the designated SSM range.
- IGMP Snooping—IGMPv3 uses new membership report messages that might not be correctly recognized by older IGMP snooping devices.
- Address management is still necessary to some degree when SSM is used with Layer 2 switching mechanisms. Cisco Group Management Protocol (CGMP), IGMP snooping, or Router-Port Group Management Protocol (RGMP) support only group-specific filtering, not (S, G) channel-specific filtering. If different receivers in a switched network request different (S, G) channels sharing the same group, they do not benefit from these existing mechanisms. Instead, both receivers receive all (S, G) channel traffic and filter out the unwanted traffic on input. Because SSM can re-use the group addresses in the SSM range for many independent applications, this situation can lead to decreased traffic filtering in a switched network. For this reason, it is important to use random IP addresses from the SSM range for an application to minimize the chance for re-use of a single address within the SSM range between different applications. For example, an application service providing a set of television channels should, even with SSM, use a different group for each television (S, G) channel. This setup guarantees that multiple receivers to different channels within the same application service never experience traffic aliasing in networks that include Layer 2 devices.
- In PIM-SSM, the last hop router will continue to periodically send (S, G) join messages if appropriate (S, G) subscriptions are on the interfaces. Therefore, as long as receivers send (S, G) subscriptions, the shortest path tree (SPT) state from the receivers to the source will be maintained, even if the source is not sending traffic for longer periods of time (or even never).

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

The following are the restrictions for configuring SSM mapping:

- The SSM Mapping feature does not share the benefit of full SSM. SSM mapping takes a group G join from a host and identifies this group with an application associated with one or more sources, therefore, it can only support one such application per group G. Nevertheless, full SSM applications may still share the same group also used in SSM mapping.
- Enable IGMPv3 with care on the last hop router when you rely solely on SSM mapping as a transition solution for full SSM. When you enable both SSM mapping and IGMPv3 and the hosts already support IGMPv3 (but not SSM), the hosts send IGMPv3 group reports. SSM mapping does not support these IGMPv3 group reports, and the router does not correctly associate sources with these reports.

Information About SSM

The source-specific multicast (SSM) feature is an extension of IP multicast in which datagram traffic is forwarded to receivers from only those multicast sources that the receivers have explicitly joined. For multicast groups configured for SSM, only SSM distribution trees (no shared trees) are created.

This section describes how to configure source-specific multicast (SSM). For a complete description of the SSM commands in this section, refer to the *IP Multicast Command Reference*.

SSM Components Overview

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

- Protocol independent multicast source-specific mode (PIM-SSM)

PIM-SSM is the routing protocol that supports the implementation of SSM and is derived from PIM sparse mode (PIM-SM).

- Internet Group Management Protocol version 3 (IGMPv3)

SSM and Internet Standard Multicast (ISM)

The current IP multicast infrastructure in the Internet and many enterprise intranets is based on the PIM-SM protocol and Multicast Source Discovery Protocol (MSDP). These protocols have the limitations of the Internet Standard Multicast (ISM) service model. For example, with ISM, the network must maintain knowledge about which hosts in the network are actively sending multicast traffic.

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

In SSM, delivery of datagrams is based on (S, G) channels. In both SSM and ISM, no signaling is required to become a source. However, in SSM, receivers must subscribe or unsubscribe to (S, G) channels to receive or not receive traffic from specific sources. In other words, receivers can receive traffic only from (S, G) channels to which they are subscribed, whereas in ISM, receivers need not know the IP addresses of sources from which they receive their traffic. The proposed standard approach for channel subscription signaling uses IGMP and includes modes membership reports, which are supported only in IGMP version 3.

SSM IP Address Range

SSM can coexist with the ISM service by applying the SSM delivery model to a configured subset of the IP multicast group address range. Cisco IOS software allows SSM configuration for the IP multicast address range of 224.0.0.0 through 239.255.255.255. When an SSM range is defined, existing IP multicast receiver applications do not receive any traffic when they try to use an address in the SSM range (unless the application is modified to use an explicit (S, G) channel subscription).

SSM Operations

An established network, in which IP multicast service is based on PIM-SM, can support SSM services. SSM can also be deployed alone in a network without the full range of protocols required for interdomain PIM-SM (for example, MSDP, Auto-RP, or bootstrap router [BSR]) if only SSM service is needed.

If SSM is deployed in a network already configured for PIM-SM, only the last-hop routers support SSM. Routers that are not directly connected to receivers do not require support for SSM. In general, these not-last-hop routers must only run PIM-SM in the SSM range and might need additional access control configuration to suppress MSDP signalling, registering, or PIM-SM shared tree operations from occurring within the SSM range.

Use the **ip pim ssm** global configuration command to configure the SSM range and to enable SSM. This configuration has the following effects:

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

SSM Mapping

In a typical set-top box (STB) deployment, each TV channel uses one separate IP multicast group and has one active server host sending the TV channel. A single server can send multiple TV channels, but each to a different group. In this network environment, if a router receives an IGMPv1 or IGMPv2 membership report for a particular group, the report addresses the well-known TV server for the TV channel associated with the multicast group.

When SSM mapping is configured, if a router receives an IGMPv1 or IGMPv2 membership report for a particular group, the router translates this report into one or more channel memberships for the well-known sources associated with this group.

When the router receives an IGMPv1 or IGMPv2 membership report for a group, the router uses SSM mapping to determine one or more source IP addresses for the group. SSM mapping then translates the membership report as an IGMPv3 report and continues as if it had received an IGMPv3 report. The router then sends PIM joins and continues to be joined to these groups as long as it continues to receive the IGMPv1 or IGMPv2 membership reports, and the SSM mapping for the group remains the same.

SSM mapping enables the last hop router to determine the source addresses either by a statically configured table on the router or through a DNS server. When the statically configured table or the DNS mapping changes, the router leaves the current sources associated with the joined groups.

Static SSM Mapping

With static SSM mapping, you can configure the last hop router to use a static map to determine the sources that are sending to groups. Static SSM mapping requires that you configure ACLs to define group ranges.

After configuring the ACLs to define group ranges, you can then map the groups permitted by those ACLs to sources by using the **ip igmp ssm-map static** global configuration command.

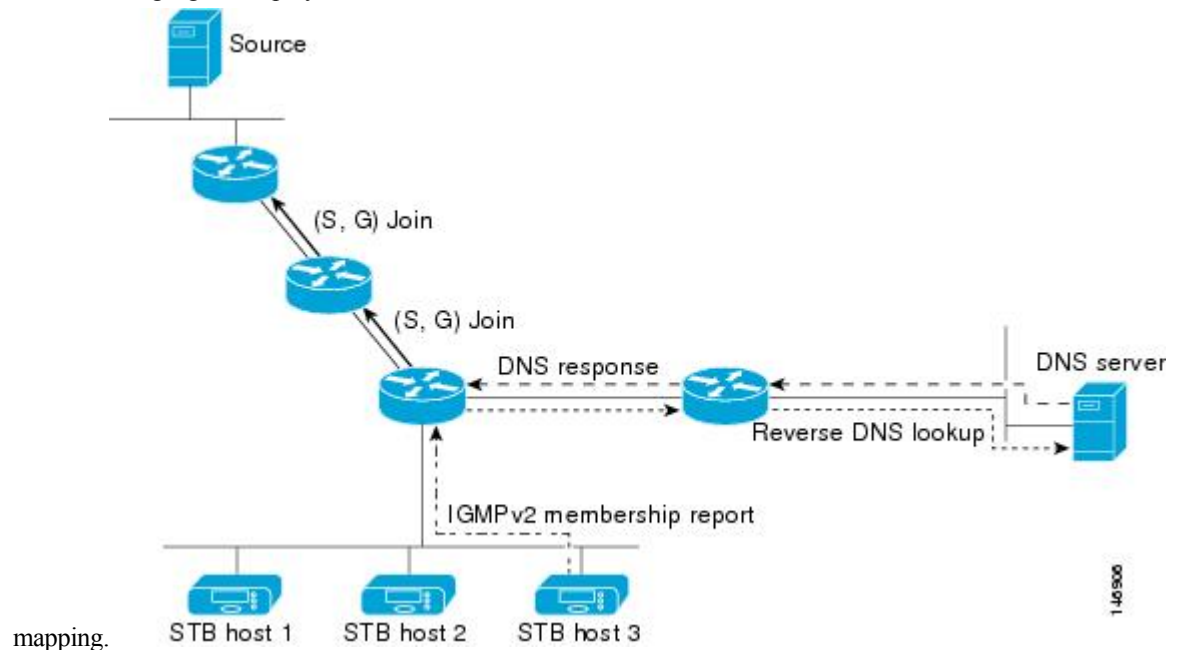
You can configure static SSM mapping in smaller networks when a DNS is not needed or to locally override DNS mappings. When configured, static SSM mappings take precedence over DNS mappings.

DNS-Based SSM Mapping

You can use DNS-based SSM mapping to configure the last hop router to perform a reverse DNS lookup to determine sources sending to groups. When DNS-based SSM mapping is configured, the router constructs a domain name that includes the group address and performs a reverse lookup into the DNS. The router looks up IP address resource records and uses them as the source addresses associated with this group. SSM mapping supports up to 20 sources for each group. The router joins all sources configured for a group.

Figure 21: DNS-Based SSM Mapping

The following figure displays DNS-based SSM



The SSM mapping mechanism that enables the last hop router to join multiple sources for a group can provide source redundancy for a TV broadcast. In this context, the last hop router provides redundancy using SSM mapping to simultaneously join two video sources for the same TV channel. However, to prevent the last hop router from duplicating the video traffic, the video sources must use a server-side switchover mechanism. One video source is active, and the other backup video source is passive. The passive source waits until an active source failure is detected before sending the video traffic for the TV channel. Thus, the server-side switchover mechanism ensures that only one of the servers is actively sending video traffic for the TV channel.

To look up one or more source addresses for a group that includes G1, G2, G3, and G4, you must configure these DNS records on the DNS server:

```
G4.G3.G2.G1 [multicast-domain] [timeout] IN A source-address-1
IN A source-address-2
IN A source-address-n
```

See your DNS server documentation for more information about configuring DNS resource records.

How to Configure SSM

Configuring SSM

Follow these steps to configure SSM:

This procedure is optional.

Before you begin

If you want to use an access list to define the Source Specific Multicast (SSM) range, configure the access list before you reference the access list in the **ip pim ssm** command.

Procedure

	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 pim ssm [default range <i>access-list</i>] Example: Device(config)# ip pim ssm range 20	Defines the SSM range of IP multicast addresses.
Step 4	interface <i>type number</i> Example: Device(config)# interface gigabitethernet 1/0/1	Selects an interface that is connected to hosts on which IGMPv3 can be enabled, and enters the interface configuration mode. The specified interface must be one of the following: <ul style="list-style-type: none"> • A routed port—A physical port that has been configured as a Layer 3 port by entering the no switchport interface configuration command.

	Command or Action	Purpose
		<ul style="list-style-type: none"> An SVI—A VLAN interface created by using the interface vlan <i>vlan-id</i> global configuration command. <p>These interfaces must have IP addresses assigned to them.</p>
Step 5	ip pim {sparse-mode } Example: <pre>Device(config-if) # ip pim sparse-mode</pre>	Enables PIM on an interface.
Step 6	ip igmp version 3 Example: <pre>Device(config-if) # ip igmp version 3</pre>	Enables IGMPv3 on this interface. The default version of IGMP is set to Version 2.
Step 7	end Example: <pre>Device(config) # end</pre>	Returns to privileged EXEC mode.
Step 8	show running-config Example: <pre>Device# show running-config</pre>	Verifies your entries.
Step 9	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Configuring Source Specific Multicast Mapping

The Source Specific Multicast (SSM) mapping feature supports SSM transition when supporting SSM on the end system is impossible or unwanted due to administrative or technical reasons. You can use SSM mapping to leverage SSM for video delivery to legacy STBs that do not support IGMPv3 or for applications that do not use the IGMPv3 host stack.

Configuring Static SSM Mapping

Follow these steps to configure static SSM Mapping:

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ip igmp ssm-map enable Example: <pre>Device(config)# ip igmp ssm-map enable</pre>	Enables SSM mapping for groups in the configured SSM range. Note By default, this command enables DNS-based SSM mapping.
Step 4	no ip igmp ssm-map query dns Example: <pre>Device(config)# no ip igmp ssm-map query dns</pre>	(Optional) Disables DNS-based SSM mapping. Note Disable DNS-based SSM mapping if you only want to rely on static SSM mapping. By default, the ip igmp ssm-map command enables DNS-based SSM mapping.
Step 5	ip igmp ssm-map static <i>access-list</i> <i>source-address</i> Example: <pre>Device(config)# ip igmp ssm-map static 11 172.16.8.11</pre>	Configures static SSM mapping. <ul style="list-style-type: none"> • The ACL supplied for the <i>access-list</i> argument defines the groups to be mapped to the source IP address entered for the <i>source-address</i> argument. Note You can configure additional static SSM mappings. If additional SSM mappings are configured and the router receives an IGMPv1 or IGMPv2 membership report for a group in the SSM range, the device determines the source addresses associated with the group by walking each configured ip igmp ssm-map static command. The device associates up to 20 sources per group. Repeat Step to configure additional static SSM mappings, if required.

	Command or Action	Purpose
Step 6	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 7	show running-config Example: <pre>Device# show running-config</pre>	Verifies your entries.
Step 8	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Configuring DNS-Based SSM Mapping

To configure DNS-based SSM mapping, you need to create a DNS server zone or add records to an existing zone. If the routers that are using DNS-based SSM mapping are also using DNS for other purposes, you should use a normally configured DNS server. If DNS-based SSM mapping is the only DNS implementation being used on the router, you can configure a false DNS setup with an empty root zone or a root zone that points back to itself.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ip igmp ssm-map enable Example: <pre>Device(config)# ip igmp ssm-map enable</pre>	Enables SSM mapping for groups in a configured SSM range.
Step 4	ip igmp ssm-map query dns	(Optional) Enables DNS-based SSM mapping.

	Command or Action	Purpose
	Example: <pre>Device(config)# ip igmp ssm-map query dns</pre>	<ul style="list-style-type: none"> By default, the ip igmp ssm-map command enables DNS-based SSM mapping. Only the no form of this command is saved to the running configuration. <p>Note Use this command to reenable DNS-based SSM mapping if DNS-based SSM mapping is disabled.</p>
Step 5	ip domain multicast <i>domain-prefix</i> Example: <pre>Device(config)# ip domain multicast ssm-map.cisco.com</pre>	(Optional) Changes the domain prefix used for DNS-based SSM mapping. <ul style="list-style-type: none"> By default, the software uses the ip-addr.arpa domain prefix.
Step 6	ip name-server <i>server-address1</i> [<i>server-address2...server-address6</i>] Example: <pre>Device(config)# ip name-server 10.48.81.21</pre>	Specifies the address of one or more name servers to use for name and address resolution.
Step 7	Repeat Step 6 to configure additional DNS servers for redundancy, if required.	
Step 8	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 9	show running-config Example: <pre>Device# show running-config</pre>	Verifies your entries.
Step 10	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Configuring Static Traffic Forwarding with SSM Mapping

Follow these steps to configure static traffic forwarding with SSM mapping on the last hop router:

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: <pre>Device(config)# interface gigabitethernet 1/0/1</pre>	Selects an interface on which to statically forward traffic for a multicast group using SSM mapping, and enters interface configuration mode. The specified interface must be one of the following: <ul style="list-style-type: none"> • A routed port—A physical port that has been configured as a Layer 3 port by entering the no switchport interface configuration command. • An SVI—A VLAN interface created by using the interface vlan <i>vlan-id</i> global configuration command. These interfaces must have IP addresses assigned to them. Note Static forwarding of traffic with SSM mapping works with either DNS-based SSM mapping or statically configured SSM mapping.
Step 4	ip igmp static-group <i>group-address</i> source <i>ssm-map</i> Example: <pre>Device(config-if)# ip igmp static-group 239.1.2.1 source ssm-map</pre>	Configures SSM mapping to statically forward a (S, G) channel from the interface. Use this command if you want to statically forward SSM traffic for certain groups. Use DNS-based SSM mapping to determine the source addresses of the channels.
Step 5	end Example:	Returns to privileged EXEC mode.

	Command or Action	Purpose
	Device(config)# end	
Step 6	show running-config Example: Device# show running-config	Verifies your entries.
Step 7	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Monitoring SSM

Use the privileged EXEC commands in the following table to monitor SSM.

Table 18: Commands for Monitoring SSM

Command	Purpose
show ip igmp groups detail	Displays the (S, G) channel subscription through IGMPv3.
show ip mroute	Displays whether a multicast group supports SSM service or whether a source-specific host report was received.

Monitoring SSM Mapping

Use the privileged EXEC commands in the following table to monitor SSM mapping.

Table 19: SSM Mapping Monitoring Commands

Command	Purpose
show ip igmp ssm-mapping	Displays information about SSM mapping.
show ip igmp ssm-mapping group-address	Displays the sources that SSM mapping uses for group.
show ip igmp groups [group-name group-address interface-type interface-number] [detail]	Displays the multicast groups with receivers that connected to the router and that were learned through SSM mapping.

Command	Purpose
show host	Displays the default domain name, the style of service, a list of name server hosts, and the cache hostnames and addresses.
debug ip igmp <i>group-address</i>	Displays the IGMP packets received and sent on host-related events.

Where to Go Next for SSM

You can configure the following:

- IGMP
- PIM
- IP Multicast Routing
- Service Discovery Gateway

Additional References for SSM

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the IP Multicast Routing Commands section of the <i>Command Reference (Catalyst 9400 Series Switches)</i>

Standards and RFCs

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

Feature History for SSM

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	SSM	SSM is an extension of IP multicast in which datagram traffic is forwarded to receivers from only those multicast sources that the receivers have explicitly joined. For multicast groups configured for SSM, only SSM distribution trees (no shared trees) are created.

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>.



CHAPTER 12

Implementing IPv6 Multicast

- [Information About Implementing IPv6 Multicast Routing, on page 241](#)
- [Implementing IPv6 Multicast, on page 249](#)
- [Additional References, on page 271](#)
- [Feature History for IPv6 Multicast, on page 271](#)

Information About Implementing IPv6 Multicast Routing

This chapter describes how to implement IPv6 multicast routing on the switch.

Traditional IP communication allows a host to send packets to a single host (unicast transmission) or to all hosts (broadcast transmission). IPv6 multicast provides a third scheme, allowing a host to send a single data stream to a subset of all hosts (group transmission) simultaneously.

IPv6 Multicast Overview

An IPv6 multicast group is an arbitrary group of receivers that want to receive a particular data stream. This group has no physical or geographical boundaries--receivers can be located anywhere on the Internet or in any private network. Receivers that are interested in receiving data flowing to a particular group must join the group by signaling their local switch. This signaling is achieved with the MLD protocol.

Switches use the MLD protocol to learn whether members of a group are present on their directly attached subnets. Hosts join multicast groups by sending MLD report messages. The network then delivers data to a potentially unlimited number of receivers, using only one copy of the multicast data on each subnet. IPv6 hosts that wish to receive the traffic are known as group members.

Packets delivered to group members are identified by a single multicast group address. Multicast packets are delivered to a group using best-effort reliability, just like IPv6 unicast packets.

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

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



Note As per RFC [4291](#), the FF0x::/12 (where the T flag is set to 0 in IPv6 destination address) is for permanently assigned (“well-known”) IPv6 multicast address range.

In Cisco Catalyst 9400 Series Switches, the default behavior for packets with this address range is to flood in the ingress VLAN.

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

How active a multicast group is, its duration, and its membership can vary from group to group and from time to time. A group that has members may have no activity.

IPv6 Multicast Routing Implementation

The Cisco IOS software supports the following protocols to implement IPv6 multicast routing:

- MLD is used by IPv6 switches to discover multicast listeners (nodes that want to receive multicast packets destined for specific multicast addresses) on directly attached links. There are two versions of MLD: MLD version 1 is based on version 2 of the Internet Group Management Protocol (IGMP) for IPv4, and MLD version 2 is based on version 3 of the IGMP for IPv4. IPv6 multicast for Cisco IOS software uses both MLD version 2 and MLD version 1. MLD version 2 is fully backward-compatible with MLD version 1 (described in RFC 2710). Hosts that support only MLD version 1 will interoperate with a switch running MLD version 2. Mixed LANs with both MLD version 1 and MLD version 2 hosts are likewise supported.
- PIM-SM is used between switches so that they can track which multicast packets to forward to each other and to their directly connected LANs.
- PIM in Source Specific Multicast (PIM-SSM) is similar to PIM-SM with the additional ability to report interest in receiving packets from specific source addresses (or from all but the specific source addresses) to an IP multicast address.

IPv6 Multicast Listener Discovery Protocol

To start implementing multicasting in the campus network, users must first define who receives the multicast. The MLD protocol is used by IPv6 switches to discover the presence of multicast listeners (for example, nodes that want to receive multicast packets) on their directly attached links, and to discover specifically which multicast addresses are of interest to those neighboring nodes. It is used for discovering local group and source-specific group membership.

The MLD protocol provides a means to automatically control and limit the flow of multicast traffic throughout your network with the use of special multicast queriers and hosts.

Multicast Queriers and Hosts

A multicast querier is a network device, such as a switch, that sends query messages to discover which network devices are members of a given multicast group.

A multicast host is a receiver, including switches, that send report messages to inform the querier of a host membership.

A set of queriers and hosts that receive multicast data streams from the same source is called a multicast group. Queriers and hosts use MLD reports to join and leave multicast groups and to begin receiving group traffic.

MLD uses the Internet Control Message Protocol (ICMP) to carry its messages. All MLD messages are link-local with a hop limit of 1, and they all have the switch alert option set. The switch alert option implies an implementation of the hop-by-hop option header.

MLD Access Group

The MLD access group provides receiver access control in Cisco IOS IPv6 multicast switches. This feature limits the list of groups a receiver can join, and it allows or denies sources used to join SSM channels.

Explicit Tracking of Receivers

The explicit tracking feature allows a switch to track the behavior of the hosts within its IPv6 network. This feature also enables the fast leave mechanism to be used with MLD version 2 host reports.

Protocol Independent Multicast

Protocol Independent Multicast (PIM) is used between switches so that they can track which multicast packets to forward to each other and to their directly connected LANs. PIM works independently of the unicast routing protocol to perform send or receive multicast route updates like other protocols. Regardless of which unicast routing protocols are being used in the LAN to populate the unicast routing table, Cisco IOS PIM uses the existing unicast table content to perform the Reverse Path Forwarding (RPF) check instead of building and maintaining its own separate routing table.

You can configure IPv6 multicast to use either PIM-SM or PIM-SSM operation, or you can use both PIM-SM and PIM-SSM together in your network.

PIM-Sparse Mode

IPv6 multicast provides support for intradomain multicast routing using PIM-SM. PIM-SM uses unicast routing to provide reverse-path information for multicast tree building, but it is not dependent on any particular unicast routing protocol.

PIM-SM is used in a multicast network when relatively few switches are involved in each multicast and these switches do not forward multicast packets for a group, unless there is an explicit request for the traffic. PIM-SM distributes information about active sources by forwarding data packets on the shared tree. PIM-SM initially uses shared trees, which requires the use of an RP.

Requests are accomplished via PIM joins, which are sent hop by hop toward the root node of the tree. The root node of a tree in PIM-SM is the RP in the case of a shared tree or the first-hop switch that is directly connected to the multicast source in the case of a shortest path tree (SPT). The RP keeps track of multicast groups and the hosts that send multicast packets are registered with the RP by that host's first-hop switch.

As a PIM join travels up the tree, switches along the path set up multicast forwarding state so that the requested multicast traffic will be forwarded back down the tree. When multicast traffic is no longer needed, a switch sends a PIM prune up the tree toward the root node to prune (or remove) the unnecessary traffic. As this PIM prune travels hop by hop up the tree, each switch updates its forwarding state appropriately. Ultimately, the forwarding state associated with a multicast group or source is removed.

A multicast data sender sends data destined for a multicast group. The designated switch (DR) of the sender takes those data packets, unicast-encapsulates them, and sends them directly to the RP. The RP receives these encapsulated data packets, de-encapsulates them, and forwards them onto the shared tree. The packets then

follow the (*, G) multicast tree state in the switches on the RP tree, being replicated wherever the RP tree branches, and eventually reaching all the receivers for that multicast group. The process of encapsulating data packets to the RP is called registering, and the encapsulation packets are called PIM register packets.

IPv6 BSR: Configure RP Mapping

PIM switches in a domain must be able to map each multicast group to the correct RP address. The BSR protocol for PIM-SM provides a dynamic, adaptive mechanism to distribute group-to-RP mapping information rapidly throughout a domain. With the IPv6 BSR feature, if an RP becomes unreachable, it will be detected and the mapping tables will be modified so that the unreachable RP is no longer used, and the new tables will be rapidly distributed throughout the domain.

Every PIM-SM multicast group needs to be associated with the IP or IPv6 address of an RP. When a new multicast sender starts sending, its local DR will encapsulate these data packets in a PIM register message and send them to the RP for that multicast group. When a new multicast receiver joins, its local DR will send a PIM join message to the RP for that multicast group. When any PIM switch sends a (*, G) join message, the PIM switch needs to know which is the next switch toward the RP so that G (Group) can send a message to that switch. Also, when a PIM switch is forwarding data packets using (*, G) state, the PIM switch needs to know which is the correct incoming interface for packets destined for G, because it needs to reject any packets that arrive on other interfaces.

A small set of switches from a domain are configured as candidate bootstrap switches (C-BSRs) and a single BSR is selected for that domain. A set of switches within a domain are also configured as candidate RPs (C-RPs); typically, these switches are the same switches that are configured as C-BSRs. Candidate RPs periodically unicast candidate-RP-advertisement (C-RP-Adv) messages to the BSR of that domain, advertising their willingness to be an RP. A C-RP-Adv message includes the address of the advertising C-RP, and an optional list of group addresses and mask length fields, indicating the group prefixes for which the candidacy is advertised. The BSR then includes a set of these C-RPs, along with their corresponding group prefixes, in bootstrap messages (BSMs) it periodically originates. BSMs are distributed hop-by-hop throughout the domain.

Bidirectional BSR support allows bidirectional RPs to be advertised in C-RP messages and bidirectional ranges in the BSM. All switches in a system must be able to use the bidirectional range in the BSM; otherwise, the bidirectional RP feature will not function.

PIM-Source Specific Multicast

PIM-SSM is the routing protocol that supports the implementation of SSM and is derived from PIM-SM. However, unlike PIM-SM where data from all multicast sources are sent when there is a PIM join, the SSM feature forwards datagram traffic to receivers from only those multicast sources that the receivers have explicitly joined, thus optimizing bandwidth utilization and denying unwanted Internet broadcast traffic. Further, instead of the use of RP and shared trees, SSM uses information found on source addresses for a multicast group. This information is provided by receivers through the source addresses relayed to the last-hop switches by MLD membership reports, resulting in shortest-path trees directly to the sources.

In SSM, delivery of datagrams is based on (S, G) channels. Traffic for one (S, G) channel consists of datagrams with an IPv6 unicast source address S and the multicast group address G as the IPv6 destination address. Systems will receive this traffic by becoming members of the (S, G) channel. Signaling is not required, but receivers must subscribe or unsubscribe to (S, G) channels to receive or not receive traffic from specific sources.

MLD version 2 is required for SSM to operate. MLD allows the host to provide source information. Before SSM can run with MLD, SSM must be supported in the Cisco IOS IPv6 switch, the host where the application is running, and the application itself.

Routable Address Hello Option

When an IPv6 interior gateway protocol is used to build the unicast routing table, the procedure to detect the upstream switch address assumes the address of a PIM neighbor is always same as the address of the next-hop switch, as long as they refer to the same switch. However, it may not be the case when a switch has multiple addresses on a link.

Two typical situations can lead to this situation for IPv6. The first situation can occur when the unicast routing table is not built by an IPv6 interior gateway protocol such as multicast BGP. The second situation occurs when the address of an RP shares a subnet prefix with downstream switches (note that the RP switch address has to be domain-wide and therefore cannot be a link-local address).

The routable address hello option allows the PIM protocol to avoid such situations by adding a PIM hello message option that includes all the addresses on the interface on which the PIM hello message is advertised. When a PIM switch finds an upstream switch for some address, the result of RPF calculation is compared with the addresses in this option, in addition to the PIM neighbor's address itself. Because this option includes all the possible addresses of a PIM switch on that link, it always includes the RPF calculation result if it refers to the PIM switch supporting this option.

Because of size restrictions on PIM messages and the requirement that a routable address hello option fits within a single PIM hello message, a limit of 16 addresses can be configured on the interface.

PIM IPv6 Stub Routing

The PIM stub routing feature reduces resource usage by moving routed traffic closer to the end user.

In a network using PIM stub routing, the only allowable route for IPv6 traffic to the user is through a switch that is configured with PIM stub routing. PIM passive interfaces are connected to Layer 2 access domains, such as VLANs, or to interfaces that are connected to other Layer 2 devices. Only directly connected multicast receivers and sources are allowed in the Layer 2 access domains. The PIM passive interfaces do not send or process any received PIM control packets.

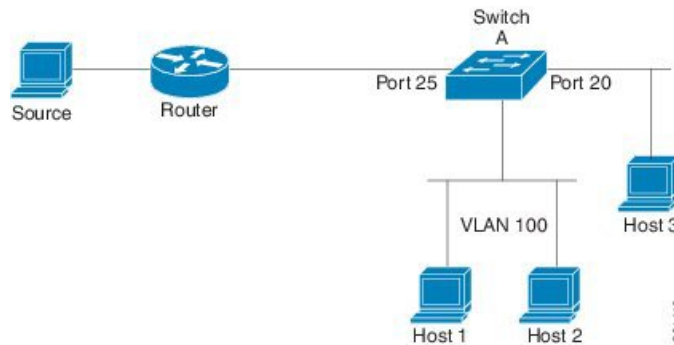
When using PIM stub routing, you should configure the distribution and remote routers to use IPv6 multicast routing and configure only the switch as a PIM stub router. The switch does not route transit traffic between distribution routers. You also need to configure a routed uplink port on the switch. The switch uplink port cannot be used with SVIs.

You must also configure EIGRP stub routing when configuring PIM stub routing on the switch.

The redundant PIM stub router topology is not supported. The redundant topology exists when there is more than one PIM router forwarding multicast traffic to a single access domain. PIM messages are blocked, and the PIM assert and designated router election mechanisms are not supported on the PIM passive interfaces. Only the non-redundant access router topology is supported by the PIM stub feature. By using a non-redundant topology, the PIM passive interface assumes that it is the only interface and designated router on that access domain.

In the figure shown below, Switch A routed uplink port 25 is connected to the router and PIM stub routing is enabled on the VLAN 100 interfaces and on Host 3. This configuration allows the directly connected hosts to receive traffic from multicast source.

Figure 22: PIM Stub Router Configuration



Rendezvous Point

IPv6 PIM provides embedded RP support. Embedded RP support allows the device to learn RP information using the multicast group destination address instead of the statically configured RP. For devices that are the RP, the device must be statically configured as the RP.

The device searches for embedded RP group addresses in MLD reports or PIM messages and data packets. On finding such an address, the device learns the RP for the group from the address itself. It then uses this learned RP for all protocol activity for the group. For devices that are the RP, the device is advertised as an embedded RP must be configured as the RP.

To select a static RP over an embedded RP, the specific embedded RP group range or mask must be configured in the access list of the static RP. When PIM is configured in sparse mode, you must also choose one or more devices to operate as an RP. An RP is a single common root placed at a chosen point of a shared distribution tree and is configured statically in each box.

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

- Data is encapsulated in register packets and unicast directly to the RP by the first-hop device operating as the DR.
- If the RP has itself joined the source tree, it is multicast-forwarded per the RPF forwarding algorithm described in the PIM-Sparse Mode section.

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

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

IPv6 multicast supports the PIM accept register feature, which is the ability to perform PIM-SM register message filtering at the RP. The user can match an access list or compare the AS path for the registered source with the AS path specified in a route map.

Static Mroutes

IPv6 static mroutes behave much in the same way as IPv4 static mroutes used to influence the RPF check. IPv6 static mroutes share the same database as IPv6 static routes and are implemented by extending static route support for RPF checks. Static mroutes support equal-cost multipath mroutes, and they also support unicast-only static routes.

MRIB

The Multicast Routing Information Base (MRIB) is a protocol-independent repository of multicast routing entries instantiated by multicast routing protocols (routing clients). Its main function is to provide independence between routing protocols and the Multicast Forwarding Information Base (MFIB). It also acts as a coordination and communication point among its clients.

Routing clients use the services provided by the MRIB to instantiate routing entries and retrieve changes made to routing entries by other clients. Besides routing clients, MRIB also has forwarding clients (MFIB instances) and special clients such as MLD. MFIB retrieves its forwarding entries from MRIB and notifies the MRIB of any events related to packet reception. These notifications can either be explicitly requested by routing clients or spontaneously generated by the MFIB.

Another important function of the MRIB is to allow for the coordination of multiple routing clients in establishing multicast connectivity within the same multicast session. MRIB also allows for the coordination between MLD and routing protocols.

MFIB

The MFIB is a platform-independent and routing-protocol-independent library for IPv6 software. Its main purpose is to provide a Cisco IOS platform with an interface with which to read the IPv6 multicast forwarding table and notifications when the forwarding table changes. The information provided by the MFIB has clearly defined forwarding semantics and is designed to make it easy for the platform to translate to its specific hardware or software forwarding mechanisms.

When routing or topology changes occur in the network, the IPv6 routing table is updated, and those changes are reflected in the MFIB. The MFIB maintains next-hop address information based on the information in the IPv6 routing table. Because there is a one-to-one correlation between MFIB entries and routing table entries, the MFIB contains all known routes and eliminates the need for route cache maintenance that is associated with switching paths such as fast switching and optimum switching.

MFIB



Note Distributed MFIB has its significance only in a stacked environment where the active switch distributes the MFIB information to the other member switches in the stack. In the following section the line cards are nothing but the member switches in the stack.

MFIB (MFIB) is used to switch multicast IPv6 packets on distributed platforms. MFIB may also contain platform-specific information on replication across line cards. The basic MFIB routines that implement the core of the forwarding logic are common to all forwarding environments.

MFIB implements the following functions:

- Relays data-driven protocol events generated in the line cards to PIM.
- Provides an MFIB platform application program interface (API) to propagate MFIB changes to platform-specific code responsible for programming the hardware acceleration engine. This API also includes entry points to switch a packet in software (necessary if the packet is triggering a data-driven event) and to upload traffic statistics to the software.

The combination of MFIB and MRIB subsystems also allows the switch to have a "customized" copy of the MFIB database in each line card and to transport MFIB-related platform-specific information from the RP to the line cards.

IPv6 Multicast Process Switching and Fast Switching

A unified MFIB is used to provide both fast switching and process switching support for PIM-SM and PIM-SSM in IPv6 multicast. In process switching, the must examine, rewrite, and forward each packet. The packet is first received and copied into the system memory. The switch then looks up the Layer 3 network address in the routing table. The Layer 2 frame is then rewritten with the next-hop destination address and sent to the outgoing interface. The also computes the cyclic redundancy check (CRC). This switching method is the least scalable method for switching IPv6 packets.

IPv6 multicast fast switching allows switches to provide better packet forwarding performance than process switching. Information conventionally stored in a route cache is stored in several data structures for IPv6 multicast switching. The data structures provide optimized lookup for efficient packet forwarding.

In IPv6 multicast forwarding, the first packet is fast-switched if the PIM protocol logic allows it. In IPv6 multicast fast switching, the MAC encapsulation header is precomputed. IPv6 multicast fast switching uses the MFIB to make IPv6 destination prefix-based switching decisions. In addition to the MFIB, IPv6 multicast fast switching uses adjacency tables to prepend Layer 2 addressing information. The adjacency table maintains Layer 2 next-hop addresses for all MFIB entries.

The adjacency table is populated as adjacencies are discovered. Each time an adjacency entry is created (such as through ARP), a link-layer header for that adjacent node is precomputed and stored in the adjacency table. Once a route is determined, it points to a next hop and corresponding adjacency entry. It is subsequently used for encapsulation during switching of packets.

A route might have several paths to a destination prefix, such as when a switch is configured for simultaneous load balancing and redundancy. For each resolved path, a pointer is added for the adjacency corresponding to the next-hop interface for that path. This mechanism is used for load balancing across several paths.

Multiprotocol BGP for the IPv6 Multicast Address Family

The multiprotocol BGP for the IPv6 multicast address family feature provides multicast BGP extensions for IPv6 and supports the same features and functionality as IPv4 BGP. IPv6 enhancements to multicast BGP include support for an IPv6 multicast address family and network layer reachability information (NLRI) and next hop (the next switch in the path to the destination) attributes that use IPv6 addresses.

Multicast BGP is an enhanced BGP that allows the deployment of interdomain IPv6 multicast. Multiprotocol BGP carries routing information for multiple network layer protocol address families; for example, IPv6 address family and for IPv6 multicast routes. The IPv6 multicast address family contains routes used for RPF lookup by the IPv6 PIM protocol, and multicast BGP IPv6 provides for interdomain transport of the same. Users must use multiprotocol BGP for IPv6 multicast when using IPv6 multicast with BGP because the unicast BGP learned routes will not be used for IPv6 multicast.

Multicast BGP functionality is provided through a separate address family context. A subsequent address family identifier (SAFI) provides information about the type of the network layer reachability information that is carried in the attribute. Multiprotocol BGP unicast uses SAFI 1 messages, and multiprotocol BGP multicast uses SAFI 2 messages. SAFI 1 messages indicate that the routes are only usable for IP unicast, but not IP multicast. Because of this functionality, BGP routes in the IPv6 unicast RIB must be ignored in the IPv6 multicast RPF lookup.

A separate BGP routing table is maintained to configure incongruent policies and topologies (forexample, IPv6 unicast and multicast) by using IPv6 multicast RPF lookup. Multicast RPF lookup is very similar to the IP unicast route lookup.

No MRIB is associated with the IPv6 multicast BGP table. However, IPv6 multicast BGP operates on the unicast IPv6 RIB when needed. Multicast BGP does not insert or update routes into the IPv6 unicast RIB.

Implementing IPv6 Multicast

Enabling IPv6 Multicast Routing

To enable IPv6 multicast routing, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enter global configuration mode.
Step 3	ipv6 multicast-routing Example: Device(config)# ipv6 multicast-routing	Enables multicast routing on all IPv6-enabled interfaces and enables multicast forwarding for PIM and MLD on all enabled interfaces of the switch.
Step 4	copy running-config startup-config Example: Device(config)# copy running-config startup-config	(Optional) Save your entries in the configuration file.

Customizing and Verifying the MLD Protocol

Customizing and Verifying MLD on an Interface

To customize and verify MLD on an interface, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. 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 GigabitEthernet 1/0/1	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 4	ipv6 mld join-group [<i>group-address</i>] [include exclude] { <i>source-address</i> source-list [<i>acl</i>]} Example: Device(config-if)# ipv6 mld join-group FF04::10	Configures MLD reporting for a specified group and source.
Step 5	ipv6 mld access-group <i>access-list-name</i> Example: Device(config-if)# ipv6 access-list acc-grp-1	Allows the user to perform IPv6 multicast receiver access control.
Step 6	ipv6 mld static-group [<i>group-address</i>] [include exclude] { <i>source-address</i> source-list [<i>acl</i>]} Example: Device(config-if)# ipv6 mld static-group ff04::10 include 100::1	Statically forwards traffic for the multicast group onto a specified interface and cause the interface to behave as if a MLD joiner were present on the interface.
Step 7	ipv6 mld query-max-response-time <i>seconds</i> Example:	Configures the timeout value before the switch takes over as the querier for the interface.

	Command or Action	Purpose
	Device(config-if)# ipv6 mld query-timeout 130	
Step 8	exit Example: Device(config-if)# exit	Enter this command twice to exit interface configuration mode and enter privileged EXEC mode.
Step 9	show ipv6 mld groups [link-local] [group-name group-address] [interface-type interface-number] [detail explicit] Example: Device# show ipv6 mld groups GigabitEthernet 1/0/1	Displays the multicast groups that are directly connected to the switch and that were learned through MLD.
Step 10	show ipv6 mld groups summary Example: Device# show ipv6 mld groups summary	Displays the number of (*, G) and (S, G) membership reports present in the MLD cache.
Step 11	show ipv6 mld interface [type number] Example: Device# show ipv6 mld interface GigabitEthernet 1/0/1	Displays multicast-related information about an interface.
Step 12	debug ipv6 mld [group-name group-address interface-type] Example: Device# debug ipv6 mld	Enables debugging on MLD protocol activity.
Step 13	debug ipv6 mld explicit [group-name group-address] Example: Device# debug ipv6 mld explicit	Displays information related to the explicit tracking of hosts.
Step 14	copy running-config startup-config	(Optional) Save your entries in the configuration file.

Implementing MLD Group Limits

Per-interface and global MLD limits operate independently of each other. Both per-interface and global MLD limits can be configured on the same switch. The number of MLD limits, globally or per interface, is not configured by default; the limits must be configured by the user. A membership report that exceeds either the per-interface or the global state limit is ignored.

Implementing MLD Group Limits Globally

To implement MLD group limits globally, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ipv6 mld [vrf vrf-name] state-limit number Example: Device(config)# ipv6 mld state-limit 300	Limits the number of MLD states globally.
Step 4	copy running-config startup-config	(Optional) Save your entries in the configuration file.

Implementing MLD Group Limits per Interface

To implement MLD group limits per interface, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface type number Example: Device(config)# interface GigabitEthernet 1/0/1	Specifies an interface type and number, and places the switch in interface configuration mode.

	Command or Action	Purpose
Step 4	ipv6 mld limit <i>number</i> [except] <i>access-list</i> Example: <pre>Device(config-if)# ipv6 mld limit 100</pre>	Limits the number of MLD states on a per-interface basis.
Step 5	copy running-config startup-config	(Optional) Save your entries in the configuration file.

Configuring Explicit Tracking of Receivers to Track Host Behavior

The explicit tracking feature allows a switch to track the behavior of the hosts within its IPv6 network and enables the fast leave mechanism to be used with MLD version 2 host reports.

To configuring explicit tracking of receivers to track host behavior, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enter global configuration mode.
Step 3	interface <i>type number</i> Example: <pre>Device(config)# interface GigabitEthernet 1/0/1</pre>	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 4	ipv6 mld explicit-tracking <i>access-list-name</i> Example: <pre>Device(config-if)# ipv6 mld explicit-tracking list1</pre>	Enables explicit tracking of hosts.
Step 5	copy running-config startup-config	(Optional) Save your entries in the configuration file.

Resetting the MLD Traffic Counters

To reset the MLD traffic counters, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	clear ipv6 mld traffic Example: Device# clear ipv6 mld traffic	Resets all MLD traffic counters.
Step 4	show ipv6 mld traffic Example: Device# show ipv6 mld traffic	Displays the MLD traffic counters.
Step 5	copy running-config startup-config	(Optional) Save your entries in the configuration file.

Clearing the MLD Interface Counters

To clearing the MLD interface counters, perform this procedure

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	clear ipv6 mld counters <i>interface-type</i> Example:	Clears the MLD interface counters.

	Command or Action	Purpose
	Device# clear ipv6 mld counters Ethernet1/0	
Step 4	copy running-config startup-config	(Optional) Save your entries in the configuration file.

Configuring PIM

This section explains how to configure PIM.

Configuring PIM-SM and Displaying PIM-SM Information for a Group Range

To configuring PIM-SM and view PIM-SM information for a group range, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ipv6 pim rp-address <i>ipv6-address[group-access-list]</i> Example: Device(config)# ipv6 pim rp-address 2001:DB8::01:800:200E:8C6C acc-grp-1	Configures the address of a PIM RP for a particular group range.
Step 4	exit Example: Device(config)# exit	Exits global configuration mode, and returns the switch to privileged EXEC mode.
Step 5	show ipv6 pim interface [state-on] [state-off] [type-number] Example: Device# show ipv6 pim interface	Displays information about interfaces configured for PIM.

	Command or Action	Purpose
Step 6	show ipv6 pim group-map [<i>group-name</i> <i>group-address</i>] [<i>group-range</i> <i>group-mask</i>] [<i>info-source</i> { <i>bsr</i> default embedded-rp static }] Example: Device# show ipv6 pim group-map	Displays an IPv6 multicast group mapping table.
Step 7	show ipv6 pim neighbor [<i>detail</i>] [<i>interface-type interface-number</i> count] Example: Device# show ipv6 pim neighbor	Displays the PIM neighbors discovered by the Cisco IOS software.
Step 8	show ipv6 pim range-list [<i>config</i>] [<i>rp-address</i> <i>rp-name</i>] Example: Device# show ipv6 pim range-list	Displays information about IPv6 multicast range lists.
Step 9	show ipv6 pim tunnel [<i>interface-type interface-number</i>] Example: Device# show ipv6 pim tunnel	Displays information about the PIM register encapsulation and de-encapsulation tunnels on an interface.
Step 10	debug ipv6 pim [<i>group-name</i> <i>group-address</i> interface <i>interface-type</i> bsr group mvpn neighbor] Example: Device# debug ipv6 pim	Enables debugging on PIM protocol activity.
Step 11	copy running-config startup-config	(Optional) Save your entries in the configuration file.

Configuring PIM Options

To configure PIM options, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ipv6 pim spt-threshold infinity [group-list access-list-name] Example: <pre>Device(config)# ipv6 pim spt-threshold infinity group-list acc-grp-1</pre>	Configures when a PIM leaf switch joins the SPT for the specified groups.
Step 4	ipv6 pim accept-register {list access-list route-map map-name} Example: <pre>Device(config)# ipv6 pim accept-register route-map reg-filter</pre>	Accepts or rejects registers at the RP.
Step 5	interface type number Example: <pre>Device(config)# interface GigabitEthernet 1/0/1</pre>	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 6	ipv6 pim dr-priority value Example: <pre>Device(config-if)# ipv6 pim dr-priority 3</pre>	Configures the DR priority on a PIM switch.
Step 7	ipv6 pim hello-interval seconds Example: <pre>Device(config-if)# ipv6 pim hello-interval 45</pre>	Configures the frequency of PIM hello messages on an interface.
Step 8	ipv6 pim join-prune-interval seconds Example: <pre>Device(config-if)# ipv6 pim join-prune-interval 75</pre>	Configures periodic join and prune announcement intervals for a specified interface.
Step 9	exit Example: <pre>Device(config-if)# exit</pre>	Enter this command twice to exit interface configuration mode and enter privileged EXEC mode.

	Command or Action	Purpose
Step 10	ipv6 pim join-prune statistic [<i>interface-type</i>] Example: <pre>Device(config-if)# show ipv6 pim join-prune statistic</pre>	Displays the average join-prune aggregation for the most recently aggregated packets for each interface.
Step 11	copy running-config startup-config	(Optional) Save your entries in the configuration file.

Resetting the PIM Traffic Counters

If PIM malfunctions or in order to verify that the expected number of PIM packets are received and sent, the user can clear PIM traffic counters. Once the traffic counters are cleared, the user can enter the `show ipv6 pim` command to verify that PIM is functioning correctly and that PIM packets are being received and sent correctly.

To resetting the PIM traffic counters, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	clear ipv6 pim traffic Example: <pre>Device# clear ipv6 pim traffic</pre>	Resets the PIM traffic counters.
Step 4	show ipv6 pim traffic Example: <pre>Device# show ipv6 pim traffic</pre>	Displays the PIM traffic counters.
Step 5	copy running-config startup-config	(Optional) Save your entries in the configuration file.

Clearing the PIM Topology Table to Reset the MRIB Connection

No configuration is necessary to use the MRIB. However, users may in certain situations want to clear the PIM topology table in order to reset the MRIB connection and verify MRIB information.

To clear the PIM topology table to reset the MRIB connection, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	clear ipv6 pim topology [<i>group-name</i> <i>group-address</i>] Example: Device# clear ipv6 pim topology FF04::10	Clears the PIM topology table.
Step 4	show ipv6 mrib client [<i>filter</i>] [<i>name {client-name client-name : client-id}</i>] Example: Device# show ipv6 mrib client	Displays multicast-related information about an interface.
Step 5	show ipv6 mrib route { <i>link-local</i> <i>summary</i> [<i>sourceaddress-or-name</i> *] [<i>groupname-or-address</i> [<i>prefix-length</i>]]] Example: Device# show ipv6 mrib route	Displays the MRIB route information.
Step 6	show ipv6 pim topology [<i>groupname-or-address</i> [<i>sourceaddress-or-name</i>] <i>link-local</i> <i>route-count</i> [<i>detail</i>]] Example: Device# show ipv6 pim topology	Displays PIM topology table information for a specific group or all groups.

	Command or Action	Purpose
Step 7	debug ipv6 mrib client Example: Device# <code>debug ipv6 mrib client</code>	Enables debugging on MRIB client management activity.
Step 8	debug ipv6 mrib io Example: Device# <code>debug ipv6 mrib io</code>	Enables debugging on MRIB I/O events.
Step 9	debug ipv6 mrib proxy Example: Device# <code>debug ipv6 mrib proxy</code>	Enables debugging on MRIB proxy activity between the switch processor and line cards on distributed switch platforms.
Step 10	debug ipv6 mrib route [<i>group-name</i> <i>group-address</i>] Example: Device# <code>debug ipv6 mrib route</code>	Displays information about MRIB routing entry-related activity.
Step 11	debug ipv6 mrib table Example: Device# <code>debug ipv6 mrib table</code>	Enables debugging on MRIB table management activity.
Step 12	copy running-config startup-config	(Optional) Save your entries in the configuration file.

Configuring PIM IPv6 Stub Routing

The PIM Stub routing feature supports multicast routing between the distribution layer and the access layer. It supports two types of PIM interfaces, uplink PIM interfaces, and PIM passive interfaces. A routed interface configured with the PIM passive mode does not pass or forward PIM control traffic, it only passes and forwards MLD traffic.

PIM IPv6 Stub Routing Configuration Guidelines

- Before configuring PIM stub routing, you must have IPv6 multicast routing configured on both the stub router and the central router. You must also have PIM mode (sparse-mode) configured on the uplink interface of the stub router.
- The PIM stub router does not route the transit traffic between the distribution routers. Unicast (EIGRP) stub routing enforces this behavior. You must configure unicast stub routing to assist the PIM stub router behavior. For more information, see the *EIGRP Stub Routing* section.
- Only directly connected multicast (MLD) receivers and sources are allowed in the Layer 2 access domains. The PIM protocol is not supported in access domains.
- The redundant PIM stub router topology is not supported.

Default IPv6 PIM Routing Configuration

This table displays the default IPv6 PIM routing configuration for the Device.

Table 20: Default Multicast Routing Configuration

Feature	Default Setting
Multicast routing	Disabled on all interfaces.
PIM version	Version 2.
PIM mode	No mode is defined.
PIM stub routing	None configured.
PIM RP address	None configured.
PIM domain border	Disabled.
PIM multicast boundary	None.
Candidate BSRs	Disabled.
Candidate RPs	Disabled.
Shortest-path tree threshold rate	0 kb/s.
PIM router query message interval	30 seconds.

Enabling IPV6 PIM Stub Routing

To enable IPV6 PIM stub routing, perform this procedure:

Before you begin

PIM stub routing is disabled in IPv6 by default.

Procedure

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

	Command or Action	Purpose
Step 3	ipv6 multicast pim-passive-enable Example: <pre>Device(config-if) # ipv6 multicast pim-passive-enable</pre>	Enables IPv6 Multicast PIM routing on the switch.
Step 4	interface interface-id Example: <pre>Device(config) # interface gigabitethernet 9/0/6</pre>	<p>Specifies the interface on which you want to enable PIM stub routing, and enters interface configuration mode.</p> <p>The specified interface must be one of the following:</p> <ul style="list-style-type: none"> • A routed port—A physical port that has been configured as a Layer 3 port by entering the no switchport interface configuration command. You will also need to enable IP PIM sparse mode on the interface, and join the interface as a statically connected member to an MLD static group. • An SVI—A VLAN interface created by using the interface vlan vlan-id global configuration command. You will also need to enable IP PIM sparse mode on the VLAN, join the VLAN as a statically connected member to an MLD static group, and then enable MLD snooping on the VLAN, the MLD static group, and physical interface. <p>These interfaces must have IPv6 addresses assigned to them.</p>
Step 5	ipv6 pim Example: <pre>Device(config-if) # ipv6 pim</pre>	Enables the PIM on the interface.
Step 6	ipv6 pim {bsr} {dr-priority value} {hello-interval seconds} {join-prune-interval seconds} {passive} Example: <pre>Device(config-if) # ipv6 pim bsr dr-priority hello-interval join-prune-interval passive</pre>	<p>Configures the various PIM stub features on the interface.</p> <p>Enter bsr to configure BSR on a PIM switch</p> <p>Enter dr-priority to configure the DR priority on a PIM switch.</p> <p>Enter hello-interval to configure the frequency of PIM hello messages on an interface.</p>

	Command or Action	Purpose
		Enter join-prune-interval to configure periodic join and prune announcement intervals for a specified interface. Enter passive to configure the PIM in the passive mode.
Step 7	end Example: Device(config-if) # end	Returns to privileged EXEC mode.

Monitoring IPv6 PIM Stub Routing

Table 21: PIM Stub Configuration show Commands

Command	Purpose
show ipv6 pim interface	Displays the PIM stub that is enabled on each interface.
show ipv6 mld groups	Displays the interested clients that have joined the specific multicast source group.
show ipv6 mroute	Verifies that the multicast stream forwards from the source to the interested clients.

Configuring a BSR

The tasks included here are described below.

Configuring a BSR and Verifying BSR Information

To configure and verify BSR Information, perform this procedure:

Procedure

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

	Command or Action	Purpose
Step 3	ipv6 pim bsr candidate bsr <i>ipv6-address[hash-mask-length] [priority priority-value]</i> Example: Device(config)# ipv6 pim bsr candidate bsr 2001:DB8:3000:3000::42 124 priority 10	Configures a switch to be a candidate BSR.
Step 4	interface type number Example: Device(config)# interface GigabitEthernet 1/0/1	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 5	ipv6 pim bsr border Example: Device(config-if)# ipv6 pim bsr border	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 6	exit Example: Device(config-if)# exit	Enter this command twice to exit interface configuration mode and enter privileged EXEC mode.
Step 7	show ipv6 pim bsr {election rp-cache candidate-rp} Example: Device(config-if)# show ipv6 pim bsr election	Displays information related to PIM BSR protocol processing.
Step 8	copy running-config startup-config	(Optional) Save your entries in the configuration file.

Sending PIM RP Advertisements to the BSR

To sending PIM RP advertisements to the BSR, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ipv6 pim bsr candidate rp <i>ipv6-address</i> [group-list <i>access-list-name</i>] [priority <i>priority-value</i>] [interval seconds] Example: Device(config)# ipv6 pim bsr candidate rp 2001:DB8:3000:3000::42 priority 0	Sends PIM RP advertisements to the BSR.
Step 4	interface <i>type number</i> Example: Device(config)# interface GigabitEthernet 1/0/1	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 5	ipv6 pim bsr border Example: Device(config-if)# ipv6 pim bsr border	Configures a border for all BSMs of any scope on a specified interface.
Step 6	copy running-config startup-config	(Optional) Save your entries in the configuration file.

Configuring BSR for Use Within Scoped Zones

To configure BSR for use within scoped zones, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ipv6 pim bsr candidate rp <i>ipv6-address</i> [<i>hash-mask-length</i>] [priority <i>priority-value</i>]	Configures a switch to be a candidate BSR.

	Command or Action	Purpose
	Example: <pre>Device(config)# ipv6 pim bsr candidate bsr 2001:DB8:1:1:4</pre>	
Step 4	ipv6 pim bsr candidate rp <i>ipv6-address</i> [group-list <i>access-list-name</i>] [priority <i>priority-value</i>] [interval seconds] Example: <pre>Device(config)# ipv6 pim bsr candidate rp 2001:DB8:1:1:1 group-list list scope 6</pre>	Configures the candidate RP to send PIM RP advertisements to the BSR.
Step 5	interface <i>type number</i> Example: <pre>Device(config-if)# interface GigabitEthernet 1/0/1</pre>	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 6	ipv6 multicast boundary scope <i>scope-value</i> Example: <pre>Device(config-if)# ipv6 multicast boundary scope 6</pre>	Configures a multicast boundary on the interface for a specified scope.
Step 7	copy running-config startup-config	(Optional) Save your entries in the configuration file.

Configuring BSR Switches to Announce Scope-to-RP Mappings

IPv6 BSR switches can be statically configured to announce scope-to-RP mappings directly instead of learning them from candidate-RP messages. A user might want to configure a BSR switch to announce scope-to-RP mappings so that an RP that does not support BSR is imported into the BSR. Enabling this feature also allows an RP positioned outside the enterprise's BSR domain to be learned by the known remote RP on the local candidate BSR switch.

To configure BSR switches to announce Scope-to-RP mappings, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example:	Enters global configuration mode.

	Command or Action	Purpose
	Device# <code>configure terminal</code>	
Step 3	ipv6 pim bsr announced rp <i>ipv6-address</i> [group-list <i>access-list-name</i>] [priority <i>priority-value</i>] Example: Device(config)# <code>ipv6 pim bsr announced rp 2001:DB8:3000:3000::42 priority 0</code>	Announces scope-to-RP mappings directly from the BSR for the specified candidate RP.
Step 4	copy running-config startup-config	(Optional) Save your entries in the configuration file.

Configuring SSM Mapping

When the SSM mapping feature is enabled, DNS-based SSM mapping is automatically enabled, which means that the switch will look up the source of a multicast MLD version 1 report from a DNS server.

You can use either DNS-based or static SSM mapping, depending on your switch configuration. If you choose to use static SSM mapping, you can configure multiple static SSM mappings. If multiple static SSM mappings are configured, the source addresses of all matching access lists will be used.



Note To use DNS-based SSM mapping, the switch needs to find at least one correctly configured DNS server, to which the switch may be directly attached.

To configuring SSM mapping, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> <code>enable</code>	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 3	ipv6 mld ssm-map enable Example:	Enables the SSM mapping feature for groups in the configured SSM range.

	Command or Action	Purpose
	<code>Device(config)# ipv6 mld ssm-map enable</code>	
Step 4	no ipv6 mld ssm-map query dns Example: <code>Device(config)# no ipv6 mld ssm-map query dns</code>	Disables DNS-based SSM mapping.
Step 5	ipv6 mld ssm-map static <i>access-list source-address</i> Example: <code>Device(config-if)# ipv6 mld ssm-map static SSM_MAP_ACL_2 2001:DB8:1::1</code>	Configures static SSM mappings.
Step 6	exit Example: <code>Device(config-if)# exit</code>	Exits global configuration mode, and returns the switch to privileged EXEC mode.
Step 7	show ipv6 mld ssm-map [<i>source-address</i>] Example: <code>Device(config-if)# show ipv6 mld ssm-map</code>	Displays SSM mapping information.
Step 8	copy running-config startup-config	(Optional) Save your entries in the configuration file.

Configuring Static Mroutes

Static multicast routes (mroutes) in IPv6 can be implemented as an extension of IPv6 static routes. You can configure your switch to use a static route for unicast routing only, to use a static multicast route for multicast RPF selection only, or to use a static route for both unicast routing and multicast RPF selection.

To configure static mroutes, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: <code>Device> enable</code>	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example:	Enters global configuration mode.

	Command or Action	Purpose
	Device# configure terminal	
Step 3	ipv6 route { <i>ipv6-prefix / prefix-length</i> <i>ipv6-address interface-type interface-number</i> <i>ipv6-address</i> } [<i>administrative-distance</i>] [<i>administrative-multicast-distance unicast multicast</i>] [tag tag] Example: Device(config)# ipv6 route 2001:DB8::/64 6::6 100	Establishes static IPv6 routes. The example shows a static route used for both unicast routing and multicast RPF selection.
Step 4	exit Example: Device# exit	Exits global configuration mode, and returns the switch to privileged EXEC mode.
Step 5	show ipv6 mroute [<i>link-local [group-name group-address [source-address source-name]] [summary] [count]</i>] Example: Device# show ipv6 mroute ff07::1	Displays the contents of the IPv6 multicast routing table.
Step 6	show ipv6 mroute [<i>link-local group-name group-address</i>] active [<i>kbps</i>] Example: Device(config-if)# show ipv6 mroute active	Displays the active multicast streams on the switch.
Step 7	show ipv6 rpf [<i>ipv6-prefix</i>] Example: Device(config-if)# show ipv6 rpf 2001::1:1:2	Checks RPF information for a given unicast host address and prefix.
Step 8	copy running-config startup-config	(Optional) Save your entries in the configuration file.

Using MFIB in IPv6 Multicast

Multicast forwarding is automatically enabled when IPv6 multicast routing is enabled.

Verifying MFIB Operation in IPv6 Multicast

To verify MFIB operation in IPv6 multicast

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	show ipv6 mfib [verbose <i>group-address-name</i> <i>ipv6-prefix/prefix-length</i> <i>source-address-name</i> count interface status summary] Example: Device# show ipv6 mfib	Displays the forwarding entries and interfaces in the IPv6 MFIB.
Step 3	show ipv6 mfib [all linkscope group-name group-address [source-name source-address]] count Example: Device# show ipv6 mfib ff07::1	Displays the contents of the IPv6 multicast routing table.
Step 4	show ipv6 mfib interface Example: Device# show ipv6 mfib interface	Displays information about IPv6 multicast-enabled interfaces and their forwarding status.
Step 5	show ipv6 mfib status Example: Device# show ipv6 mfib status	Displays general MFIB configuration and operational status.
Step 6	show ipv6 mfib summary Example: Device# show ipv6 mfib summary	Displays summary information about the number of IPv6 MFIB entries and interfaces.
Step 7	debug ipv6 mfib [<i>group-name</i> <i>group-address</i>] [adjacency db fs init interface mrrib [detail] nat pak platform ppr ps signal table] Example: Device# debug ipv6 mfib FF04::10 pak	Enables debugging output on the IPv6 MFIB.

Resetting MFIB Traffic Counters

To reset MFIB traffic counters, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	clear ipv6 mfib counters [<i>group-name</i> <i>group-address</i> [<i>source-address</i> <i>source-name</i>]] Example: Device# clear ipv6 mfib counters FF04::10	Resets all active MFIB traffic counters.

Additional References

Standards and RFCs

Standard/RFC	Title
RFC 4292	<i>IP Forwarding Table</i>
RFC 4293	<i>Management Information Base for the Internet Protocol (IP)</i>

Feature History for IPv6 Multicast

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	IPv6 multicast	IPv6 multicast allows a host to send a single data stream to a subset of all hosts (group transmission) simultaneously

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>.



CHAPTER 13

Configuring MLD Snooping

This module contains details of configuring MLD snooping

- [Information About Configuring IPv6 MLD Snooping, on page 273](#)
- [How to Configure IPv6 MLD Snooping, on page 276](#)
- [Displaying MLD Snooping Information, on page 284](#)
- [Configuration Examples for Configuring MLD Snooping, on page 285](#)
- [Additional References, on page 286](#)
- [Feature History for MLD Snooping, on page 287](#)

Information About Configuring IPv6 MLD Snooping

You can use Multicast Listener Discovery (MLD) snooping to enable efficient distribution of IP Version 6 (IPv6) multicast data to clients and routers in a switched network on the switch. Unless otherwise noted, the term switch refers to a standalone switch and to a switch stack.

To use IPv6, you must configure the dual IPv4 and IPv6 Switch Database Management (SDM) template on the switch.

Understanding MLD Snooping

In IP Version 4 (IPv4), Layer 2 switches can use Internet Group Management Protocol (IGMP) snooping to limit the flooding of multicast traffic by dynamically configuring Layer 2 interfaces so that multicast traffic is forwarded to only those interfaces associated with IP multicast devices. In IPv6, MLD snooping performs a similar function. With MLD snooping, IPv6 multicast data is selectively forwarded to a list of ports that want to receive the data, instead of being flooded to all ports in a VLAN. This list is constructed by snooping IPv6 multicast control packets.

MLD is a protocol used by IPv6 multicast routers to discover the presence of multicast listeners (nodes wishing to receive IPv6 multicast packets) on the links that are directly attached to the routers and to discover which multicast packets are of interest to neighboring nodes. MLD is derived from IGMP; MLD Version 1 (MLDv1) is equivalent to IGMPv2, and MLD Version 2 (MLDv2) is equivalent to IGMPv3. MLD is a subprotocol of Internet Control Message Protocol Version 6 (ICMPv6), and MLD messages are a subset of ICMPv6 messages, identified in IPv6 packets by a preceding Next Header value of 58.

The switch supports two versions of MLD snooping:

- MLDv1 snooping detects MLDv1 control packets and sets up traffic bridging based on IPv6 destination multicast addresses.
- MLDv2 basic snooping (MBSS) uses MLDv2 control packets to set up traffic forwarding based on IPv6 destination multicast addresses.

The switch can snoop on both MLDv1 and MLDv2 protocol packets and bridge IPv6 multicast data based on destination IPv6 multicast addresses.



Note The switch does not support MLDv2 enhanced snooping, which sets up IPv6 source and destination multicast address-based forwarding.

MLD snooping can be enabled or disabled globally or per VLAN. When MLD snooping is enabled, a per-VLAN IPv6 multicast address table is constructed in software and hardware. The switch then performs IPv6 multicast-address based bridging in hardware.

MLD Messages

MLDv1 supports three types of messages:

- Listener Queries are the equivalent of IGMPv2 queries and are either General Queries or Multicast-Address-Specific Queries (MASQs).
- Multicast Listener Reports are the equivalent of IGMPv2 reports
- Multicast Listener Done messages are the equivalent of IGMPv2 leave messages.

MLDv2 supports MLDv2 queries and reports, as well as MLDv1 Report and Done messages.

Message timers and state transitions resulting from messages being sent or received are the same as those of IGMPv2 messages. MLD messages that do not have valid link-local IPv6 source addresses are ignored by MLD routers and switches.

MLD Queries

The switch sends out MLD queries, constructs an IPv6 multicast address database, and generates MLD group-specific and MLD group-and-source-specific queries in response to MLD Done messages. The switch also supports report suppression, report proxying, Immediate-Leave functionality, and static IPv6 multicast group address configuration.

When MLD snooping is disabled, all MLD queries are flooded in the ingress VLAN.

When MLD snooping is enabled, received MLD queries are flooded in the ingress VLAN, and a copy of the query is sent to the CPU for processing. From the received query, MLD snooping builds the IPv6 multicast address database. It detects multicast router ports, maintains timers, sets report response time, learns the querier IP source address for the VLAN, learns the querier port in the VLAN, and maintains multicast-address aging.

When a group exists in the MLD snooping database, the switch responds to a group-specific query by sending an MLDv1 report. When the group is unknown, the group-specific query is flooded to the ingress VLAN.

When a host wants to leave a multicast group, it can send out an MLD Done message (equivalent to IGMP Leave message). When the switch receives an MLDv1 Done message, if Immediate-Leave is not enabled, the switch sends an MASQ to the port from which the message was received to determine if other devices connected to the port should remain in the multicast group.

Multicast Client Aging Robustness

You can configure port membership removal from addresses based on the number of queries. A port is removed from membership to an address only when there are no reports to the address on the port for the configured number of queries. The default number is 2.

Multicast Router Discovery

Like IGMP snooping, MLD snooping performs multicast router discovery, with these characteristics:

- Ports configured by a user never age out.
- Dynamic port learning results from MLDv1 snooping queries and IPv6 PIMv2 packets.
- If there are multiple routers on the same Layer 2 interface, MLD snooping tracks a single multicast router on the port (the router that most recently sent a router control packet).
- Dynamic multicast router port aging is based on a default timer of 5 minutes; the multicast router is deleted from the router port list if no control packet is received on the port for 5 minutes.
- IPv6 multicast router discovery only takes place when MLD snooping is enabled on the switch.
- Received IPv6 multicast router control packets are always flooded to the ingress VLAN, whether or not MLD snooping is enabled on the switch.
- After the discovery of the first IPv6 multicast router port, unknown IPv6 multicast data is forwarded only to the discovered router ports (before that time, all IPv6 multicast data is flooded to the ingress VLAN).

MLD Reports

The processing of MLDv1 join messages is essentially the same as with IGMPv2. When no IPv6 multicast routers are detected in a VLAN, reports are not processed or forwarded from the switch. When IPv6 multicast routers are detected and an MLDv1 report is received, an IPv6 multicast group address is entered in the VLAN MLD database. Then all IPv6 multicast traffic to the group within the VLAN is forwarded using this address. When MLD snooping is disabled, reports are flooded in the ingress VLAN.

When MLD snooping is enabled, MLD report suppression, called listener message suppression, is automatically enabled. With report suppression, the switch forwards the first MLDv1 report received by a group to IPv6 multicast routers; subsequent reports for the group are not sent to the routers. When MLD snooping is disabled, report suppression is disabled, and all MLDv1 reports are flooded to the ingress VLAN.

The switch also supports MLDv1 proxy reporting. When an MLDv1 MASQ is received, the switch responds with MLDv1 reports for the address on which the query arrived if the group exists in the switch on another port and if the port on which the query arrived is not the last member port for the address.

MLD Done Messages and Immediate-Leave

When the Immediate-Leave feature is enabled and a host sends an MLDv1 Done message (equivalent to an IGMP leave message), the port on which the Done message was received is immediately deleted from the group. You enable Immediate-Leave on VLANs and (as with IGMP snooping), you should only use the feature on VLANs where a single host is connected to the port. If the port was the last member of a group, the group is also deleted, and the leave information is forwarded to the detected IPv6 multicast routers.

When Immediate Leave is not enabled in a VLAN (which would be the case when there are multiple clients for a group on the same port) and a Done message is received on a port, an MASQ is generated on that port.

The user can control when a port membership is removed for an existing address in terms of the number of MASQs. A port is removed from membership to an address when there are no MLDv1 reports to the address on the port for the configured number of queries.

The number of MASQs generated is configured by using the **ipv6 mld snooping last-listener-query count** global configuration command. The default number is 2.

The MASQ is sent to the IPv6 multicast address for which the Done message was sent. If there are no reports sent to the IPv6 multicast address specified in the MASQ during the switch maximum response time, the port on which the MASQ was sent is deleted from the IPv6 multicast address database. The maximum response time is the time configured by using the **ipv6 mld snooping last-listener-query-interval** global configuration command. If the deleted port is the last member of the multicast address, the multicast address is also deleted, and the switch sends the address leave information to all detected multicast routers.

Topology Change Notification Processing

When topology change notification (TCN) solicitation is enabled by using the **ipv6 mld snooping tcn query solicit** global configuration command, MLDv1 snooping sets the VLAN to flood all IPv6 multicast traffic with a configured number of MLDv1 queries before it begins sending multicast data only to selected ports. You set this value by using the **ipv6 mld snooping tcn flood query count** global configuration command. The default is to send two queries. The switch also generates MLDv1 global Done messages with valid link-local IPv6 source addresses when the switch becomes the STP root in the VLAN or when it is configured by the user. This is same as done in IGMP snooping.

How to Configure IPv6 MLD Snooping

Default MLD Snooping Configuration

Table 22: Default MLD Snooping Configuration

Feature	Default Setting
MLD snooping (Global)	Disabled.
MLD snooping (per VLAN)	Enabled. MLD snooping must be globally enabled for VLAN MLD snooping to take place.
IPv6 Multicast addresses	None configured.
IPv6 Multicast router ports	None configured.
MLD snooping Immediate Leave	Disabled.
MLD snooping robustness variable	Global: 2; Per VLAN: 0. Note The VLAN value overrides the global setting. When the VLAN value is 0, the VLAN uses the global count.

Feature	Default Setting
Last listener query count	Global: 2; Per VLAN: 0. Note The VLAN value overrides the global setting. When the VLAN value is 0, the VLAN uses the global count.
Last listener query interval	Global: 1000 (1 second); VLAN: 0. Note The VLAN value overrides the global setting. When the VLAN value is 0, the VLAN uses the global interval.
TCN query solicit	Disabled.
TCN query count	2.
MLD listener suppression	Disabled.

MLD Snooping Configuration Guidelines

When configuring MLD snooping, consider these guidelines:

- You can configure MLD snooping characteristics at any time, but you must globally enable MLD snooping by using the **ipv6 mld snooping** global configuration command for the configuration to take effect.
- MLD snooping and IGMP snooping act independently of each other. You can enable both features at the same time on the switch.
- The maximum number of address entries allowed for the switch is 4000.

Enabling or Disabling MLD Snooping on the Switch

By default, IPv6 MLD snooping is globally disabled on the switch and enabled on all VLANs. When MLD snooping is globally disabled, it is also disabled on all VLANs. When you globally enable MLD snooping, the VLAN configuration overrides the global configuration. That is, MLD snooping is enabled only on VLAN interfaces in the default state (enabled).

You can enable and disable MLD snooping on a per-VLAN basis or for a range of VLANs, but if you globally disable MLD snooping, it is disabled in all VLANs. If global snooping is enabled, you can enable or disable VLAN snooping.

To globally enable MLD snooping on the switch, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example:	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
	Device> enable	
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ipv6 mld snooping Example: Device(config)# ipv6 mld snooping	Enables MLD snooping on the switch.
Step 4	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 5	copy running-config startup-config Example: Device(config)# copy running-config startup-config	(Optional) Save your entries in the configuration file.
Step 6	reload Example: Device(config)# reload	Reload the operating system.

Enabling or Disabling MLD Snooping on a VLAN

To enable MLD snooping on a VLAN, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ipv6 mld snooping Example: <pre>Device(config)# ipv6 mld snooping</pre>	Enables MLD snooping on the switch.
Step 4	ipv6 mld snooping vlan <i>vlan-id</i> Example: <pre>Device(config)# ipv6 mld snooping vlan 1</pre>	Enables MLD snooping on the VLAN. The VLAN ID range is 1 to 1001 and 1006 to 4094. Note MLD snooping must be globally enabled for VLAN snooping to be enabled.
Step 5	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.

Configuring a Static Multicast Group

Hosts or Layer 2 ports normally join multicast groups dynamically, but you can also statically configure an IPv6 multicast address and member ports for a VLAN.

To add a Layer 2 port as a member of a multicast group, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ipv6 mld snooping vlan <i>vlan-id</i> static <i>ipv6_multicast_address</i> interface <i>interface-id</i>	Configures a multicast group with a Layer 2 port as a member of a multicast group:

	Command or Action	Purpose
	Example: <pre>Device(config)# ipv6 mld snooping vlan 1 static 3333.0000.1111 interface gigabitethernet 1/ 1/0/1</pre>	<ul style="list-style-type: none"> • <i>vlan-id</i> is the multicast group VLAN ID. The VLAN ID range is 1 to 1001 and 1006 to 4094. • <i>ipv6_multicast_address</i> is the 128-bit group IPv6 address. The address must be in the form specified in RFC 2373. • <i>interface-id</i> is the member port. It can be a physical interface or a port channel (1 to 48).
Step 4	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 5	Use one of the following: <ul style="list-style-type: none"> • show ipv6 mld snooping address • show ipv6 mld snooping address vlan <i>vlan-id</i> Example: <pre>Device# show ipv6 mld snooping address OR Device# show ipv6 mld snooping vlan 1</pre>	Verifies the static member port and the IPv6 address.

Configuring a Multicast Router Port



Note Static connections to multicast routers are supported only on switch ports.

To add a multicast router port to a VLAN, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 3	ipv6 mld snooping vlan <i>vlan-id</i> mrouter interface <i>interface-id</i> Example: Device(config)# <code>ipv6 mld snooping vlan 1 mrouter interface gigabitethernet 1/0/2</code>	Specifies the multicast router VLAN ID, and specify the interface to the multicast router. <ul style="list-style-type: none"> • The VLAN ID range is 1 to 1001 and 1006 to 4094. • The interface can be a physical interface or a port channel. The port-channel range is 1 to 48.
Step 4	end Example: Device(config)# <code>end</code>	Returns to privileged EXEC mode.
Step 5	show ipv6 mld snooping mrouter [vlan <i>vlan-id</i>] Example: Device# <code>show ipv6 mld snooping mrouter vlan 1</code>	Verifies that IPv6 MLD snooping is enabled on the VLAN interface.

Enabling MLD Immediate Leave

To enable MLDv1 immediate leave, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> <code>enable</code>	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 3	ipv6 mld snooping vlan <i>vlan-id</i> immediate-leave Example:	Enables MLD Immediate Leave on the VLAN interface.

	Command or Action	Purpose
	Device(config)# ipv6 mld snooping vlan 1 immediate-leave	
Step 4	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 5	show ipv6 mld snooping vlan <i>vlan-id</i> Example: Device# show ipv6 mld snooping vlan 1	Verifies that Immediate Leave is enabled on the VLAN interface.

Configuring MLD Snooping Queries

To configure MLD snooping query characteristics for the switch or for a VLAN, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ipv6 mld snooping robustness-variable <i>value</i> Example: Device(config)# ipv6 mld snooping robustness-variable 3	(Optional) Sets the number of queries that are sent before switch will deletes a listener (port) that does not respond to a general query. The range is 1 to 3; the default is 2.
Step 4	ipv6 mld snooping vlan <i>vlan-id</i> robustness-variable <i>value</i> Example: Device(config)# ipv6 mld snooping vlan 1 robustness-variable 3	(Optional) Sets the robustness variable on a VLAN basis, which determines the number of general queries that MLD snooping sends before aging out a multicast address when there is no MLD report response. The range is 1 to 3; the default is 0. When set to 0, the number used is the global robustness variable value.
Step 5	ipv6 mld snooping last-listener-query-count <i>count</i>	(Optional) Sets the number of MASQs that the switch sends before aging out an MLD client.

	Command or Action	Purpose
	Example: Device(config)# ipv6 mld snooping last-listener-query-count 7	The range is 1 to 7; the default is 2. The queries are sent 1 second apart.
Step 6	ipv6 mld snooping vlan <i>vlan-id</i> last-listener-query-count <i>count</i> Example: Device(config)# ipv6 mld snooping vlan 1 last-listener-query-count 7	(Optional) Sets the last-listener query count on a VLAN basis. This value overrides the value configured globally. The range is 1 to 7; the default is 0. When set to 0, the global count value is used. Queries are sent 1 second apart.
Step 7	ipv6 mld snooping last-listener-query-interval <i>interval</i> Example: Device(config)# ipv6 mld snooping last-listener-query-interval 2000	(Optional) Sets the maximum response time that the switch waits after sending out a MASQ before deleting a port from the multicast group. The range is 100 to 32,768 thousands of a second. The default is 1000 (1 second).
Step 8	ipv6 mld snooping vlan <i>vlan-id</i> last-listener-query-interval <i>interval</i> Example: Device(config)# ipv6 mld snooping vlan 1 last-listener-query-interval 2000	(Optional) Sets the last-listener query interval on a VLAN basis. This value overrides the value configured globally. The range is 0 to 32,768 thousands of a second. The default is 0. When set to 0, the global last-listener query interval is used.
Step 9	ipv6 mld snooping tcn query solicit Example: Device(config)# ipv6 mld snooping tcn query solicit	(Optional) Enables topology change notification (TCN) solicitation, which means that VLANs flood all IPv6 multicast traffic for the configured number of queries before sending multicast data to only those ports requesting to receive it. The default is for TCN to be disabled.
Step 10	ipv6 mld snooping tcn flood query count <i>count</i> Example: Device(config)# ipv6 mld snooping tcn flood query count 5	(Optional) When TCN is enabled, specifies the number of TCN queries to be sent. The range is from 1 to 10; the default is 2.
Step 11	end	Returns to privileged EXEC mode.
Step 12	show ipv6 mld snooping querier [<i>vlan</i> <i>vlan-id</i>] Example: Device(config)# show ipv6 mld snooping querier vlan 1	(Optional) Verifies that the MLD snooping querier information for the switch or for the VLAN.

Disabling MLD Listener Message Suppression

MLD snooping listener message suppression is enabled by default. When it is enabled, the switch forwards only one MLD report per multicast router query. When message suppression is disabled, multiple MLD reports could be forwarded to the multicast routers.

To disable MLD listener message suppression, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enter global configuration mode.
Step 3	no ipv6 mld snooping listener-message-suppression Example: Device(config)# no ipv6 mld snooping listener-message-suppression	Disable MLD message suppression.
Step 4	end Example: Device(config)# end	Return to privileged EXEC mode.
Step 5	show ipv6 mld snooping Example: Device# show ipv6 mld snooping	Verify that IPv6 MLD snooping report suppression is disabled.

Displaying MLD Snooping Information

You can display MLD snooping information for dynamically learned and statically configured router ports and VLAN interfaces. You can also display IPv6 group address multicast entries for a VLAN configured for MLD snooping.

Table 23: Commands for Displaying MLD Snooping Information

Command	Purpose
show ipv6 mld snooping [vlan <i>vlan-id</i>]	Displays the MLD snooping configuration information for all VLANs on the switch or for a specified VLAN. (Optional) Enter vlan <i>vlan-id</i> to display information for a single VLAN. The VLAN ID range is 1 to 1001 and 1006 to 4094.
show ipv6 mld snooping mrouter [vlan <i>vlan-id</i>]	Displays information on dynamically learned and manually configured multicast router interfaces. When you enable MLD snooping, the switch automatically learns the interface to which a multicast router is connected. These are dynamically learned interfaces. (Optional) Enters vlan <i>vlan-id</i> to display information for a single VLAN. The VLAN ID range is 1 to 1001 and 1006 to 4094.
show ipv6 mld snooping querier [vlan <i>vlan-id</i>]	Displays information about the IPv6 address and incoming port for the most-recently received MLD query messages in the VLAN. (Optional) Enters vlan <i>vlan-id</i> to display information for a single VLAN. The VLAN ID range is 1 to 1001 and 1006 to 4094.
show ipv6 mld snooping address [vlan <i>vlan-id</i>] [count dynamic user]	Displays all IPv6 multicast address information or specific IPv6 multicast address information for the switch or a VLAN. <ul style="list-style-type: none"> Enters count to show the group count on the switch or in a VLAN. Enters dynamic to display MLD snooping learned group information for the switch or for a VLAN. Enters user to display MLD snooping user-configured group information for the switch or for a VLAN.
show ipv6 mld snooping address vlan <i>vlan-id</i> [<i>ipv6-multicast-address</i>]	Displays MLD snooping for the specified VLAN and IPv6 multicast address.

Configuration Examples for Configuring MLD Snooping

Configuring a Static Multicast Group: Example

This example shows how to statically configure an IPv6 multicast group:

```
Device# configure terminal
Device(config)# ipv6 mld snooping vlan 2 static 3333.0000.1111 interface gigabitethernet1/0/1
Device(config)# end
```

Configuring a Multicast Router Port: Example

This example shows how to add a multicast router port to VLAN 200:

```
Device# configure terminal
Device(config)# ipv6 mld snooping vlan 200 mrouter interface gigabitethernet
1/0/2
Device(config)# exit
```

Enabling MLD Immediate Leave: Example

This example shows how to enable MLD Immediate Leave on VLAN 130:

```
Device# configure terminal
Device(config)# ipv6 mld snooping vlan 130 immediate-leave
Device(config)# exit
```

Configuring MLD Snooping Queries: Example

This example shows how to set the MLD snooping global robustness variable to 3:

```
Device# configure terminal
Device(config)# ipv6 mld snooping robustness-variable 3
Device(config)# exit
```

This example shows how to set the MLD snooping last-listener query count for a VLAN to 3:

```
Device# configure terminal
Device(config)# ipv6 mld snooping vlan 200 last-listener-query-count 3
Device(config)# exit
```

This example shows how to set the MLD snooping last-listener query interval (maximum response time) to 2000 (2 seconds):

```
Device# configure terminal
Device(config)# ipv6 mld snooping last-listener-query-interval 2000
Device(config)# exit
```

Additional References

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	<i>Command Reference (Catalyst 9400 Series Switches)</i>

Feature History for MLD Snooping

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Releases	Feature Name	Feature Information
Cisco IOS XE Everest 16.6.1	MLD Snooping	MLD snooping allows the switch to examine MLD packets and make forwarding decisions based on their content.

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <https://cfng.cisco.com/>



CHAPTER 14

Configuring Multicast Virtual Private Network

- [Prerequisites for Configuring Multicast VPN, on page 289](#)
- [Restrictions for Configuring Multicast VPN, on page 289](#)
- [Information About Configuring Multicast VPN, on page 289](#)
- [How to Configure Multicast VPN, on page 293](#)
- [Configuration Examples for Multicast VPN, on page 300](#)
- [Additional References for Configuring Multicast VPN, on page 301](#)
- [Feature History for Multicast VPN, on page 301](#)

Prerequisites for Configuring Multicast VPN

Enable IP multicast and configure the PIM interfaces using the tasks described in the “Configuring Basic IP Multicast” module.

Restrictions for Configuring Multicast VPN

- The update source interface for the Border Gateway Protocol (BGP) peerings must be the same for all BGP peerings configured on the device in order for the default multicast distribution tree (MDT) to be configured properly. If you use a loopback address for BGP peering, PIM sparse mode must be enabled on the loopback address.
- MVPN does not support multiple BGP peering update sources.
- Multiple BGP update sources are not supported, and configuring them can break MVPN reverse path forwarding (RPF) checking. The source IP address of the MVPN tunnels is determined by the highest IP address used for the BGP peering update source. If this IP address is not the IP address used as the BGP peering address with the remote provider edge (PE) device, MVPN will not function properly.

Information About Configuring Multicast VPN

This section provides information about configuring Multicast VPN:

Multicast VPN Operation

MVPN IP allows a service provider to configure and support multicast traffic in an MPLS VPN environment. This feature supports routing and forwarding of multicast packets for each individual VRF instance, and it also provides a mechanism to transport VPN multicast packets across the service provider backbone.

A VPN is network connectivity across a shared infrastructure, such as an ISP. Its function is to provide the same policies and performance as a private network, at a reduced cost of ownership, thus creating many opportunities for cost savings through operations and infrastructure.

An MVPN allows an enterprise to transparently interconnect its private network across the network backbone of a service provider. The use of an MVPN to interconnect an enterprise network in this way does not change the way that enterprise network is administered, nor does it change general enterprise connectivity.

Benefits of Multicast VPN

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

Multicast VPN Routing and Forwarding and Multicast Domains

MVPN introduces multicast routing information to the VPN routing and forwarding table. When a provider edge (PE) device receives multicast data or control packets from a customer edge (CE) router, forwarding is performed according to the information in the Multicast VPN routing and forwarding instance (MVRF). MVPN does not use label switching.

A set of MVRFs that can send multicast traffic to each other constitutes a multicast domain. For example, the multicast domain for a customer that wanted to send certain types of multicast traffic to all global employees would consist of all CE routers associated with that enterprise.

Multicast Distribution Trees

MVPN establishes a static default multicast distribution tree (MDT) for each multicast domain. The default MDT defines the path used by PE routers to send multicast data and control messages to every other PE router in the multicast domain.

If Source Specific Multicast (SSM) is used as the core multicast routing protocol, the multicast IP addresses used for the default and data MDT must be configured within the SSM range on all PE routers.

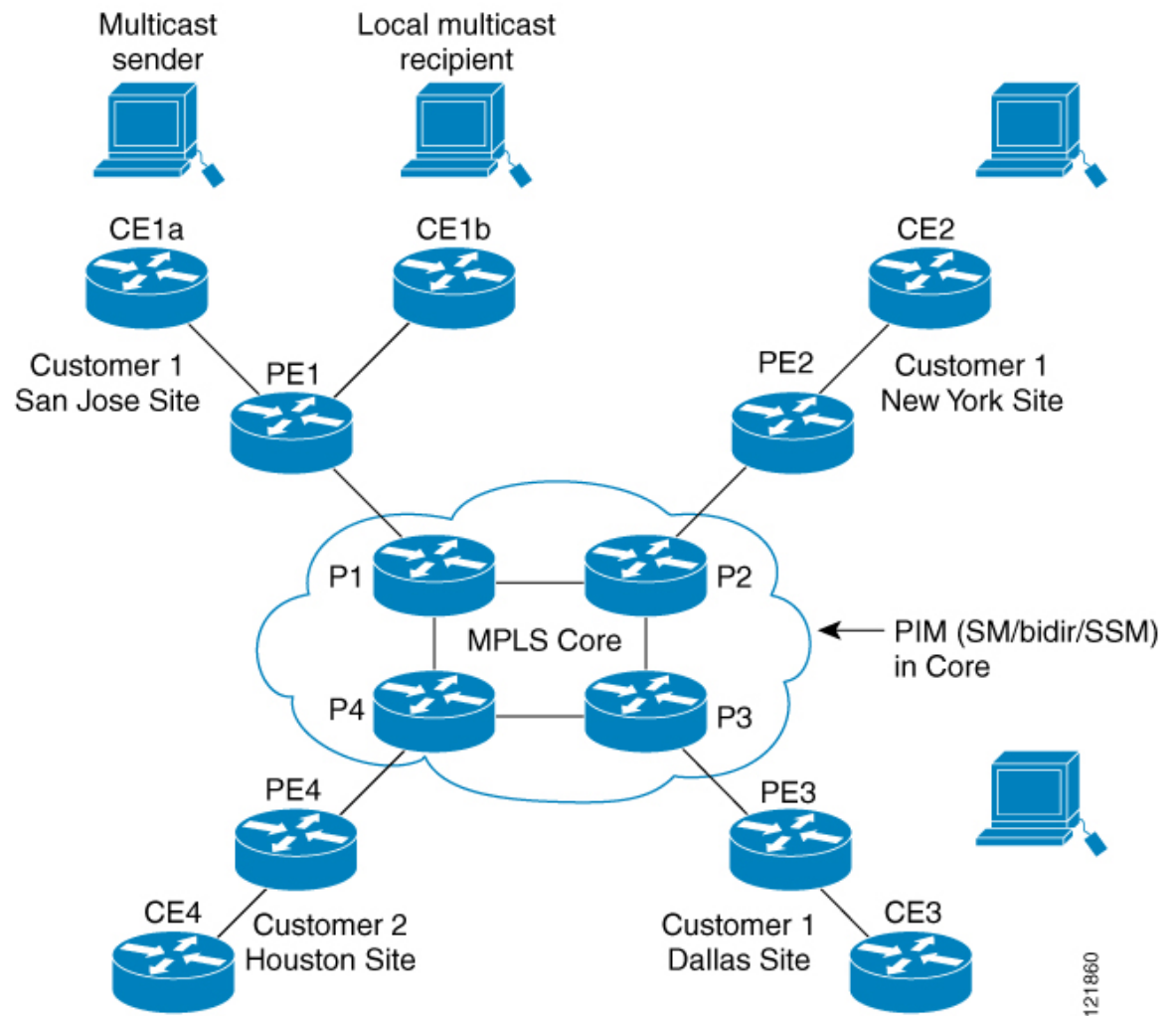
MVPN also supports the dynamic creation of MDTs for high-bandwidth transmission. Data MDTs are a feature unique to Cisco IOS software. Data MDTs are intended for high-bandwidth sources such as full-motion video inside the VPN to ensure optimal traffic forwarding in the MPLS VPN core. The threshold at which the data MDT is created can be configured on a per-router or a per-VRF basis. When the multicast transmission exceeds the defined threshold, the sending PE router creates the data MDT and sends a UDP message, which contains information about the data MDT, to all routers on the default MDT. The statistics to determine whether a multicast stream has exceeded the data MDT threshold are examined once every second. After a PE router sends the UDP message, it waits 3 more seconds before switching over; 13 seconds is the worst case switchover time, and 3 seconds is the best case.

Data MDTs are created only for (S, G) multicast route entries within the VRF multicast routing table. They are not created for (*, G) entries regardless of the value of the individual source data rate.

In the following example, a service provider has a multicast customer with offices in San Jose, New York, and Dallas. A one-way multicast presentation is occurring in San Jose. The service provider network supports all three sites associated with this customer, in addition to the Houston site of a different enterprise customer.

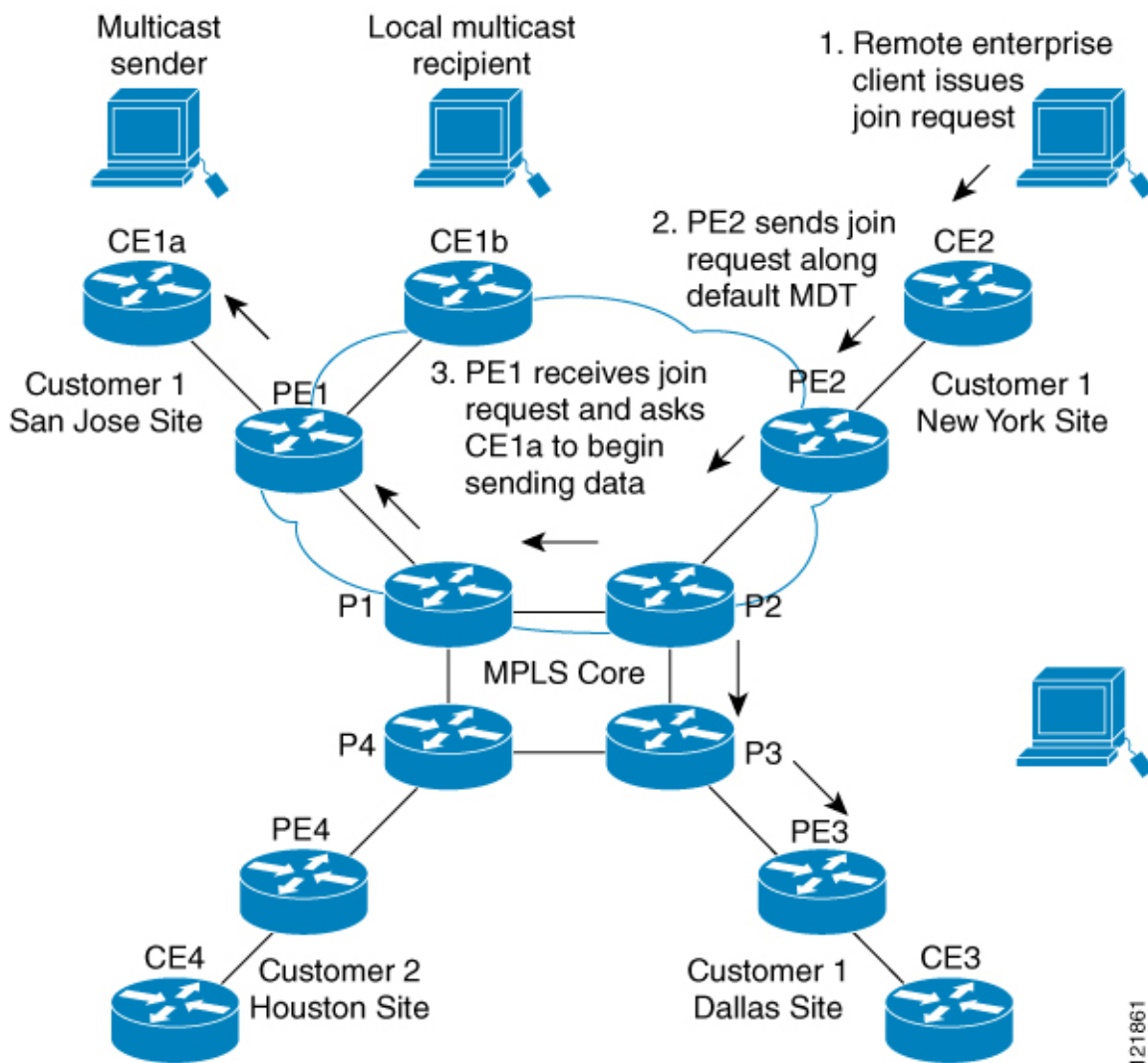
The default MDT for the enterprise customer consists of provider routers P1, P2, and P3 and their associated PE routers. PE4 is not part of the default MDT, because it is associated with a different customer. The figure shows that no data flows along the default MDT, because no one outside of San Jose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

Figure 23: Default Multicast Distribution Tree Overview



An employee in New York joins the multicast session. The PE router associated with the New York site sends a join request that flows across the default MDT for the multicast domain of the customer. PE1, the PE router associated with the multicast session source, receives the request. The figure depicts that the PE router forwards the request to the CE router associated with the multicast source (CE1a).

Figure 24: Initializing the Data MDT



The CE router (CE1a) begins to send the multicast data to the associated PE router (PE1), which sends the multicast data along the default MDT. Immediately sending the multicast data, PE1 recognizes that the multicast data exceeds the bandwidth threshold for which a data MDT should be created. Therefore, PE1 creates a data MDT, sends a message to all routers using the default MDT, which contains information about the data MDT, and, three seconds later, begins sending the multicast data for that particular stream using the data MDT. Only PE2 has interested receivers for this source, so only PE2 will join the data MDT and receive traffic on it.

PE routers maintain a PIM relationship with other PE routers over the default MDT and a PIM relationship with directly attached PE routers.

Multicast Tunnel Interface

An MVRF, which is created per multicast domain, requires the device to create a tunnel interface from which all MVRF traffic is sourced. A multicast tunnel interface is an interface that the MVRF uses to access the

multicast domain. It can be thought of as a conduit that connects an MVRF and the global MVRF. One tunnel interface is created per MVRF.

MDT Address Family in BGP for Multicast VPN

The **mdt** keyword has been added to the **address-family ipv4** command to configure an MDT address-family session. MDT address-family sessions are used to pass the source PE address and MDT group address to PIM using Border Gateway Protocol (BGP) MDT Subaddress Family Identifier (SAFI) updates.

BGP Advertisement Methods for Multicast VPN Support

In a single autonomous system, if the default MDT for an MVPN is using PIM sparse mode (PIM-SM) with a rendezvous point (RP), then PIM is able to establish adjacencies over the Multicast Tunnel Interface (MTI) because the source PE and receiver PE discover each other through the RP. In this scenario, the local PE (the source PE) sends register messages to the RP, which then builds a shortest-path tree (SPT) toward the source PE. The remote PE, which acts as a receiver for the MDT multicast group, then sends (*, G) joins toward the RP and joins the distribution tree for that group.

However, if the default MDT group is configured in a PIM Source Specific Multicast (PIM-SSM) environment rather than a PIM-SM environment, the receiver PE needs information about the source PE and the default MDT group. This information is used to send (S, G) joins toward the source PE to build a distribution tree from the source PE (without the need for an RP). The source PE address and default MDT group address are sent using BGP.

BGP Extended Community

When BGP extended communities are used, the PE loopback (source address) information is sent as a VPNv4 prefix using Route Distinguisher (RD) Type 2 (to distinguish it from unicast VPNv4 prefixes). The MDT group address is carried in a BGP extended community. Using a combination of the embedded source in the VPNv4 address and the group in the extended community, PE routers in the same MVRF instance can establish SSM trees to each other.



Note Prior to the introduction of MDT SAFI support, the BGP extended community attribute was used as an interim solution to advertise the IP address of the source PE and default MDT group before IETF standardization. A BGP extended community attribute in an MVPN environment, however, has certain limitations: it cannot be used in inter-AS scenarios (because the attribute is nontransitive), and it uses RD Type 2 (which is not a supported standard).

How to Configure Multicast VPN

This section provides the steps to follow while configuring Multicast VPN:

Configuring the Data Multicast Group

A data MDT group can include a maximum of 256 multicast groups per VPN per VRF per PE device. Multicast groups used to create the data MDT group are dynamically chosen from a pool of configured IP addresses. Use the following procedure to configure data multicast group on the device.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	vrf definition <i>vrf-name</i> Example: <pre>Device(config)# vrf definition vrf1</pre>	Enters VRF configuration mode and defines the VPN routing instance by assigning a VRF name.
Step 4	rd <i>route-distinguisher</i> Example: <pre>Device(config-vrf)# rd 1:1</pre>	Creates routing and forwarding tables for a VRF. <ul style="list-style-type: none"> • The <i>route-distinguisher</i> argument specifies to add an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter a <i>route-distinguisher</i> in either of these formats: • 16-bit autonomous system number (ASN): your 32-bit number. For example, 101:3. • 32-bit IP address: your 16-bit number. For example, 192.168.122.15:1.
Step 5	route-target both <i>ASN:nn</i> or <i>IP-address:nn</i> Example: <pre>Device(config-vrf)# route-target both 1:1</pre>	Creates a route-target extended community for a VRF. The both keyword specifies to import both import and export routing information to the target VPN extended community.
Step 6	address family ipv4 unicast <i>value</i> Example: <pre>Device(config-vrf)# address family ipv4 unicast</pre>	Enters VRF address family configuration mode to specify an address family for a VRF. <ul style="list-style-type: none"> • The ipv4 keyword specifies an IPv4 address family for a VRF
Step 7	mdt default <i>group-address</i> Example: <pre>Device(config-vrf-af)# mdt default 226.10.10.10</pre>	Configures the multicast group address range for data MDT groups for a VRF. <ul style="list-style-type: none"> • A tunnel interface is created as a result of this command.

	Command or Action	Purpose
		<ul style="list-style-type: none"> The default MDT group address configuration must be the same on all PEs in the same VRF.
Step 8	mdt data <i>group number</i> Example: <pre>Device(config-vrf-af)# mdt data 232.0.1.0 0.0.0.31</pre>	Specifies a range of addresses to be used in the data MDT pool.
Step 9	mdt data threshold <i>kbps</i> Example: <pre>Device(config-vrf-af)# mdt data threshold 50</pre>	Specifies the threshold in <i>kbps</i> . The range is from 1 to 4294967.
Step 10	mdt log-reuse Example: <pre>Device(config-vrf-af)# mdt log-reuse</pre>	(Optional) Enables the recording of data MDT reuse and generates a syslog message when a data MDT has been reused.
Step 11	end Example: <pre>Device(config-vrf-af)# end</pre>	Returns to privileged EXEC mode.

Configuring a Default MDT Group for a VRF

Perform this task to configure a default MDT group for a VRF.

The default MDT group must be the same group configured on all devices that belong to the same VPN. The source IP address will be the address used to source the BGP sessions.

Procedure

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

	Command or Action	Purpose
Step 3	ip multicast-routing Example: <pre>Device(config)# ip multicast-routing</pre>	Enables multicast routing.
Step 4	ip multicast-routing vrf vrf-name Example: <pre>Device(config)# ip multicast-routing vrf vrf1</pre>	Supports the MVPN VRF instance.
Step 5	vrf definition vrf-name Example: <pre>Device(config)# vrf definition vrf1</pre>	Enters VRF configuration mode and defines the VPN routing instance by assigning a VRF name.
Step 6	rd route-distinguisher Example: <pre>Device(config-vrf)# rd 1:1</pre>	Creates routing and forwarding tables for a VRF. <ul style="list-style-type: none"> • The <i>route-distinguisher</i> argument specifies to add an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter a <i>route-distinguisher</i> in either of these formats: • 16-bit autonomous system number (ASN): your 32-bit number. For example, 101:3. • 32-bit IP address: your 16-bit number. For example, 192.168.122.15:1.
Step 7	route-target both ASN:nn or IP-address:nn Example: <pre>Device(config-vrf)# route-target both 1:1</pre>	Creates a route-target extended community for a VRF. The both keyword specifies to import both import and export routing information to the target VPN extended community.
Step 8	address family ipv4 unicast value Example: <pre>Device(config-vrf)# address family ipv4 unicast</pre>	Enters VRF address family configuration mode to specify an address family for a VRF. <ul style="list-style-type: none"> • The ipv4 keyword specifies an IPv4 address family for a VRF
Step 9	mdt default group-address Example: <pre>Device(config-vrf-af)# mdt default 226.10.10.10</pre>	Configures the multicast group address range for data MDT groups for a VRF. <ul style="list-style-type: none"> • A tunnel interface is created as a result of this command.

	Command or Action	Purpose
		<ul style="list-style-type: none"> The default MDT group address configuration must be the same on all PEs in the same VRF.
Step 10	end Example: Device(config-vrf-af)# end	Returns to privileged EXEC mode.
Step 11	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 12	ip pim vrf vrf-name rp-address value Example: Device(config-vrf-af)# ip pim vrf vrf1 rp-address 1.1.1.1	Enters the RP configuration mode.

Configuring the MDT Address Family in BGP for Multicast VPN

Perform this task to configure an MDT address family session on PE devices to establish MDT peering sessions for MVPN.

Before you begin

Before MVPN peering can be established through an MDT address family, MPLS and Cisco Express Forwarding (CEF) must be configured in the BGP network and multiprotocol BGP on PE devices that provide VPN services to CE devices.



Note The following policy configuration parameters are not supported:

- Route-originator attribute
- Network Layer Reachability Information (NLRI) prefix filtering (prefix lists, distribute lists)
- Extended community attributes (route target and site of origin)

Procedure

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

	Command or Action	Purpose
	Device> enable	
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Device(config)# router bgp 65535	Enters router configuration mode and creates a BGP routing process.
Step 4	address-family ipv4 mdt Example: Device(config-router)# address-family ipv4 mdt	Enters address family configuration mode to create an IP MDT address family session.
Step 5	neighbor <i>neighbor-address</i> activate Example: Device(config-router-af)# neighbor 192.168.1.1 activate	Enables the MDT address family for this neighbor.
Step 6	neighbor <i>neighbor-address</i> send-community [both extended standard] Example: Device(config-router-af)# neighbor 192.168.1.1 send-community extended	Enables community and (or) extended community exchange with the specified neighbor.
Step 7	exit Example: Device(config-router-af)# exit	Exits address family configuration mode and returns to router configuration mode.
Step 8	address-family vpnv4 Example: Device(config-router)# address-family vpnv4	Enters address family configuration mode to create a VPNv4 address family session.
Step 9	neighbor <i>neighbor-address</i> activate Example: Device(config-router-af)# neighbor 192.168.1.1 activate	Enables the VPNv4 address family for this neighbor.

	Command or Action	Purpose
Step 10	neighbor <i>neighbor-address</i> send-community [both extended standard] Example: Device(config-router-af)# neighbor 192.168.1.1 send-community extended	Enables community and (or) extended community exchange with the specified neighbor.
Step 11	end Example: Device(config-router-af)# end	Exits address family configuration mode and enters privileged EXEC mode.

Verifying Information for the MDT Default Group

Procedure

- Step 1** **enable**
- Example:**
- ```
Device> enable
```
- Enables privileged EXEC mode.
- Enter your password if prompted.
- Step 2**      **show ip pim [vrf vrf-name] mdt bgp**
- Example:**
- ```
Device# show ip pim mdt bgp
```
- ```
MDT-default group 232.2.1.4
rid:1.1.1.1 next_hop:1.1.1.1
```
- Displays information about the BGP advertisement of the RD for the MDT default group.
- Step 3**      **show ip pim [vrf vrf-name] mdt send**
- Example:**
- ```
Device# show ip pim mdt send
```
- ```
MDT-data send list for VRF:vpn8
(source, group) MDT-data group ref_count
(10.100.8.10, 225.1.8.1) 232.2.8.0 1
(10.100.8.10, 225.1.8.2) 232.2.8.1 1
(10.100.8.10, 225.1.8.3) 232.2.8.2 1
(10.100.8.10, 225.1.8.4) 232.2.8.3 1
(10.100.8.10, 225.1.8.5) 232.2.8.4 1
(10.100.8.10, 225.1.8.6) 232.2.8.5 1
(10.100.8.10, 225.1.8.7) 232.2.8.6 1
(10.100.8.10, 225.1.8.8) 232.2.8.7 1
(10.100.8.10, 225.1.8.9) 232.2.8.8 1
(10.100.8.10, 225.1.8.10) 232.2.8.9 1
```

Displays detailed information about the MDT data group including MDT advertisements that the specified device has made.

**Step 4** `show ip pim vrf vrf-name mdt history interval minutes`

**Example:**

```
Device# show ip pim vrf vrf1 mdt history interval 20
```

```
MDT-data send history for VRF - vrf1 for the past 20 minutes
MDT-data group Number of reuse
10.9.9.8 3
10.9.9.9 2
```

Displays the data MDTs that have been reused during the past configured interval.

## Configuration Examples for Multicast VPN

The following section provides the configuration examples for Multicast VPN:

### Example: Configuring MVPN and SSM

In the following example, PIM-SSM is configured in the backbone. Therefore, the default and data MDT groups are configured within the SSM range of IP addresses. Inside the VPN, PIM-SM is configured and only Auto-RP announcements are accepted.

```
ip vrf vrf1
 rd 1:1
 route-target export 1:1
 route-target import 1:1
 mdt default 232.0.0.1
 mdt data 232.0.1.0 0.0.0.255 threshold 500 list 101
!
ip pim ssm default
ip pim vrf vrf1 accept-rp auto-rp
```

### Example: Enabling a VPN for Multicast Routing

In the following example, multicast routing is enabled with a VPN routing instance named vrf1:

```
ip multicast-routing vrf1
```

### Example: Configuring the Multicast Group Address Range for Data MDT Groups

In the following example, the VPN routing instance is assigned a VRF named blue. The MDT default group for a VPN VRF is 239.1.1.1, and the multicast group address range for MDT groups is 239.1.2.0 with wildcard bits of 0.0.0.3:

```
ip vrf blue
 rd 55:1111
 route-target both 55:1111
```

```
mdt default 239.1.1.1
mdt data 239.1.2.0 0.0.0.3
end
```

## Example: Limiting the Number of Multicast Routes

In the following example, the number of multicast routes that can be added to a multicast routing table is set to 200,000 and the threshold value of the number of mroutes that will cause a warning message to occur is set to 20,000:

```
!
ip multicast-routing
ip multicast-routing vrf cisco
ip multicast cache-headers
ip multicast route-limit 200000 20000
ip multicast vrf cisco route-limit 200000 20000
no mpls traffic-eng auto-bw timers frequency 0
!
```

## Additional References for Configuring Multicast VPN

### Related Documents

| Related Topic                                                                    | Document Title                                                                                         |
|----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| For complete syntax and usage information for the commands used in this chapter. | See the Multicast VPN Commands section of the <i>Command Reference (Catalyst 9400 Series Switches)</i> |

## Feature History for Multicast VPN

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

| Releases                    | Feature Name  | Feature Information                                                                                                                       |
|-----------------------------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| Cisco IOS XE Everest 16.6.1 | Multicast VPN | A Multicast VPN allows an enterprise to transparently interconnect its private network across the network backbone of a service provider. |

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <https://cfng.cisco.com/>





## CHAPTER 15

# Configuring MVPNv6

- [Prerequisites for MVPNv6, on page 303](#)
- [Restrictions for MVPNv6, on page 303](#)
- [Information about MVPNv6, on page 303](#)
- [How to Configure MVPNv6, on page 304](#)
- [Configuration Example for MVPNv6, on page 307](#)
- [Feature History for MVPNv6, on page 308](#)

## Prerequisites for MVPNv6

- BGP must be configured and operational on all devices sending or receiving multicast traffic.
- BGP extended communities must be enabled to support the use of Multicast Distribution Trees (MDTs) in the network. Use the **neighbor send-community both** or **neighbor send-community extended** command to enable BGP extended communities.
- VPN routing and forwarding (MVRP) instances to be used for MVPNv6 must be configured on the PE devices.

## Restrictions for MVPNv6

- Point-to-point GRE tunnel as an output interface in a VRF for MVPNv6 is not supported.

## Information about MVPNv6

To provide Layer 3 multicast services to customers with multiple distributed sites, service providers need a secure and scalable mechanism to transmit multicast traffic across the service-provider network. IPv4 Multicast VPN (MVPN) provides such services for IPv4 multicast traffic over a shared service provider backbone.

IPv6 Multicast Virtual Private Network (MVPNv6) provides the same services for IPv6 traffic, enabling service providers to provide multicast-enabled private IPv6 networks to their customers using their existing IPv4 back bone. The IPv4 and IPv6 VPN traffic is carried over the same tunnels simultaneously.

# How to Configure MVPNv6

## Configuring Multicast Routing

Perform this task to enable IPv4 and IPv6 multicast routing for the multicast VPN routing and forwarding (MVRF) instance to be used for MVPNv6.

### Procedure

|               | Command or Action                                                                            | Purpose                                                           |
|---------------|----------------------------------------------------------------------------------------------|-------------------------------------------------------------------|
| <b>Step 1</b> | <b>enable</b><br><b>Example:</b><br>Device> enable                                           | Enables privileged EXEC mode.<br>Enter your password if prompted. |
| <b>Step 2</b> | <b>configure terminal</b><br><b>Example:</b><br>Device# configure terminal                   | Enters global configuration mode.                                 |
| <b>Step 3</b> | <b>ip routing</b><br><b>Example:</b><br>Device(config)# ip routing                           | Enables IPv4 multicast routing.                                   |
| <b>Step 4</b> | <b>ip routing vrf vrf-name</b><br><b>Example:</b><br>Device(config)# ip routing vrf blue     | Enables IPv4 multicast routing for the specified MVRF instance.   |
| <b>Step 5</b> | <b>ipv6 routing</b><br><b>Example:</b><br>Device(config)# ipv6 routing                       | Enables IPv6 multicast routing.                                   |
| <b>Step 6</b> | <b>ipv6 routing vrf vrf-name</b><br><b>Example:</b><br>Device(config)# ipv6 routing vrf blue | Enables IPv6 multicast routing for the specified MVRF instance.   |
| <b>Step 7</b> | <b>exit</b><br><b>Example:</b><br>Device(config)# exit                                       | Exits global configuration mode.                                  |

## Configuring MVRF on PE Devices

### Procedure

|               | Command or Action                                                                                                                   | Purpose                                                                                                                                                                  |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>enable</b><br><b>Example:</b><br>Device> enable                                                                                  | Enables privileged EXEC mode.<br>Enter your password if prompted.                                                                                                        |
| <b>Step 2</b> | <b>configure terminal</b><br><b>Example:</b><br>Device# configure terminal                                                          | Enters global configuration mode.                                                                                                                                        |
| <b>Step 3</b> | <b>interface <i>type number</i></b><br><b>Example:</b><br>Device(config)# interface<br>GigabitEthernet 3/0/3                        | Enters interface configuration mode.                                                                                                                                     |
| <b>Step 4</b> | <b>vrf forwarding <i>vrf-name</i></b><br><b>Example:</b><br>Device(config-if)# vrf forwarding blue                                  | Associates a VRF with the interface.                                                                                                                                     |
| <b>Step 5</b> | <b>ip address <i>ip-address mask</i></b><br><b>Example:</b><br>Device(config-if)# ip address 10.1.0.1<br>255.255.0.0                | Configures an IPv4 address on the interface.                                                                                                                             |
| <b>Step 6</b> | <b>ip pim sparse-mode</b><br><b>Example:</b><br>Device(config-if)# ip pim sparse-mode                                               | Enables Protocol Independent Multicast (PIM) on the interface.                                                                                                           |
| <b>Step 7</b> | <b>delay <i>tens-of-seconds</i></b><br><b>Example:</b><br>Device(config-if)# delay 1000                                             | Configures delay value on the interface.                                                                                                                                 |
| <b>Step 8</b> | <b>ipv6 address <i>ipv6-address link-local</i></b><br><b>Example:</b><br>Device(config-if)# ipv6 address<br>FE80::20:1:1 link-local | Specifies a link-local IPv6 address.<br>This address is used instead of the link-local address that was automatically configured when IPv6 was enabled on the interface. |
| <b>Step 9</b> | <b>ipv6 address <i>ipv6-address-prefix</i></b><br><b>Example:</b><br>Device(config-if)# ipv6 address FC00::/7                       | Configures an IPv6 address on the interface.                                                                                                                             |

|                | Command or Action                                                                                                                   | Purpose                                                                                         |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| <b>Step 10</b> | <b>ipv6 pim</b><br><b>Example:</b><br>Device(config-if)# ipv6 pim                                                                   | Enables Protocol Independent Multicast (PIM) for IPv6.                                          |
| <b>Step 11</b> | <b>exit</b><br><b>Example:</b><br>Device(config-if)# exit                                                                           | Exits interface configuration mode.                                                             |
| <b>Step 12</b> | <b>ip pim rp-address ip-address</b><br><b>Example:</b><br>Device(config)# ip pim rp-address 10.10.10.10                             | Configure the address of a PIM rendezvous point (RP) for multicast groups.                      |
| <b>Step 13</b> | <b>ip pim vrf vrf-name rp-address address</b><br><b>Example:</b><br>Device(config)# ip pim vrf blue rp-address 10.10.0.10           | Configures the IPv4 address of a PIM RP and associates the RP with the specified MVRF instance. |
| <b>Step 14</b> | <b>ipv6 pim vrf vrf-name rp-address ipv6-address</b><br><b>Example:</b><br>Device(config)# ipv6 pim vrf blue rp-address FC00::1:1:1 | Configures the IPv6 address of a PIM RP and associates the RP with the specified MVRF instance. |
| <b>Step 15</b> | <b>exit</b><br><b>Example:</b><br>Device(config)# exit                                                                              | Exits global configuration mode.                                                                |

## Configuring Routing Protocols Between the PE and CE Devices

### Before you begin

The PE and CE devices must be configured with the same routing protocol.

### Procedure

|               | Command or Action                                                          | Purpose                                                           |
|---------------|----------------------------------------------------------------------------|-------------------------------------------------------------------|
| <b>Step 1</b> | <b>enable</b><br><b>Example:</b><br>Device> enable                         | Enables privileged EXEC mode.<br>Enter your password if prompted. |
| <b>Step 2</b> | <b>configure terminal</b><br><b>Example:</b><br>Device# configure terminal | Enters global configuration mode.                                 |



|               | Command or Action                                                                                                        | Purpose                                                                                                |
|---------------|--------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| <b>Step 3</b> | <b>router bgp <i>as-number</i></b><br><b>Example:</b><br>Device(config)# router bgp 55                                   | Specifies the number of an autonomous system that identifies the device to other BGP devices.          |
| <b>Step 4</b> | <b>address-family ipv6 vrf <i>vrf-name</i></b><br><b>Example:</b><br>Device(config-router)# address-family ipv6 vrf blue | Specifies the name of the VRF to associate with subsequent address family configuration mode commands. |
| <b>Step 5</b> | <b>redistribute connected</b><br><b>Example:</b><br>Device(config-router-af)# redistribute connected                     | Redistributes the directly connected networks to BGP.                                                  |
| <b>Step 6</b> | <b>redistribute eigrp <i>as-number</i></b><br><b>Example:</b><br>Device(config-router-af)# redistribute eigrp 11         | Redistributes the EIGRP routes into BGP.                                                               |
| <b>Step 7</b> | <b>redistribute static</b><br><b>Example:</b><br>Device(config-router-af)# redistribute static                           | Redistribute the static routes into BGP.                                                               |
| <b>Step 8</b> | <b>end</b><br><b>Example:</b><br>Device(config-router-af)# end                                                           | Returns to privileged EXEC mode.                                                                       |

## Configuration Example for MVPNv6

The following is an example for configuring MVPNv6:

```

mls ipv6 vrf
!
vrf definition blue
 rd 55:1111
 route-target export 55:1111
 route-target import 55:1111
!
 address-family ipv4
 mdt default 232.1.1.1
 exit-address-family
!
 address-family ipv6
 mdt default 232.1.1.1
 exit-address-family
!

ip multicast-routing

```

```

ip multicast-routing vrf blue
!
!
ipv6 unicast-routing
ipv6 multicast-routing
ipv6 multicast-routing vrf blue
!

interface GigabitEthernet3/0/3
 vrf forwarding blue
 ip address 10.1.0.1 255.255.255.0
 no ip redirects
 no ip proxy-arp
 ip pim sparse-dense-mode
 delay 100
 ipv6 address FE80::20:1:1 link-local
 ipv6 address FC00::/7
 no mls qos trust
!
router bgp 55
 address-family ipv6 vrf blue
 redistribute connected
 redistribute eigrp 11
 redistribute static
 exit-address-family
!

ip pim vrf blue rp-address 10.10.0.10
!
ipv6 pim vrf blue rp-address FC00::1:1:1
!
!
```

## Feature History for MVPNv6

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

| Release                        | Feature | Feature Information                                                                                                                                |
|--------------------------------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| Cisco IOS XE Gibraltar 16.11.1 | MVPNv6  | This feature enables service providers to use their existing IPv4 back bone to provide multicast-enabled private IPv6 networks to their customers. |

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>.



## CHAPTER 16

# Configuring Multicast VPN Extranet Support

- [Restrictions for Configuring mVPN Extranet Support, on page 309](#)
- [Information About mVPN Extranet Support, on page 309](#)
- [How to Configure mVPN Extranet Support, on page 314](#)
- [Configuration Examples for mVPN Extranet Support, on page 320](#)
- [Additional References, on page 336](#)
- [Feature History for mVPN Extranet Support, on page 336](#)

## Restrictions for Configuring mVPN Extranet Support

- The multicast VPN (mVPN) extranet support feature supports only Protocol Independent Multicast (PIM) sparse mode (PIM-SM) and Source Specific Multicast (SSM) traffic; PIM dense mode (PIM-DM) and bidirectional PIM (bidir-PIM) traffic are not supported.
- When configuring mVPN extranet in a PIM-SM environment, the source and the rendezvous point (RP) must reside in the same site of the mVPN behind the same provider edge (PE) router.
- IPv6 based mVPN extranet is not supported.

## Information About mVPN Extranet Support

The mVPN extranet support feature enables service providers to distribute IP multicast content originating from one enterprise site to other enterprise sites. With this feature, service providers can offer the next generation of flexible extranet services, helping to enable business partnerships between different enterprise VPN customers. Service providers can offer multicast extranet contracts to meet various business partnership requirements, including short-term, annual, and rolling contracts.

An extranet can be viewed as part of a company's intranet that is extended to users outside the company. With this feature, a VPN is used as a way to do business with other companies as well as to sell products and content to customers and companies. An extranet is a VPN connecting a corporate site or sites to external business partners or suppliers to securely share part of a business's information or operations among them. mVPN extranet support feature enables efficient content distribution between enterprises and from service providers or content providers to their different enterprise VPN customers.

Multiprotocol Label Switching (MPLS) VPNs inherently provide security, ensuring that users access only appropriate information. MPLS VPN extranet services offer extranet users unicast connectivity without

compromising the integrity of their corporate data. The mVPN extranet support feature extends this offer to include multicast connectivity to the extranet community of interest.

## Overview of mVPN Extranet Support

For unicast, there is no difference between an intranet or extranet from a routing perspective, that is, when a VRF imports a prefix, that prefix is reachable through a label-switched path (LSP). If the enterprise owns the prefix, the prefix is considered a part of the corporate intranet. Otherwise, the prefix is considered a part of an extranet. For multicast, however, the reachability of a prefix (especially through an LSP) is not sufficient to build a multicast distribution tree (MDT).

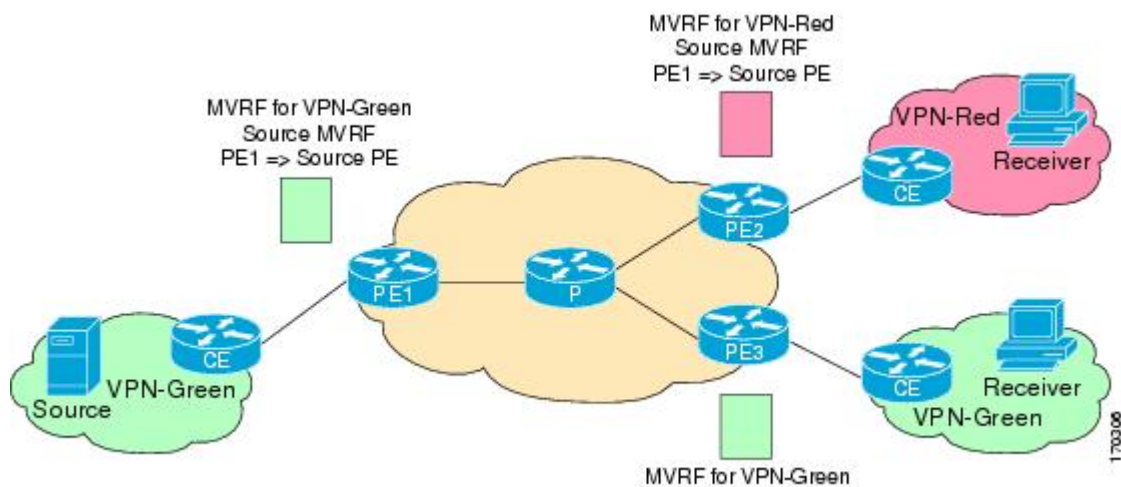
In order to provide support for mVPN extranet services, the same default MDT group must be configured in the source and receiver multicast VPN routing and forwarding (MVRF).

In the mVPN extranet support feature, the receiver and source MVRF multicast route (mroute) entries are linked. The Reverse Path Forwarding (RPF) check relies on unicast routing information to determine the interface through which the source is reachable. This interface is used as the RPF interface.

### Components of an mVPN Extranet

The figure below illustrates the components that constitute an mVPN extranet.

**Figure 25: Components of an mVPN Extranet**



- **MVRF:** An MVRF is a multicast-enabled VRF. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a provider edge (PE) router.
- **Source MVRF:** An MVRF that can reach the source through a directly connected customer edge (CE) router.
- **Receiver MVRF:** An MVRF to which receivers are connected through one or more CE devices.
- **Source PE:** A PE router that has a multicast source behind a directly connected CE router.
- **Receiver PE:** A PE router that has one or more interested receivers behind a directly connected CE router.

### Configuring mVPN Extranet Support

The following mVPN extranet service configuration options are available:

- Option 1: Configure the source MVRF on the receiver PE router.
- Option 2: Configure the receiver MVRF on the source PE router.

## mVPN Extranet Support Configuration - Option 1

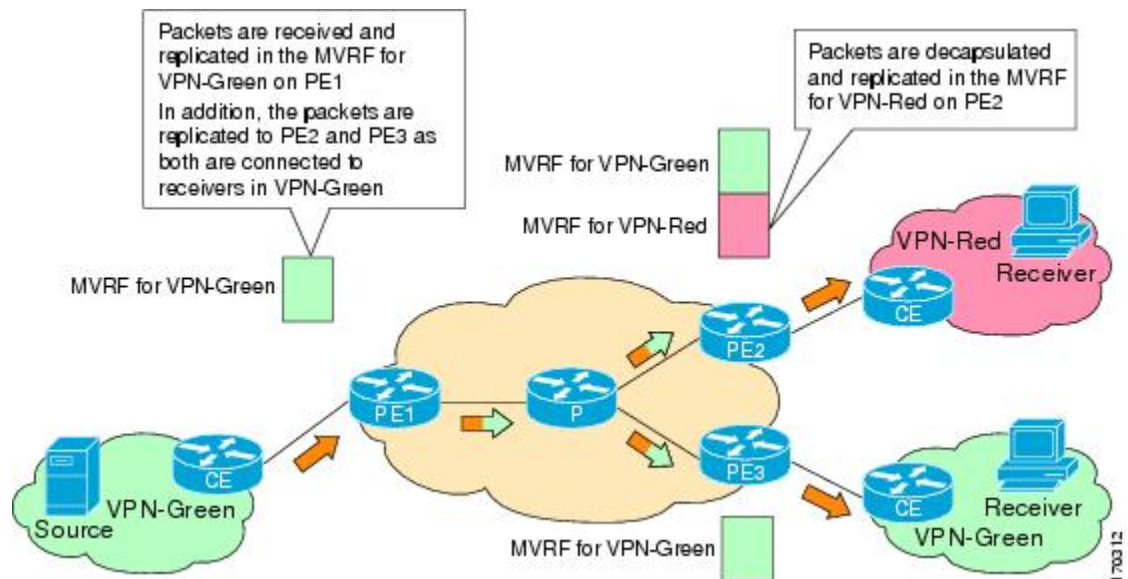
You can provide mVPN extranet services to enterprise VPN customers by configuring a source MVRF on a receiver PE router.

- On a receiver PE router that has one or more interested receivers in a extranet site behind a directly connected CE router, configure an additional MVRF that has the same default MDT group as the site connected to the multicast source, if the MVRF is not configured.
- Configure the same unicast routing policy to import routes from the source MVRF to the receiver MVRF.

The figure illustrates the flow of multicast traffic in an extranet mVPN topology where the source MVRF is configured on a receiver PE router (Option 1). In the topology, an MVRF is configured for VPN-Green and VPN-Red on PE2, a receiver PE router. A multicast source behind PE1, the source PE router, is sending out a multicast stream to the MVRF for VPN-Green. There are interested receivers behind PE2, the receiver PE router for VPN-Red, and behind PE3, the receiver PE router for VPN-Green. After PE1 receives the packets from the source in the MVRF for VPN-Green, it replicates and forwards the packets to PE2 and PE3, because both routers are connected to receivers in VPN-Green. The packets that originated from VPN-Green are then replicated on PE2 and forwarded to the interested receivers in VPN-Red and are replicated on PE3 and forwarded to the interested receivers in VPN-Green.

While configuring the source MVRF on the receiver PE router, the MDT group configuration of the source MVRF must be the same on both the source and receiver PE routers. In addition, you must configure the same unicast routing policy to import routes from the source MVRF (the MVRF for VPN-Green) to the receiver MVRF (the MVRF for VPN-Red).

**Figure 26: Packet Flow for mVPN Extranet Support Configuration Option 1**



## mVPN Extranet Support Configuration - Option 2

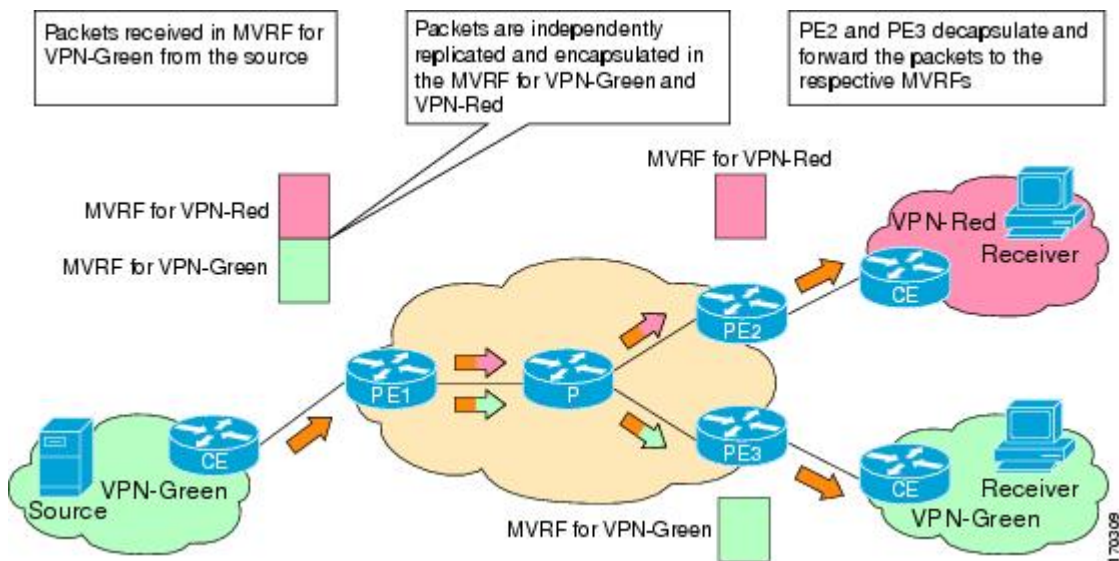
You can provide mVPN extranet services to enterprise VPN customers by configuring the receiver MVRF on the source PE router.

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

The figure illustrates the flow of multicast traffic in an mVPN extranet topology where a receiver MVRF is configured on the source PE router (Option 2). In the topology, an MVRF is configured for VPN-Green and VPN-Red on PE1, the source PE router. A multicast source behind PE1 is sending out a multicast stream to the MVRF for VPN-Green, and there are interested receivers behind PE2 and PE3, the receiver PE routers for VPN-Red and VPN-Green, respectively. After PE1 receives the packets from the source in the MVRF for VPN-Green, it independently replicates and encapsulates the packets in the MVRF for VPN-Green and VPN-Red and forwards the packets. After receiving the packets from this source, PE2 and PE3 decapsulate and forward the packets to the respective MVRFs.

While configuring the receiver MVRF on the source PE router, in the receiver MVRF configuration, the default MDT group must be the same on both the source and receiver PE routers. In addition, you must configure the same unicast routing policy to import routes from the source MVRF (the MVRF for VPN-Green) to the receiver MVRF (the MVRF for VPN-Red).

**Figure 27: Packet Flow for mVPN Extranet Support Configuration Option 2**



## RPF for mVPN Extranet Support Using Imported Routes

You must configure either the receiver MVRF on the source PE router (Option 1) or the source MVRF on the receiver PE router (Option 2) for extranet links to be created. Once configured, RPF relies on unicast routing information to determine the interface through which the source is reachable. This interface is used as the RPF interface. No additional configuration is required for RPF resolution. The mVPN extranet support

feature supports RPF from one VRF to another VRF, from a VRF to the global routing table, and from the global routing table to a VRF.

## RPF for mVPN Extranet Support Using Static Mroutes

By default, an mVPN extranet relies on unicast routing policies to determine the RPF interface. When the RPF lookup originates in a receiver MVRF, and it finds that the RPF interface does not lie in the same MVRF, the router uses the information in the Border Gateway Protocol (BGP) imported route to determine the source MVRF. The RPF lookup then continues and resolves in the source MVRF. In cases where the multicast and unicast topologies are incongruent, you can override the default behavior by configuring a static mroute in the receiver MVRF to explicitly specify the source MVRF using the **ip mroute** command with the **fallback-lookup** keyword and **vrf vrf-name** keyword and argument.

Static mroutes can also be configured to support RPF for mVPN extranet in the case where the source is present in an MVRF and the receiver is in the global table. In this case, because BGP does not allow VPNv4 routes to be imported into the IPv4 routing table, unicast cannot obtain the source MVRF information needed to resolve the RPF lookup. To enable the RPF lookup to be resolved in this case, a static mroute can be configured to explicitly specify the source MVRF using the **ip mroute** command with the **fallback-lookup** keyword and the **global** keyword.

## mVPN Extranet VRF Select

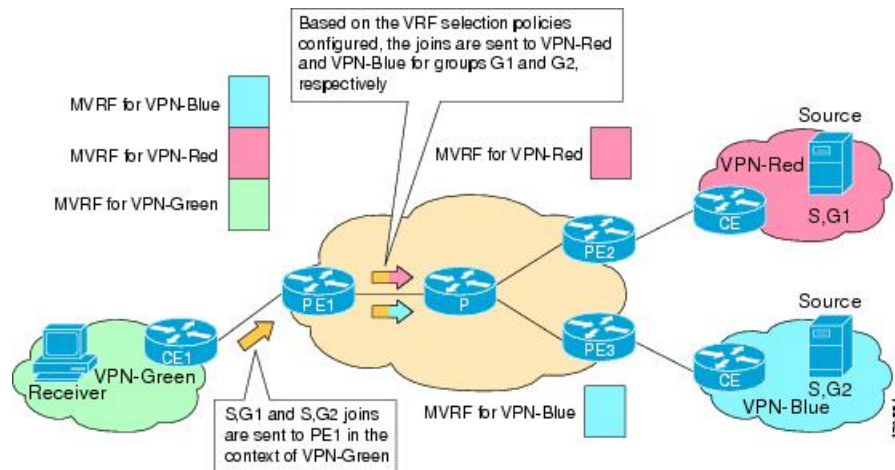
The mVPN extranet VRF Select feature provides the capability for RPF lookups to be performed to the same source address in different VRFs using the group address as the VRF selector. This feature enhances mVPN extranets by enabling service providers to distribute content streams coming in from different mVPNs and redistributing them from there.

The mVPN VRF Select feature is configured by creating group-based VRF selection policies. Group-based VRF selection policies are configured using the **ip multicast rpf select** command. The **ip multicast rpf select** command is used to configure RPF lookups originating in a receiver MVRF or in the global routing table to be resolved in a source MVRF or in the global routing table based on group address. Access Control Lists (ACLs) are used to define the groups to be applied to group-based VRF selection policies.

The figure illustrates an mVPN extranet topology with the mVPN VRF Select feature configured. In this topology, (S, G1) and (S, G2) PIM joins originating from VPN-Green, the receiver VRF, are forwarded to PE1, the receiver PE. Based on the group-based VRF selection policies configured, PE1 sends the PIM joins to VPN-Red and VPN-Blue for groups G1 and G2, respectively.



Figure 28: RPF Lookups Using Group-Based VRF Selection Policies



# How to Configure mVPN Extranet Support

## Configuring mVPN Support

Perform one of the following tasks to provide mVPN extranet capabilities in an IPv4 core network:

### Configuring the Source MVRF on the Receiver PE - Option 1

Perform this task to configure the source MVRF on the receiver PE router (Option 1) and provide support for mVPN extranet services.

#### Before you begin

You must configure Intranet VPN in the source and receiver VPNs prior to performing this task.

#### Procedure

|               | Command or Action                                                                     | Purpose                                                                                                             |
|---------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>enable</b><br><b>Example:</b><br><pre>Device&gt; enable</pre>                      | Enables privileged EXEC mode.<br><ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul> |
| <b>Step 2</b> | <b>configure terminal</b><br><b>Example:</b><br><pre>Device# configure terminal</pre> | Enters global configuration mode.                                                                                   |



|               | Command or Action                                                                                                                                | Purpose                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 3</b> | <b>vrf definition</b> <i>vrf-name</i><br><b>Example:</b><br><pre>Device(config)# vrf definition VPN-Red</pre>                                    | <p>Defines the VPN routing instance by assigning a VRF name and enters VRF configuration mode.</p> <ul style="list-style-type: none"> <li>The <i>vrf-name</i> argument is the name assigned to a VRF.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Step 4</b> | <b>rd</b> <i>route-distinguisher</i><br><b>Example:</b><br><pre>Device(config-vrf)# rd 55:1111</pre>                                             | <p>Creates routing and forwarding tables.</p> <ul style="list-style-type: none"> <li>The <i>route-distinguisher</i> argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter an RD in either of these formats: <ul style="list-style-type: none"> <li>16-bit autonomous system number: your 32-bit number, for example, 101:3</li> <li>32-bit IP address: your 16-bit number, for example, 192.168.122.15:1</li> </ul> </li> </ul>                                                                                                                                                                                                     |
| <b>Step 5</b> | <b>route-target import</b><br><i>route-target-ext-community</i><br><b>Example:</b><br><pre>Device(config-vrf)# route-target import 55:1111</pre> | <p>Creates a route-target extended community for a VRF.</p> <ul style="list-style-type: none"> <li>The <b>import</b> keyword exports routing information to the target VPN extended community.</li> <li>The <i>route-target-ext-community</i> argument adds the route-target extended community attributes to the VRF's list of import, export, or both (import and export) route-target extended communities.</li> </ul> <p><b>Note</b> For content to be distributed from the source MVRF to the receiver MVRF, you must configure the same unicast routing policy on the source and receiver PE routers to import routes from the source VRF to the receiver VRF.</p> |
| <b>Step 6</b> | <b>mdt default</b> <i>group-address</i><br><b>Example:</b><br><pre>Device(config-vrf)# mdt default 232.1.1.1</pre>                               | <p>Configures the multicast group address range for data MDT groups for a VRF.</p> <ul style="list-style-type: none"> <li>A tunnel interface is created as a result of this command.</li> <li>By default, the destination address of the tunnel header is the <i>group-address</i> argument.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                  |

|               | Command or Action                                                                                                                                                                                                                                                                        | Purpose                                                                                         |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| <b>Step 7</b> | <b>end</b><br><b>Example:</b><br><pre>Device(config-vrf)# end</pre>                                                                                                                                                                                                                      | Exits VRF configuration mode and returns to privileged EXEC mode.                               |
| <b>Step 8</b> | <b>show ip mroute [vrf vrf-name] group-address</b><br><b>Example:</b><br><pre>Device# show ip mroute 232.1.1.1</pre>                                                                                                                                                                     | (Optional) Displays the contents of the IP multicast mroute table for a specific group address. |
| <b>Step 9</b> | <b>show platform software fed switch {switch-number   active   standby }ip multicast groups [vrf-id vrf-id   vrf-name vrf-name] [group-address   count   summary]</b><br><b>Example:</b><br><pre>Device# show platform software fed switch active ip multicast groups 232.3.3.3/32</pre> | (Optional) Displays information related to the specified multicast group.                       |

## Configuring the Receiver MVRF on the Source PE - Option 2

Perform this task to configure the receiver MVRF on the source PE router (Option 2) and provide support for mVPN extranet services.

### Before you begin

You must configure Intranet VPN in the source and receiver VPNs prior to performing this task.

### Procedure

|               | Command or Action                                                                                      | Purpose                                                                                                            |
|---------------|--------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>enable</b><br><b>Example:</b><br><pre>Device&gt; enable</pre>                                       | Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul> |
| <b>Step 2</b> | <b>configure terminal</b><br><b>Example:</b><br><pre>Device# configure terminal</pre>                  | Enters global configuration mode.                                                                                  |
| <b>Step 3</b> | <b>vrf definition vrf-name</b><br><b>Example:</b><br><pre>Device(config)# vrf definition VPN-Red</pre> | Defines the VPN routing instance by assigning a VRF name and enters VRF configuration mode.                        |

|               | Command or Action                                                                                                                                | Purpose                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               |                                                                                                                                                  | <ul style="list-style-type: none"> <li>The <i>vrf-name</i> argument is the name assigned to a VRF.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>Step 4</b> | <b>rd</b> <i>route-distinguisher</i><br><b>Example:</b><br><pre>Device(config-vrf)# rd 55:2222</pre>                                             | <p>Creates routing and forwarding tables.</p> <ul style="list-style-type: none"> <li>Specify the <i>route-distinguisher</i> argument to add an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter an RD in either of these formats: <ul style="list-style-type: none"> <li>16-bit autonomous system number: your 32-bit number, for example, 101:3</li> <li>32-bit IP address: your 16-bit number, for example, 192.168.122.15:1</li> </ul> </li> </ul>                                                                                                                                                                                             |
| <b>Step 5</b> | <b>route-target import</b><br><i>route-target-ext-community</i><br><b>Example:</b><br><pre>Device(config-vrf)# route-target import 55:1111</pre> | <p>Creates a route-target extended community for a VRF.</p> <ul style="list-style-type: none"> <li>The <b>import</b> keyword imports routing information from the target VPN extended community.</li> <li>The <i>route-target-ext-community</i> argument adds the route-target extended community attributes to the VRF's list of import, export, or both (import and export) route-target extended communities.</li> </ul> <p><b>Note</b> For content to be distributed from the source MVRF to the receiver MVRF, you must configure the same unicast routing policy on the source and receiver PE routers to import routes from the source VRF to the receiver VRF.</p> |
| <b>Step 6</b> | <b>mdt default</b> <i>group-address</i><br><b>Example:</b><br><pre>Device(config-vrf)# mdt default 232.3.3.3</pre>                               | <p>Configures the multicast group address range for data MDT groups for a VRF.</p> <ul style="list-style-type: none"> <li>A tunnel interface is created as a result of this command.</li> <li>By default, the destination address of the tunnel header is the <i>group-address</i> argument.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Step 7</b> | <b>end</b><br><b>Example:</b><br><pre>Device(config-vrf)# end</pre>                                                                              | <p>Exits VRF configuration mode and returns to privileged EXEC mode.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

|               | Command or Action                                                                                                                | Purpose                                                                                         |
|---------------|----------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| <b>Step 8</b> | <b>show ip mroute</b> [ <i>vrf vrf-name</i> ]<br><i>group-address</i><br><b>Example:</b><br><br>Device# show ip mroute 232.3.3.3 | (Optional) Displays the contents of the IP multicast mroute table for a specific group address. |

## Configuring RPF for mVPN Extranet Support Using Static Mroutes

### Before you begin

You must configure support for mVPN extranet services prior to performing this task.

### Procedure

|               | Command or Action                                                                                                                                                                                                                                                                          | Purpose                                                                                                                                                                                                                                                                                                                                                                                                                               |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>enable</b><br><b>Example:</b><br><br>Device> enable                                                                                                                                                                                                                                     | Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>                                                                                                                                                                                                                                                                                                                      |
| <b>Step 2</b> | <b>configure terminal</b><br><b>Example:</b><br><br>Device# configure terminal                                                                                                                                                                                                             | Enters global configuration mode.                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Step 3</b> | <b>ip mroute vrf</b> <i>vrf-name</i> <i>source-address</i> <i>mask</i> <b>fallback-lookup</b> { <b>global</b>   <b>vrf</b> <i>vrf-name</i> } [ <i>distance</i> ]<br><b>Example:</b><br><br>Device(config)# ip mroute vrf VPN-Red 224.100.0.5 255.255.255.255 fallback-lookup vrf VPN-Green | Configures the RPF lookup originating in a receiver MVRF to continue and be resolved in a source MVRF or in the global routing table using a static mroute. <ul style="list-style-type: none"> <li>The <b>global</b> keyword is used to specify that the source MVRF is in the global routing table.</li> <li>The <b>vrf</b> keyword and <i>vrf-name</i> argument are used to explicitly specify a VRF as the source MVRF.</li> </ul> |
| <b>Step 4</b> | <b>end</b><br><b>Example:</b><br><br>Device(config)# end                                                                                                                                                                                                                                   | Exits global configuration mode and enters privileged EXEC mode.                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Step 5</b> | <b>show ip mroute</b> [ <i>vrf vrf-name</i> ]<br><i>group-address</i><br><b>Example:</b>                                                                                                                                                                                                   | (Optional) Displays the contents of the IP multicast mroute table for a specific group address.                                                                                                                                                                                                                                                                                                                                       |

|  | Command or Action                  | Purpose |
|--|------------------------------------|---------|
|  | Device# show ip mroute 224.100.0.5 |         |

## Configuring Group-Based VRF Selection Policies with mVPN Extranet

Perform this task to configure group-based VRF selection policies with mVPN.

This task enables RPF lookups to be performed to the same source address in different VRFs using the group address as the VRF selector.

### Before you begin

- You must configure support for mVPN extranet services prior to performing this task.
- You must configure the ACLs to be applied to group-based VRF selection policies.

### Procedure

|               | Command or Action                                                                                                                                                                                                         | Purpose                                                                                                                                                                       |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>enable</b><br><br><b>Example:</b><br><br>Device> enable                                                                                                                                                                | Enables privileged EXEC mode.<br><br>• Enter your password if prompted.                                                                                                       |
| <b>Step 2</b> | <b>configure terminal</b><br><br><b>Example:</b><br><br>Device# configure terminal                                                                                                                                        | Enters global configuration mode.                                                                                                                                             |
| <b>Step 3</b> | <b>ip multicast [vrf receiver-vrf-name] rpf select {global   vrf source-vrf-name} group-list access-list</b><br><br><b>Example:</b><br><br>Device(config)# ip multicast vrf VPN-Green rpf select vrf VPN-Red group-list 1 | • Configures RPF lookups originating in a receiver MVRF or in the global routing table to be resolved in a source MVRF or in the global routing table based on group address. |
| <b>Step 4</b> | Repeat step 3 to create additional group-based VRF selection policies.                                                                                                                                                    | --                                                                                                                                                                            |
| <b>Step 5</b> | <b>end</b><br><br><b>Example:</b><br><br>Device(config)# end                                                                                                                                                              | Exits global configuration mode and enters privileged EXEC mode.                                                                                                              |
| <b>Step 6</b> | <b>show ip} rpf [vrf vrf-name] select</b><br><br><b>Example:</b><br><br>Device# show ip rpf select                                                                                                                        | Displays group-to-VRF mapping information.                                                                                                                                    |



```

!
vrf definition VPN-Red
 rd 55:2222
 route-target export 55:2222
 route-target import 55:2222
 route-target import 55:1111
 mdt default 232.3.3.3
!
vrf definition VPN-Green
 rd 55:1111
 route-target export 55:1111
 route-target import 55:1111
 mdt default 232.1.1.1
!
ip multicast-routing
ip multicast-routing vrf VPN-Red
ip multicast-routing vrf VPN-Green
!
interface Loopback0
 ip address 10.2.0.2 255.255.255.0
 ip pim sparse-dense-mode
!
.
.
.
!
router bgp 55
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.1.0.1 remote-as 55
 neighbor 10.1.0.1 update-source Loopback0
!
 address-family ipv4 mdt
 neighbor 10.1.0.1 activate
 neighbor 10.1.0.1 send-community extended
!
 address-family vpnv4
 neighbor 10.1.0.1 activate
 neighbor 10.1.0.1 send-community extended
!

```

### PE1 Configuration

```

ip cef
!
vrf definition VPN-Green
 rd 55:1111
 route-target export 55:1111
 route-target import 55:1111
 mdt default 232.1.1.1
!
ip multicast-routing
ip multicast-routing vrf VPN-Green
!
interface Loopback0
 ip address 10.1.0.1 255.255.255.0
 ip pim sparse-dense-mode
!
.
.
.
!
router bgp 55

```

**Example: Configuring the Source VRF on the Receiver PE Router- Option 1**

```

no synchronization
bgp log-neighbor-changes
neighbor 10.2.0.2 remote-as 55
neighbor 10.2.0.2 update-source Loopback0
!
address-family ipv4 mdt
neighbor 10.2.0.2 activate
neighbor 10.2.0.2 send-community extended
!
address-family vpnv4
neighbor 10.2.0.2 activate
neighbor 10.2.0.2 send-community extended
!

```

**States in the Global Table on PE1 and PE2 for the MDT Default Group 232.1.1.1**

The following are sample outputs from the **show ip mroute** command on PE1 and PE2. The sample outputs show the global table for the MDT default group 232.1.1.1 on PE1 and PE2.

```

Device# show ip mroute 232.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
 L - Local, P - Pruned, R - RP-bit set, F - Register flag,
 T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
 X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
 U - URD, I - Received Source Specific Host Report,
 Z - Multicast Tunnel, z - MDT-data group sender,
 Y - Joined MDT-data group, y - Sending to MDT-data group,
 V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(10.2.0.2, 232.1.1.1), 00:01:19/00:02:42, flags: sTIZ
 Incoming interface: Ethernet0/0, RPF nbr 10.0.1.4
 Outgoing interface list:
 MVRF VPN-Green, Forward/Sparse-Dense, 00:01:19/00:02:07
(10.1.0.1, 232.1.1.1), 00:02:19/00:03:11, flags: sT
 Incoming interface: Loopback0, RPF nbr 0.0.0.0
 Outgoing interface list:
 Ethernet0/0, Forward/Sparse-Dense, 00:02:00/00:02:36
Device# show ip mroute 232.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
 L - Local, P - Pruned, R - RP-bit set, F - Register flag,
 T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
 X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
 U - URD, I - Received Source Specific Host Report,
 Z - Multicast Tunnel, z - MDT-data group sender,
 Y - Joined MDT-data group, y - Sending to MDT-data group,
 V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(10.1.0.1, 232.1.1.1), 00:02:04/00:02:38, flags: sTIZ
 Incoming interface: Ethernet1/0, RPF nbr 10.0.2.4
 Outgoing interface list:
 MVRF VPN-Green, Forward/Sparse-Dense, 00:02:04/00:02:09
(10.2.0.2, 232.1.1.1), 00:02:04/00:03:09, flags: sT
 Incoming interface: Loopback0, RPF nbr 0.0.0.0
 Outgoing interface list:
 Ethernet1/0, Forward/Sparse-Dense, 00:01:22/00:03:09

```



### States in the Global Table on PE1 and PE2 for the MDT Default Group 232.1.1.1 When PE1 and PE2 Are Switches Configured for mVPN Extranet Support

```

Device# show ip mroute 232.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
 L - Local, P - Pruned, R - RP-bit set, F - Register flag,
 T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
 X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
 U - URD, I - Received Source Specific Host Report,
 Z - Multicast Tunnel, z - MDT-data group sender,
 Y - Joined MDT-data group, y - Sending to MDT-data group,
 V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(10.2.0.2, 232.1.1.1), 00:01:19/00:02:42, flags: sTIZ
Incoming interface: GigabitEthernet2/16, RPF nbr 10.0.1.4, RPF-MFD
Outgoing interface list:
 MVRF VPN-Green, Forward/Sparse-Dense, 00:01:19/00:02:07, H
(10.1.0.1, 232.1.1.1), 00:02:19/00:03:11, flags: sT
Incoming interface: Loopback0, RPF nbr 0.0.0.0, RPF-MFD
Outgoing interface list:
 GigabitEthernet2/16, Forward/Sparse-Dense, 00:02:00/00:02:36, H
Device# show ip mroute 232.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
 L - Local, P - Pruned, R - RP-bit set, F - Register flag,
 T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
 X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
 U - URD, I - Received Source Specific Host Report,
 Z - Multicast Tunnel, z - MDT-data group sender,
 Y - Joined MDT-data group, y - Sending to MDT-data group,
 V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(10.1.0.1, 232.1.1.1), 00:02:04/00:02:38, flags: sTIZ
Incoming interface: GigabitEthernet4/1, RPF nbr 10.0.2.4, RPF-MFD
Outgoing interface list:
 MVRF VPN-Green, Forward/Sparse-Dense, 00:02:04/00:02:09, H
(10.2.0.2, 232.1.1.1), 00:02:04/00:03:09, flags: sT
Incoming interface: Loopback0, RPF nbr 0.0.0.0, RPF-MFD
Outgoing interface list:
 GigabitEthernet4/1, Forward/Sparse-Dense, 00:01:22/00:03:09, H

```

### States in the VRF Table for VPN-Green on PE1 After Receivers in VPN-Red Join Multicast Group 228.8.8.8

The following is sample output from the **show ip mroute** command on PE1. The sample output shows the state of the VRF table for VPN-Green on PE1 when receivers join the multicast group 228.8.8.8.

```

Device# show ip mroute vrf VPN-Green 228.8.8.8
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
 L - Local, P - Pruned, R - RP-bit set, F - Register flag,
 T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
 X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
 U - URD, I - Received Source Specific Host Report,
 Z - Multicast Tunnel, z - MDT-data group sender,
 Y - Joined MDT-data group, y - Sending to MDT-data group,
 V - RD & Vector, v - Vector

```

**Example: Configuring the Source VRF on the Receiver PE Router- Option 1**

```

Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 228.8.8.8), 00:01:43/00:02:52, RP 10.100.0.5, flags: S
 Incoming interface: Ethernet3/0, RPF nbr 10.1.1.5
 Outgoing interface list:
 Tunnel0, Forward/Sparse-Dense, 00:01:43/00:02:52
(10.1.1.200, 228.8.8.8), 00:01:15/00:03:26, flags: T
 Incoming interface: Ethernet3/0, RPF nbr 10.1.1.5
 Outgoing interface list:
 Tunnel0, Forward/Sparse-Dense, 00:01:15/00:03:19

```

**States in the VRF Table for VPN-Green on PE1 After Receivers in VPN-Red Join Multicast Group 228.8.8.8 When PE1 Is a Switch Configured for mVPN Extranet Support**

```

Device# show ip mroute vrf VPN-Green 228.8.8.8
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
 L - Local, P - Pruned, R - RP-bit set, F - Register flag,
 T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
 X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
 U - URD, I - Received Source Specific Host Report,
 Z - Multicast Tunnel, z - MDT-data group sender,
 Y - Joined MDT-data group, y - Sending to MDT-data group,
 V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 228.8.8.8), 00:01:43/00:02:52, RP 10.100.0.5, flags: S
 Incoming interface: GigabitEthernet3/1, RPF nbr 10.1.1.5, RPF-MFD
 Outgoing interface list:
 Tunnel0, Forward/Sparse-Dense, 00:01:43/00:02:52, H
(10.1.1.200, 228.8.8.8), 00:01:15/00:03:26, flags: T
 Incoming interface: GigabitEthernet3/1, RPF nbr 10.1.1.5, RPF-MFD
 Outgoing interface list:
 Tunnel0, Forward/Sparse-Dense, 00:01:15/00:03:19, H

```

**States in the VRF Table for VPN-Green on PE2 After Receivers in VPN-Red Join Multicast Group 228.8.8.8**

The following is sample output from the **show ip mroute** command on PE2. The output shows the state of the VRF table for VPN-Green on PE1 when receivers join the multicast group 228.8.8.8. The output indicates that extranet receivers in VPN-Red are receiving content from the source in VPN-Green that is sending to multicast group 228.8.8.8. The “E” flag indicates that a (\*, G) or (S, G) entry in the VRF routing table is a source VRF entry and has extranet receiver MVRF mroute entries linked to it.

```

Device# show ip mroute vrf VPN-Green 228.8.8.8
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
 L - Local, P - Pruned, R - RP-bit set, F - Register flag,
 T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
 X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
 U - URD, I - Received Source Specific Host Report,
 Z - Multicast Tunnel, z - MDT-data group sender,
 Y - Joined MDT-data group, y - Sending to MDT-data group,
 V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 228.8.8.8), 00:01:59/stopped, RP 10.100.0.5, flags: SE
 Incoming interface: Tunnel0, RPF nbr 10.1.0.1

```

```

Outgoing interface list: Null
Extranet receivers in vrf VPN-Red:
(*, 228.8.8.8), 00:01:59/stopped, RP 10.100.0.5, OIF count: 1, flags: S
(10.1.1.200, 228.8.8.8), 00:01:31/00:02:59, flags: TE
Incoming interface: Tunnel0, RPF nbr 10.1.0.1
Outgoing interface list: Null
Extranet receivers in vrf VPN-Red:
(10.1.1.200, 228.8.8.8), 00:01:31/00:03:29, OIF count: 1, flags:

```

### States in the VRF Table for VPN-Green on PE2 After Receivers in VPN-Red Join Multicast Group 228.8.8.8 When PE2 Is a Switch Configured for mVPN Extranet Support

```

Device# show ip mroute vrf VPN-Green 228.8.8.8
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
 L - Local, P - Pruned, R - RP-bit set, F - Register flag,
 T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
 X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
 U - URD, I - Received Source Specific Host Report,
 Z - Multicast Tunnel, z - MDT-data group sender,
 Y - Joined MDT-data group, y - Sending to MDT-data group,
 V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 228.8.8.8), 00:01:59/stopped, RP 10.100.0.5, flags: SE
Incoming interface: Tunnel0, RPF nbr 10.1.0.1, RPF-MFD
Outgoing interface list: Null
Extranet receivers in vrf VPN-Red:
(*, 228.8.8.8), 00:01:59/stopped, RP 10.100.0.5, OIF count: 1, flags: S
(10.1.1.200, 228.8.8.8), 00:01:31/00:02:59, flags: TE
Incoming interface: Tunnel0, RPF nbr 10.1.0.1, RPF-MFD
Outgoing interface list: Null
Extranet receivers in vrf VPN-Red:
(10.1.1.200, 228.8.8.8), 00:01:31/00:03:29, OIF count: 1, flags:

```

### States in the VRF Table for VPN-Red on PE2 After Receivers in VPN-Red Join Multicast Group 228.8.8.8

The following is sample output from the **show ip mroute** command on PE2. The sample output shows the state of the VRF table for VPN-Red on PE2 when receivers join the multicast group 228.8.8.8. The “using vrf VPN-Green” field indicates that VPN-Red is using unicast routing information from VPN-Green to determine the RPF interface through which the source is reachable.

```

Device# show ip mroute vrf VPN-Red 228.8.8.8
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
 L - Local, P - Pruned, R - RP-bit set, F - Register flag,
 T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
 X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
 U - URD, I - Received Source Specific Host Report,
 Z - Multicast Tunnel, z - MDT-data group sender,
 Y - Joined MDT-data group, y - Sending to MDT-data group,
 V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 228.8.8.8), 00:02:00/stopped, RP 10.100.0.5, flags: S
Incoming interface: Tunnel0, RPF nbr 10.1.0.1, using vrf VPN-Green
Outgoing interface list:

```

**Example: Configuring the Receiver VRF on the Source PE Router - Option 2**

```

Ethernet9/0, Forward/Sparse-Dense, 00:02:00/00:02:34
(10.1.1.200, 228.8.8.8), 00:01:32/00:03:28, flags:
Incoming interface: Tunnel0, RPF nbr 10.1.0.1, using vrf VPN-Green
Outgoing interface list:
Ethernet9/0, Forward/Sparse-Dense, 00:01:32/00:03:01

```

**States in the VRF Table for VPN-Red on PE2 After Receivers in VPN-Red Join Multicast Group 228.8.8.8 When PE2 Is a Switch Configured for mVPN Extranet Support**

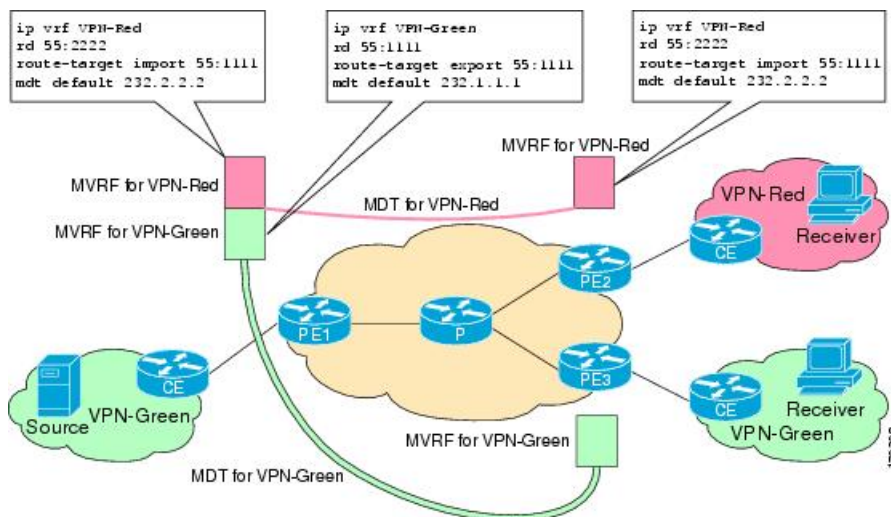
```

Device# show ip mroute vrf VPN-Red 228.8.8.8
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
 L - Local, P - Pruned, R - RP-bit set, F - Register flag,
 T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
 X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
 U - URD, I - Received Source Specific Host Report,
 Z - Multicast Tunnel, z - MDT-data group sender,
 Y - Joined MDT-data group, y - Sending to MDT-data group,
 V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 228.8.8.8), 00:02:00/stopped, RP 10.100.0.5, flags: S
Incoming interface: Tunnel0, RPF nbr 10.1.0.1, using vrf VPN-Green, RPF-MFD
Outgoing interface list:
GigabitEthernet9/1, Forward/Sparse-Dense, 00:02:00/00:02:34, H
(10.1.1.200, 228.8.8.8), 00:01:32/00:03:28, flags:
Incoming interface: Tunnel0, RPF nbr 10.1.0.1, using vrf VPN-Green, RPF-MFD
Outgoing interface list:
GigabitEthernet9/1, Forward/Sparse-Dense, 00:01:32/00:03:01, H

```

**Example: Configuring the Receiver VRF on the Source PE Router - Option 2**

The following example shows the configurations for PE1, the source PE router, and PE2, the receiver PE router, in the figure. In this example, mVPN extranet services are supported between VPN-Green and VPN-Red by configuring the receiver MVRF for VPN-Red on PE1, the source PE router. The MVRF configuration for VPN-Red is configured to import routes from the MVRF for VPN-Green to the MVRF for VPN-Red.

**Figure 30: Topology for mVPN Extranet Support Option 2 Configuration Example**

## PE1 Configuration

```
ip cef
!
vrf definition VPN-Green
 rd 55:1111
 route-target export 55:1111
 route-target import 55:1111
 mdt default 232.1.1.1
!
vrf definition VPN-Red
 rd 55:2222
 route-target export 55:2222
 route-target import 55:2222
 route-target import 55:1111
 mdt default 232.3.3.3
!
ip multicast-routing
ip multicast-routing vrf VPN-Green
ip multicast-routing vrf VPN-Red
!
interface Loopback0
 ip address 10.1.0.1 255.255.255.0
 ip pim sparse-dense-mode
!
.
.
.
!
router bgp 55
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.2.0.2 remote-as 55
 neighbor 10.2.0.2 update-source Loopback0
!
 address-family ipv4 mdt
 neighbor 10.2.0.2 activate
 neighbor 10.2.0.2 send-community extended
!
 address-family vpnv4
 neighbor 10.2.0.2 activate
 neighbor 10.2.0.2 send-community extended
!
```

## PE2 Configuration

```
!
vrf definition VPN-Red
 rd 55:2222
 route-target export 55:2222
 route-target import 55:2222
 route-target import 55:1111
 mdt default 232.3.3.3
!
ip multicast-routing
ip multicast-routing vrf VPN-Red
!
interface Loopback0
 ip address 10.2.0.2 255.255.255.0
 ip pim sparse-dense-mode
!
.
```

**Example: Configuring the Receiver VRF on the Source PE Router - Option 2**

```

.
.
!
router bgp 55
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.1.0.1 remote-as 55
 neighbor 10.1.0.1 update-source Loopback0
 !
 address-family ipv4 mdt
 neighbor 10.1.0.1 activate
 neighbor 10.1.0.1 send-community extended
 !
 address-family vpnv4
 neighbor 10.1.0.1 activate
 neighbor 10.1.0.1 send-community extended
 !

```

**States in the Global Table on PE1 and PE2 for the MDT Default Group 232.3.3.3**

The following are sample outputs from the **show ip mroute** command on PE1 and PE2. The sample outputs show the global table for the MDT default group 232.3.3.3 on PE1 and PE2.

PE1# **show ip mroute 232.3.3.3**

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,  
 L - Local, P - Pruned, R - RP-bit set, F - Register flag,  
 T - SPT-bit set, J - Join SPT, M - MSDP created entry,  
 X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,  
 U - URD, I - Received Source Specific Host Report,  
 Z - Multicast Tunnel, z - MDT-data group sender,  
 Y - Joined MDT-data group, y - Sending to MDT-data group  
 V - RD & Vector, v - Vector

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(10.1.0.1, 232.3.3.3), 00:46:27/00:03:27, flags: sT

Incoming interface: Loopback0, RPF nbr 0.0.0.0

Outgoing interface list:

Ethernet0/0, Forward/Sparse-Dense, 00:45:17/00:02:44

(10.2.0.2, 232.3.3.3), 00:45:17/00:02:57, flags: sTIZ

Incoming interface: Ethernet0/0, RPF nbr 224.0.1.4

Outgoing interface list:

MVRF VPN-Red, Forward/Sparse-Dense, 00:45:17/00:01:09

PE2# **show ip mroute 232.3.3.3**

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,  
 L - Local, P - Pruned, R - RP-bit set, F - Register flag,  
 T - SPT-bit set, J - Join SPT, M - MSDP created entry,  
 X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,  
 U - URD, I - Received Source Specific Host Report,  
 Z - Multicast Tunnel, z - MDT-data group sender,  
 Y - Joined MDT-data group, y - Sending to MDT-data group  
 V - RD & Vector, v - Vector

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(10.1.0.1, 232.3.3.3), 00:45:08/00:02:37, flags: sTIZ

Incoming interface: Ethernet1/0, RPF nbr 224.0.2.4

Outgoing interface list:

MVRF VPN-Red, Forward/Sparse-Dense, 00:45:08/00:01:27

(10.2.0.2, 232.3.3.3), 00:46:19/00:03:07, flags: sT

```
Incoming interface: Loopback0, RPF nbr 0.0.0.0
Outgoing interface list:
 Ethernet1/0, Forward/Sparse-Dense, 00:45:08/00:02:49
```

### States in the Global Table on PE1 and PE2 for the MDT Default Group 232.3.3.3 When PE1 and PE2 Are Switches Configured for mVPN Extranet Support

The following are sample outputs from the **show ip mroute** on PE1 and PE2, when PE1 and PE2 are switches that have been configured to support mVPN extranet services. The sample output from the **show ip mroute** command shows the global table for the MDT default group 232.3.3.3 on PE1 and PE2. In the output, the “RPF-MFD” flag indicates that a multicast flow is completely hardware switched and “H” flag indicates that the flow is being hardware switched on an outgoing interface.

```
Device# show ip mroute 232.3.3.3
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
 L - Local, P - Pruned, R - RP-bit set, F - Register flag,
 T - SPT-bit set, J - Join SPT, M - MSDP created entry,
 X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
 U - URD, I - Received Source Specific Host Report,
 Z - Multicast Tunnel, z - MDT-data group sender,
 Y - Joined MDT-data group, y - Sending to MDT-data group
 V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(10.1.0.1, 232.3.3.3), 00:46:27/00:03:27, flags: sT
 Incoming interface: Loopback0, RPF nbr 0.0.0.0, RPF-MFD
 Outgoing interface list:
 GigabitEthernet2/16, Forward/Sparse-Dense, 00:45:17/00:02:44, H
(10.2.0.2, 232.3.3.3), 00:45:17/00:02:57, flags: sTIZ
 Incoming interface: GigabitEthernet2/16, RPF nbr 224.0.1.4, RPF-MFD
 Outgoing interface list:
 MVRP VPN-Red, Forward/Sparse-Dense, 00:45:17/00:01:09, H
```

```
Device# show ip mroute 232.3.3.3
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
 L - Local, P - Pruned, R - RP-bit set, F - Register flag,
 T - SPT-bit set, J - Join SPT, M - MSDP created entry,
 X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
 U - URD, I - Received Source Specific Host Report,
 Z - Multicast Tunnel, z - MDT-data group sender,
 Y - Joined MDT-data group, y - Sending to MDT-data group
 V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(10.1.0.1, 232.3.3.3), 00:45:08/00:02:37, flags: sTIZ
 Incoming interface: GigabitEthernet4/1, RPF nbr 224.0.2.4, RPF-MFD
 Outgoing interface list:
 MVRP VPN-Red, Forward/Sparse-Dense, 00:45:08/00:01:27, H
(10.2.0.2, 232.3.3.3), 00:46:19/00:03:07, flags: sT
 Incoming interface: Loopback0, RPF nbr 0.0.0.0, RPF-MFD
 Outgoing interface list:
 GigabitEthernet4/1, Forward/Sparse-Dense, 00:45:08/00:02:49, H
```

**Example: Configuring the Receiver VRF on the Source PE Router - Option 2****States in the VRF Table for VPN-Green on PE1 After Receivers in VPN-Red Join Multicast Group 228.8.8.8**

The following is sample output from the **show ip mroute** command on PE1. The sample output shows the state of the VRF table for VPN-Green on PE1 when receivers join the multicast group 228.8.8.8. The output indicates that extranet receivers in VPN-Red are receiving content from a source in VPN-Green that is sending to multicast group 228.8.8.8. The “E” flag in the output indicates that a (\*, G) or (S, G) entry in the VRF routing table is a source VRF entry and has extranet receiver MVRF mroute entries linked to it.

```
Device# show ip mroute vrf VPN-Green 228.8.8.8
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
 L - Local, P - Pruned, R - RP-bit set, F - Register flag,
 T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
 X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
 U - URD, I - Received Source Specific Host Report,
 Z - Multicast Tunnel, z - MDT-data group sender,
 Y - Joined MDT-data group, y - Sending to MDT-data group,
 V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 228.8.8.8), 00:01:38/stopped, RP 10.100.0.5, flags: SE
 Incoming interface: Ethernet3/0, RPF nbr 10.1.1.5
 Outgoing interface list: Null
 Extranet receivers in vrf VPN-Red:
(*, 228.8.8.8), 00:01:38/stopped, RP 10.100.0.5, OIF count: 1, flags: S
(10.1.1.200, 228.8.8.8), 00:00:05/00:02:54, flags: TE
 Incoming interface: Ethernet3/0, RPF nbr 10.1.1.5
 Outgoing interface list: Null
 Extranet receivers in vrf VPN-Red:
(10.1.1.200, 228.8.8.8), 00:00:05/stopped, OIF count: 1, flags:
```

**States in the VRF Table for VPN-Green on PE1 After Receivers in VPN-Red Join Multicast Group 228.8.8.8 When PE1 Is a Switch Configured for mVPN Extranet Support**

The following are sample outputs from the **show ip mroute** on PE1, when PE1 is a Catalyst 6500 series switch configured to support extranet MVPN services. The sample output from the **show ip mroute** command shows the state of the VRF table for VPN-Green on PE1 when receivers join the multicast group 228.8.8.8. The sample output indicates that extranet receivers in VPN-Red are receiving content from a source in VPN-Green that is sending to multicast group 228.8.8.8.

```
Device# show ip mroute vrf VPN-Green 228.8.8.8
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
 L - Local, P - Pruned, R - RP-bit set, F - Register flag,
 T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
 X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
 U - URD, I - Received Source Specific Host Report,
 Z - Multicast Tunnel, z - MDT-data group sender,
 Y - Joined MDT-data group, y - Sending to MDT-data group,
 V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 228.8.8.8), 00:01:38/stopped, RP 10.100.0.5, flags: SE
 Incoming interface: GigabitEthernet3/1, RPF nbr 10.1.1.5, RPF-MFD
 Outgoing interface list: Null
 Extranet receivers in vrf VPN-Red:
(*, 228.8.8.8), 00:01:38/stopped, RP 10.100.0.5, OIF count: 1, flags: S
(10.1.1.200, 228.8.8.8), 00:00:05/00:02:54, flags: TE
```



```

Incoming interface: GigabitEthernet3/1, RPF nbr 10.1.1.5, RPF-MFD
Outgoing interface list: Null
Extranet receivers in vrf VPN-Red:
(10.1.1.200, 228.8.8.8), 00:00:05/stopped, OIF count: 1, flags:

```

### States in the VRF Table for VPN-Red on PE1 After Receivers in VPN-Red Join Multicast Group 228.8.8.8

The following is sample output from the **show ip mroute** command on PE1. The sample output shows the state of the VRF table for VPN-Red on PE1 when receivers join the multicast group 228.8.8.8. The “using vrf VPN-Green” field indicates that VPN-Red is using unicast routing information from VPN-Green to determine the RPF interface through which the source is reachable.

```

Device# show ip mroute vrf VPN-Red 228.8.8.8
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
 L - Local, P - Pruned, R - RP-bit set, F - Register flag,
 T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
 X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
 U - URD, I - Received Source Specific Host Report,
 Z - Multicast Tunnel, z - MDT-data group sender,
 Y - Joined MDT-data group, y - Sending to MDT-data group,
 V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 228.8.8.8), 00:01:45/stopped, RP 10.100.0.5, flags: S
 Incoming interface: Ethernet3/0, RPF nbr 10.1.1.5, using vrf VPN-Green
 Outgoing interface list:
 Tunnel2, Forward/Sparse-Dense, 00:01:45/00:02:49
(10.1.1.200, 228.8.8.8), 00:00:12/00:03:27, flags:
 Incoming interface: Ethernet3/0, RPF nbr 10.1.1.5, using vrf VPN-Green
 Outgoing interface list:
 Tunnel2, Forward/Sparse-Dense, 00:00:12/00:03:18

```

### States in the VRF Table for VPN-Red on PE1 After Receivers in VPN-Red Join Multicast Group 228.8.8.8 When PE1 Is a Switch Configured for mVPN Extranet Support

```

Device# show ip mroute vrf VPN-Red 228.8.8.8
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
 L - Local, P - Pruned, R - RP-bit set, F - Register flag,
 T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
 X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
 U - URD, I - Received Source Specific Host Report,
 Z - Multicast Tunnel, z - MDT-data group sender,
 Y - Joined MDT-data group, y - Sending to MDT-data group,
 V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 228.8.8.8), 00:01:45/stopped, RP 10.100.0.5, flags: S
 Incoming interface: GigabitEthernet3/1, RPF nbr 10.1.1.5, using vrf VPN-Green, RPF-MFD
 Outgoing interface list:
 Tunnel2, Forward/Sparse-Dense, 00:01:45/00:02:49, H
(10.1.1.200, 228.8.8.8), 00:00:12/00:03:27, flags:
 Incoming interface: GigabitEthernet3/1, RPF nbr 10.1.1.5, using vrf VPN-Green, RPF-MFD
 Outgoing interface list:
 Tunnel2, Forward/Sparse-Dense, 00:00:12/00:03:18, H

```

**Example: Displaying Statistics for mVPN Extranet Support****States in the VRF Table for VPN-Red on PE2 After Receivers in VPN-Red Join Multicast Group 228.8.8.8**

The following is sample output from the **show ip mroute** command on PE2. The sample output shows the VRF table for VPN-Red on PE2 when receivers join the multicast group 228.8.8.8.

```
PE2# show ip mroute vrf VPN-Red 228.8.8.8
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
 L - Local, P - Pruned, R - RP-bit set, F - Register flag,
 T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
 X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
 U - URD, I - Received Source Specific Host Report,
 Z - Multicast Tunnel, z - MDT-data group sender,
 Y - Joined MDT-data group, y - Sending to MDT-data group,
 V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 228.8.8.8), 00:00:28/stopped, RP 10.100.0.5, flags: S
 Incoming interface: Tunnell, RPF nbr 10.1.0.1
 Outgoing interface list:
 Ethernet9/0, Forward/Sparse-Dense, 00:00:28/00:03:02
(10.1.1.200, 228.8.8.8), 00:00:00/00:03:29, flags:
 Incoming interface: Tunnell, RPF nbr 10.1.0.1
 Outgoing interface list:
 Ethernet9/0, Forward/Sparse-Dense, 00:00:00/00:03:29
```

**States in the VRF Table for VPN-Red on PE2 After Receivers in VPN-Red Join Multicast Group 228.8.8.8 When PE2 Is a Switch Configured for mVPN Extranet Support**

```
PE2# show ip mroute vrf VPN-Red 228.8.8.8
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
 L - Local, P - Pruned, R - RP-bit set, F - Register flag,
 T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
 X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
 U - URD, I - Received Source Specific Host Report,
 Z - Multicast Tunnel, z - MDT-data group sender,
 Y - Joined MDT-data group, y - Sending to MDT-data group,
 V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 228.8.8.8), 00:00:28/stopped, RP 10.100.0.5, flags: S
 Incoming interface: Tunnell, RPF nbr 10.1.0.1, RPF-MFD
 Outgoing interface list:
 GigabitEthernet9/1, Forward/Sparse-Dense, 00:00:28/00:03:02, H
(10.1.1.200, 228.8.8.8), 00:00:00/00:03:29, flags:
 Incoming interface: Tunnell, RPF nbr 10.1.0.1, RPF-MFD
 Outgoing interface list:
 GigabitEthernet9/1, Forward/Sparse-Dense, 00:00:00/00:03:29, H
```

**Example: Displaying Statistics for mVPN Extranet Support**

This example is a stand alone example and does not refer to any other technologies.

The MFIB-based implementation of IP multicast updates counters in source MVRF mroute entries for mVPN extranet. Counters in the source MVRF can be displayed using Cisco IOS commands. Counters in the receiver MVRF mroute entries will remain zero.

Use the **show ip mroute** command to determine the source and receiver MVRFs. The following sample output shows that VRF blue is the source MVRF and VRF red is the receiver MVRF:

```
Device# show ip mroute vrf blue 228.1.1.1

IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
 L - Local, P - Pruned, R - RP-bit set, F - Register flag,
 T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
 X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
 U - URD, I - Received Source Specific Host Report,
 Z - Multicast Tunnel, z - MDT-data group sender,
 Y - Joined MDT-data group, y - Sending to MDT-data group,
 V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 228.1.1.1), 00:05:48/stopped, RP 202.100.0.5, flags: SE
 Incoming interface: Ethernet3/0, RPF nbr 200.1.1.5
 Outgoing interface list: Null
 Extranet receivers in vrf red:
(*, 228.1.1.1), 00:05:48/stopped, RP 202.100.0.5, OIF count: 1, flags: S
(220.1.1.200, 228.1.1.1), 00:02:42/00:02:09, flags: TE
 Incoming interface: Ethernet3/0, RPF nbr 200.1.1.5
 Outgoing interface list: Null
 Extranet receivers in vrf red:
(220.1.1.200, 228.1.1.1), 00:02:42/stopped, OIF count: 1, flags: T
```

```
Device# show ip mroute vrf red 228.1.1.1

IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
 L - Local, P - Pruned, R - RP-bit set, F - Register flag,
 T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
 X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
 U - URD, I - Received Source Specific Host Report,
 Z - Multicast Tunnel, z - MDT-data group sender,
 Y - Joined MDT-data group, y - Sending to MDT-data group,
 V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 228.1.1.1), 00:05:55/stopped, RP 202.100.0.5, flags: S
 Incoming interface: Ethernet3/0, RPF nbr 200.1.1.5, using vrf blue
 Outgoing interface list:
 Tunnel16, Forward/Sparse-Dense, 00:05:55/00:03:26
(220.1.1.200, 228.1.1.1), 00:02:49/stopped, flags: T
 Incoming interface: Ethernet3/0, RPF nbr 200.1.1.5, using vrf blue
 Outgoing interface list:
 Tunnel16, Forward/Sparse-Dense, 00:02:49/00:03:26
```

Use the **show ip mfib vrf vrf-name** command, with the source MVRF for the *vrf-name* argument, to display statistics.

The following example shows statistics for the source MVRF blue. Inspect the output to ensure that the forwarding statistics in the source MVRF MFIB are correct and that the A and F flags are set in the source MVRF. Notice that there is no indication of extranet forwarding in the MFIB.

```
Device# show ip mfib vrf blue 228.1.1.1

Entry Flags: C - Directly Connected, S - Signal, IA - Inherit A
flag,
 ET - Data Rate Exceeds Threshold, K - Keepalive
```

### Example: Displaying Statistics for mVPN Extranet Support

```

DDE - Data Driven Event, HW - Hardware Installed
I/O Item Flags: IC - Internal Copy, NP - Not platform switched,
NS - Negate Signalling, SP - Signal Present,
A - Accept, F - Forward, RA - MRIB Accept, RF - MRIB
Forward,
MA - MFIB Accept
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kbits per
second
Other counts: Total/RPF failed/Other drops
I/O Item Counts: FS Pkt Count/PS Pkt Count
VRF blue
(*,228.1.1.1) Flags: C
SW Forwarding: 1/0/100/0, Other: 0/0/0
Ethernet3/0 Flags: A
Tunnel16, MDT/239.3.3.3 Flags: F
Pkts: 1/0
(220.1.1.200,228.1.1.1) Flags:
SW Forwarding: 37/0/100/0, Other: 0/0/0
Ethernet3/0 Flags: A NS
Tunnel16, MDT/239.3.3.3 Flags: F
Pkts: 37/0

```

The following example shows the following information for the receiver MVRF red:

- There are no forwarding statistics in the receiver MVRF MFIB because these statistics are collected in the source MVRF.
- The A and F flags are not set because these flags are only set in the source MVRF for mVPN extranet.
- There is no indication of extranet forwarding in the MFIB.



**Note** The NS flag in the output is present for the purpose of receiving PIM control traffic in the receiver MVRF.

```

Device# show ip mfib vrf red 228.1.1.1

Entry Flags: C - Directly Connected, S - Signal, IA - Inherit A
flag,
ET - Data Rate Exceeds Threshold, K - Keepalive
DDE - Data Driven Event, HW - Hardware Installed
I/O Item Flags: IC - Internal Copy, NP - Not platform switched,
NS - Negate Signalling, SP - Signal Present,
A - Accept, F - Forward, RA - MRIB Accept, RF - MRIB
Forward,
MA - MFIB Accept
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kbits per
second
Other counts: Total/RPF failed/Other drops
I/O Item Counts: FS Pkt Count/PS Pkt Count
VRF red
(*,228.1.1.1) Flags: C
SW Forwarding: 0/0/0/0, Other: 0/0/0
Tunnel16, MDT/239.3.3.3 Flags: NS
(220.1.1.200,228.1.1.1) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
Tunnel16, MDT/239.3.3.3 Flags: NS

```

You can also use the **show ip mroute count** command to display the mVPN extranet statistics. However, we recommend that you use the **show ip mfib** command instead. If you use the **show ip mroute count** command to display statistics, inspect the output to ensure that the forwarding statistics in the source MVRF are correct and that there are no forwarding statistics in the receiver MVRF.

The following sample output from the **show ip mroute count** command shows statistics for the source MVRF blue:

```
Device# show ip mroute vrf blue 228.1.1.1 count
```

Use "show ip mfib count" to get better response time for a large number of mroutes.

```
IP Multicast Statistics
3 routes using 1354 bytes of memory
2 groups, 0.50 average sources per group
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per second
Other counts: Total/RPF failed/Other drops(OIF-null, rate-limit etc)
Group: 228.1.1.1, Source count: 1, Packets forwarded: 38, Packets received: 38
 RP-tree: Forwarding: 1/0/100/0, Other: 1/0/0
 Source: 220.1.1.200/32, Forwarding: 37/0/100/0, Other: 37/0/0
```

The following sample output from the **show ip mroute count** command is for the receiver MVRF red:

```
Device# show ip mroute vrf red 228.1.1.1 count
```

Use "show ip mfib count" to get better response time for a large number of mroutes.

```
IP Multicast Statistics
3 routes using 1672 bytes of memory
2 groups, 0.50 average sources per group
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per second
Other counts: Total/RPF failed/Other drops(OIF-null, rate-limit etc)
Group: 228.1.1.1, Source count: 1, Packets forwarded: 0, Packets received: 0
 RP-tree: Forwarding: 0/0/0/0, Other: 0/0/0
 Source: 220.1.1.200/32, Forwarding: 0/0/0/0, Other: 0/0/0
```

## Example: Configuring RPF for mVPN Extranet Support Using Static Mroutes

The following example shows how to configure the RPF lookup originating in VPN-Red to be resolved in VPN-Green using the static mroute 192.168.1.1:

```
ip mroute vrf VPN-Red 192.168.1.1 255.255.255.255 fallback-lookup vrf VPN-Green
```

## Example: Configuring Group-Based VRF Selection Policies with mVPN Extranet Support

The following example shows how to use group-based VRF selection policies to configure RPF lookups originating in VPN-Green to be performed in VPN-Red for group addresses that match ACL 1 and to be performed in VPN-Blue for group addresses that match ACL 2.

```
ip multicast vrf VPN-Green rpf select vrf VPN-Red group-list 1
ip multicast vrf VPN-Green rpf select vrf VPN-Blue group-list 2
!
```

```

.
.
.
!
access-list 1 permit 239.0.0.0 0.255.255.255
access-list 2 permit 238.0.0.0 0.255.255.255
!

```

## Additional References

### Related Documents

| Related Topic                                                  | Document Title                                    |
|----------------------------------------------------------------|---------------------------------------------------|
| Basic IP multicast concepts, configuration tasks, and examples | “Configuring Basic IP Multicast Routing” module   |
| IP multicast overview                                          | “IP Multicast Routing Technology Overview” module |
| MPLS Layer 3 VPN concepts and configuration tasks              | “Configuring MPLS Layer 3 VPN” module             |
| Multicast VPN concepts, configuration tasks, and examples      | “Configuring Multicast VPN” module                |

## Feature History for mVPN Extranet Support

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

| Release                       | Feature               | Feature Information                                                                                                                                                                                                                                                                                                                                  |
|-------------------------------|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cisco IOS XE Amsterdam 17.1.1 | mVPN Extranet Support | The mVPN extranet support feature enables service providers to distribute IP multicast content originated from one enterprise site to other enterprise sites. This feature enables service providers to offer the next generation of flexible extranet services, helping to enable business partnerships between different enterprise VPN customers. |

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <https://cfnng.cisco.com/>



## CHAPTER 17

# MLDP-Based MVPN

---

- [MLDP-Based MVPN, on page 337](#)
- [Prerequisites for MLDP-Based MVPN, on page 337](#)
- [Restrictions for MLDP-Based VPN, on page 337](#)
- [Information About MLDP-Based MVPN, on page 338](#)
- [Overview of MVPN MLDP Partitioned MDT, on page 348](#)
- [Supported MLDP Profiles, on page 349](#)
- [How to Configure MLDP-Based MVPN, on page 350](#)
- [Configuration Examples for MLDP-Based MVPN, on page 354](#)
- [Feature History for MLDP-Based MVPN, on page 364](#)

## MLDP-Based MVPN

The MLDP-based MVPN feature provides extensions to Label Distribution Protocol (LDP) for the setup of point-to-multipoint (P2MP) and multipoint-to-multipoint (MP2MP) label switched paths (LSPs) for transport in the Multicast Virtual Private Network (MVPN) core network.

## Prerequisites for MLDP-Based MVPN

- You must be familiar with IPv4 multicast routing configuration tasks and concepts.
- Cisco Express Forwarding (CEF) must be enabled on the router for label switching.
- Unicast routing must be operational.
- To enable MLDP-based multicast VPN, you must configure a VPN routing and forwarding (VRF) instance.

## Restrictions for MLDP-Based VPN

- Only MLDP profiles 1, 13, and 14 are supported.
- MLDP extranet is not supported.
- GRE tunnel in core is not supported for MLDP.

- MLDP FRR is not supported.
- Supported content group modes are Protocol Independent Multicast (PIM) sparse mode (PIM-SM), and Source Specific Multicast (SSM). Bidirectional PIM (PIM-Bidir) traffic is supported only on Profile 1.
- PIM dense mode (PIM-DM) is not supported.
- RSVP-TE-based LSM is not supported.
- The PIM-sparse content group mode is supported if the RP is configured behind the PE router (on CE) or on the source PE router.
- IGP MLDP ECMP is not supported. You must configure **no mpls mldp forwarding recursive** to use MLDP multipath.
- Dual Homing of L2 PEs are not supported for any MVPN profiles.
- MLDP in Seamless MPLS architecture is not supported.

## Information About MLDP-Based MVPN

### Overview of MLDP-Based MVPN

MVPN allows a service provider to configure and support multicast traffic in an MPLS VPN environment. This feature supports routing and forwarding of multicast packets for each individual VPN routing and forwarding (VRF) instance, and it also provides a mechanism to transport VPN multicast packets across the service provider backbone.

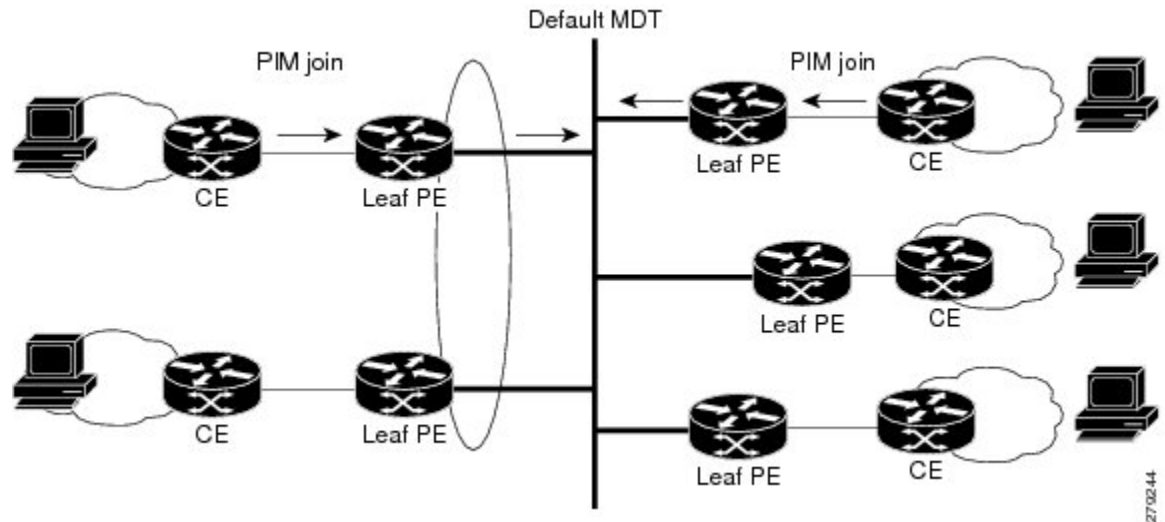
A VPN is network connectivity across a shared infrastructure, such as an Internet service provider (ISP). Its function is to provide the same policies and performance as a private network, at a reduced cost of ownership, thus creating many opportunities for cost savings through operations and infrastructure.

An MVPN allows an enterprise to transparently interconnect its private network across the network backbone of a service provider. The use of an MVPN to interconnect an enterprise network in this way does not change the way that the enterprise network is administered, nor does it change general enterprise connectivity.

As shown in the figure, in an MLDP-based MVPN, a static default multicast distribution tree (MDT) is established for each multicast domain. The default MDT defines the path used by provider edge (PE) devices to send multicast data and control messages to every other PE device in the multicast domain. A default MDT is created in the core network using a single MP2MP LSP. The default MDT behaves like a virtual LAN.

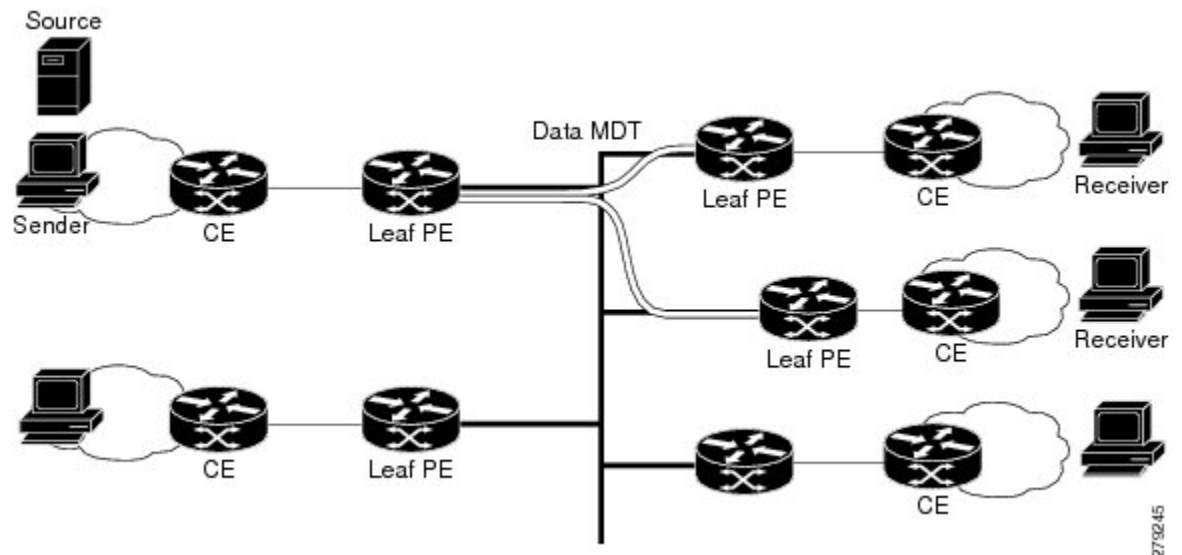


Figure 31: MLDP with the Default MDT Scenario



As shown in the figure, an MLDP-based MVPN also supports the dynamic creation of data MDTs for high-bandwidth transmission. For high-rate data sources, a data MDT is created using P2MP LSPs to off-load traffic from the default MDT to avoid unnecessary waste of bandwidth to PEs that did not join the stream. The creation of the data MDT is signaled dynamically using MDT Join TLV messages. Data MDTs are a feature unique to Cisco IOS software. Data MDTs are intended for high-bandwidth sources such as full-motion video inside the VPN to ensure optimal traffic forwarding in the MPLS VPN core. The threshold at which the data MDT is created can be configured on a per-device or a per-VRF basis. When the multicast transmission exceeds the defined threshold, the sending PE device creates the data MDT and sends a User Datagram Protocol (UDP) message, which contains information about the data MDT to all devices on the default MDT.

Figure 32: MLDP with the Data MDT Scenario



Data MDTs are created only for (S, G) multicast route entries within the VRF multicast routing table. They are not created for (\*, G) entries regardless of the value of the individual source data rate.

The only transport mechanism previously available was Protocol Independent Multicast (PIM) with Multipoint Generic Routing Encapsulation (mGRE) over an IP core network. The introduction of Multicast Label Distribution Protocol (MLDP) provides transport by using MLDP with label encapsulation over an MPLS core network.

MLDP creates the MDTs as follows:

- The default MDT uses MP2MP LSPs.
  - Supports low bandwidth and control traffic between VRFs.
- The data MDT uses P2MP LSPs.
  - Supports a single high-bandwidth source stream from a VRF.

All other operations of MVPN remain the same regardless of the tunneling mechanism:

- PIM neighbors in a VRF are seen across a Label Switched Path virtual interface (LSP-VIF).
- The VPN multicast state is signaled by PIM.

The only other difference when using MLDP is that the MDT group address used in the mGRE solution is replaced with a VPN ID.

MLDP-based MVPN provides the following benefits:

- Enables the use of a single MPLS forwarding plane for both unicast and multicast traffic.
- Enables existing MPLS protection (for example, MPLS Traffic Engineering/Resource Reservation Protocol (TE/RSVP link protection) and MPLS Operations Administration and Maintenance (OAM) mechanisms to be used for multicast traffic.
- Reduces operational complexity due to the elimination of the need for PIM in the MPLS core network.

## Initial Deployment of an MLDP-Based MVPN

Initial deployment of an MLDP-based MVPN involves the configuration of a default MDT and one or more data MDTs.

A static default MDT is established for each multicast domain. The default MDT defines the path used by PE devices to send multicast data and control messages to every other PE device in the multicast domain. A default MDT is created in the core network using a single MP2MP LSP.

An MLDP-based MVPN also supports the dynamic creation of data MDTs for high-bandwidth transmission.

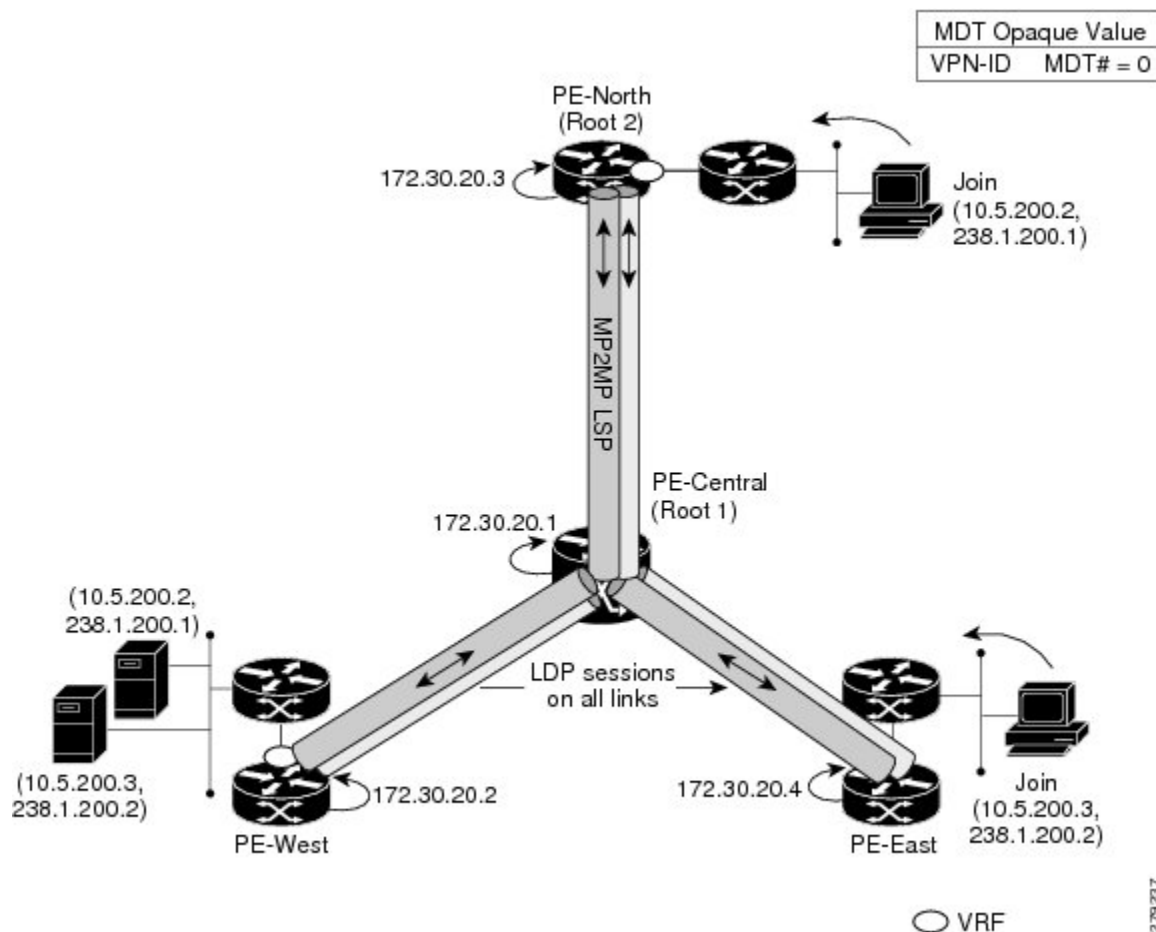
### Default MDT Creation

The figure shows the default MDT scenario. The Opaque value used to signal a default MDT consists of two parameters: the VPN ID and the MDT number for the VPN in the format (vpn-id, 0) where vpn-id is a manually configured 7-byte number that uniquely identifies this VPN. The default MDT is set to zero.

In this scenario, each of the three PE devices belong to the VRF called VRF and they have the same VPN ID. Each PE device with the same VPN ID will join the same MP2MP tree. The PE devices have created a primary MP2MP tree rooted at P-Central (Root 1) and a backup MP2MP tree rooted at PE-North (Root 2). There are two sources at PE-West and interested receivers at both PE-North and PE-East. PE-West will choose one of

the MP2MP trees to transmit the customer VPN traffic, but all PE devices can receive traffic on either of the MP2MP trees.

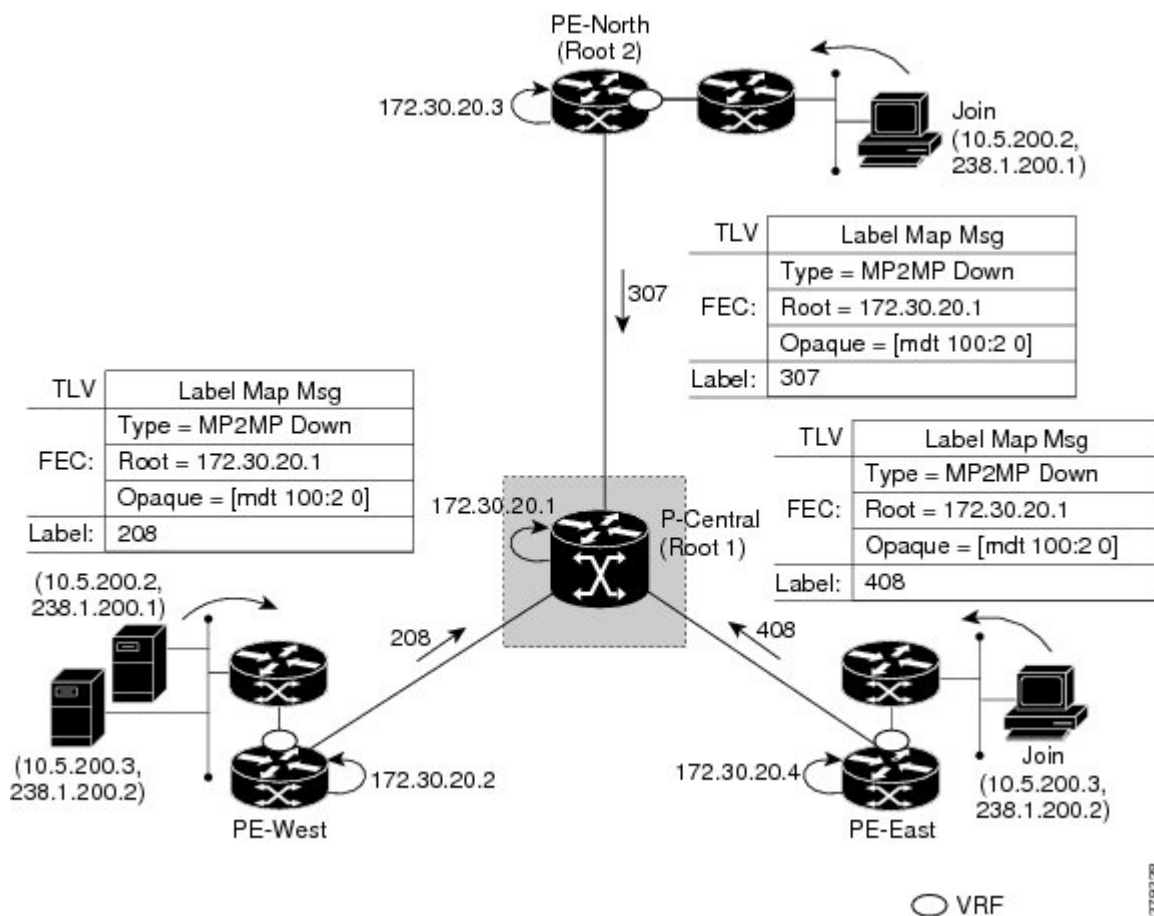
**Figure 33: Default MDT Scenario**



### LSP Downstream Default MDT Creation

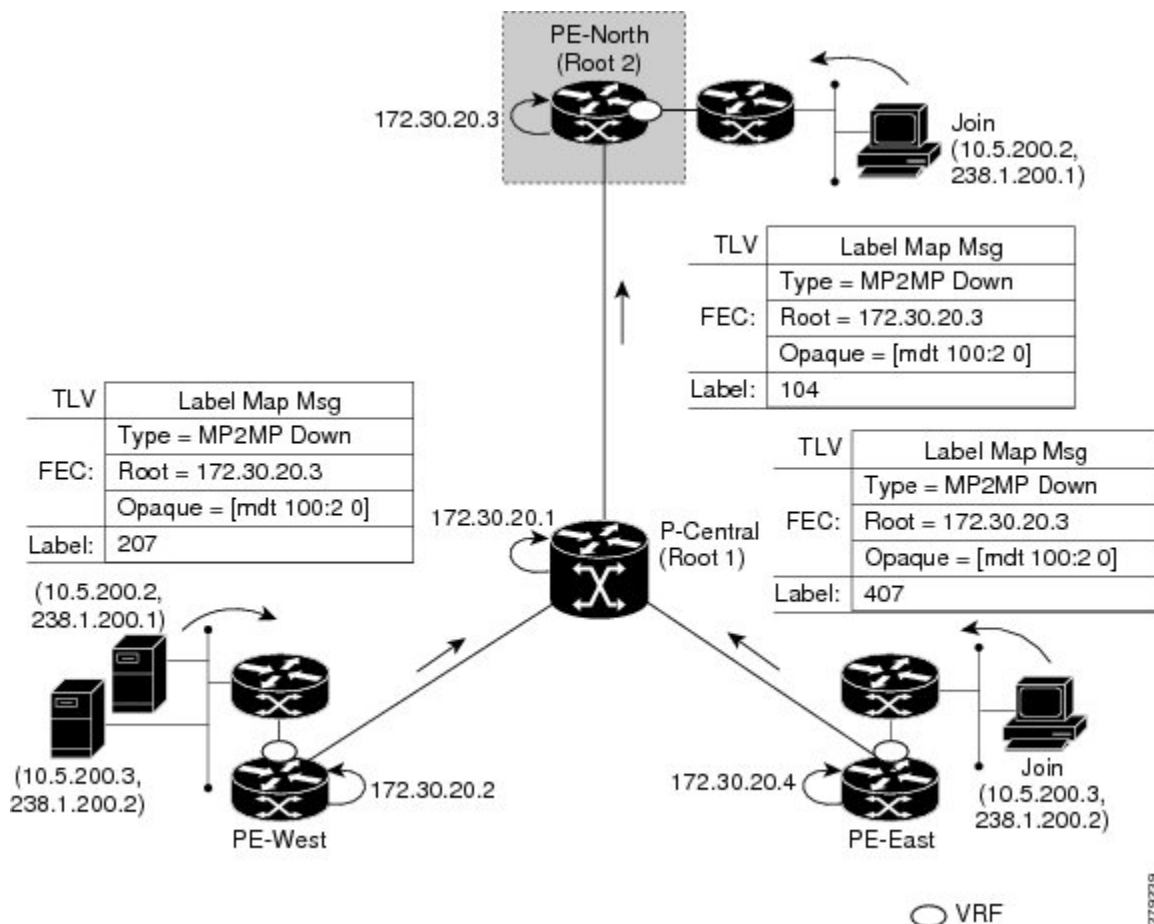
The figures show the downstream tree creation for each of the roots. Each PE device configured with VPN ID 100:2 creates the same Forwarding Equivalence Class (FEC) Type Length Value (TLV), but with a different root and downstream labels per MP2MP tree. The FEC type will be MP2MP Down, which prompts the receiving Label Switched Route (LSR) to respond with an upstream label mapping message to create the upstream path.

Figure 34: Default MDT Downstream--Root 1



2750238

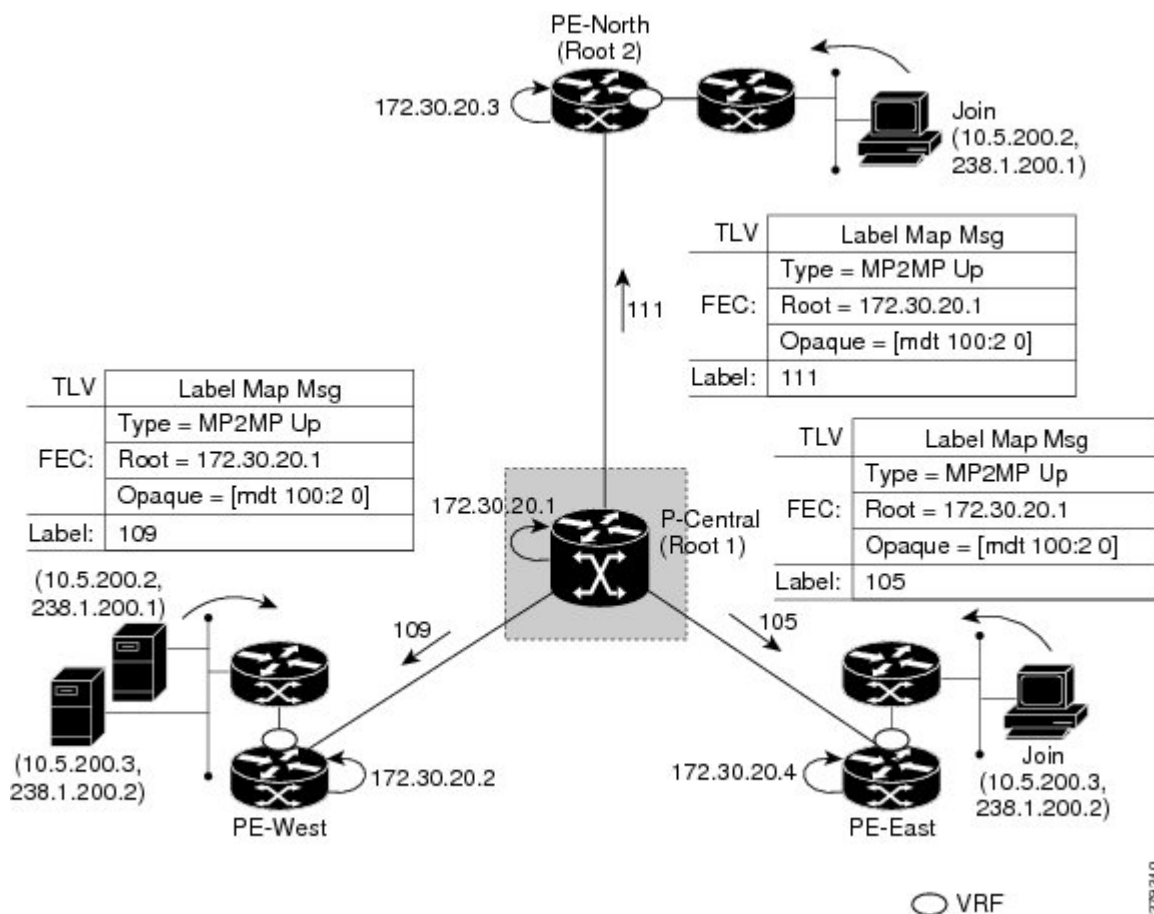
Figure 35: Default MDT Downstream--Root 2



### LSP Upstream Default MDT Creation

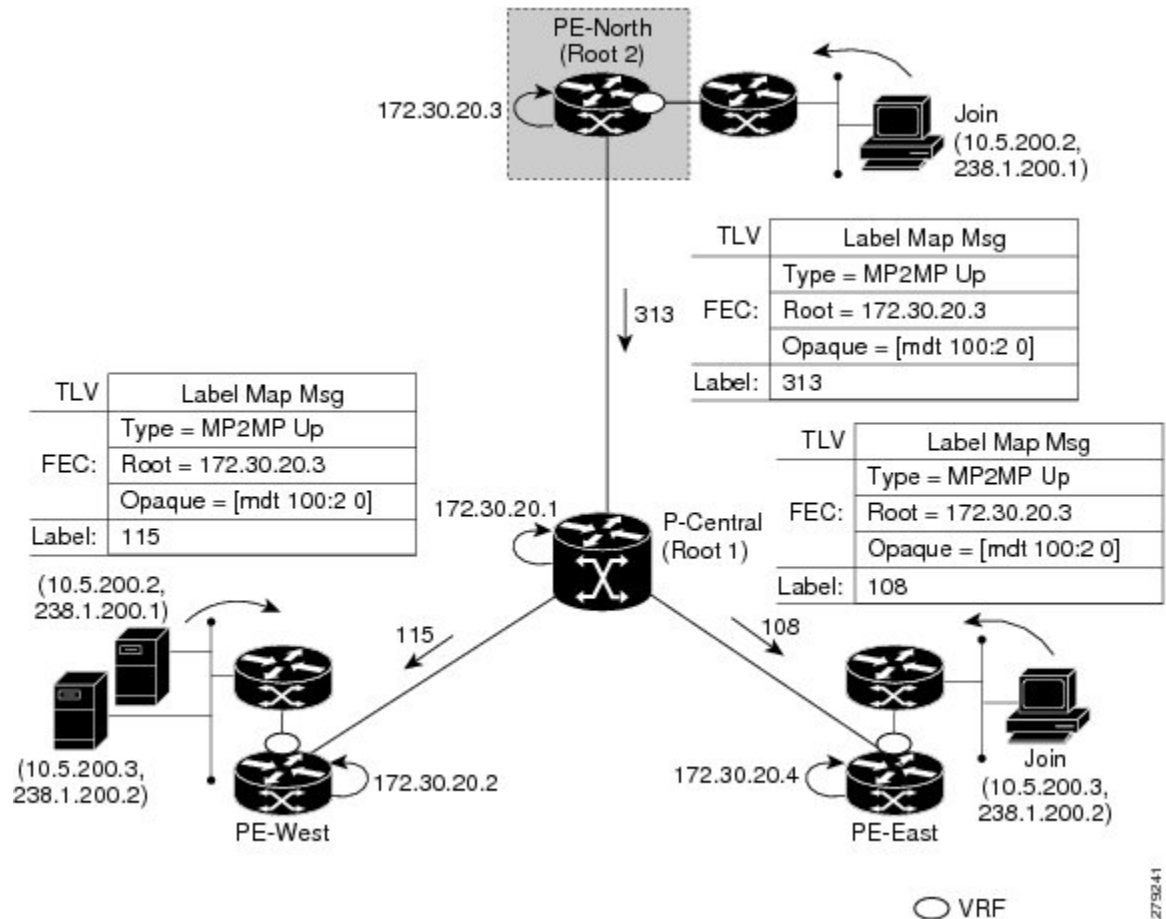
The figures show the upstream LSP creation for the default MDTs. For each downstream label received, a corresponding upstream label is sent. In the first figure, P-Central sends out three upstream labels (111, 109, and 105) to each downstream directly connected neighbor (downstream is away from the root). The process for PE-North is the same except that it only sends a single upstream label (313) as there is only one directly connected downstream neighbor, as shown in the second figure.

Figure 36: Default MDT Upstream--Root 1



275340

Figure 37: Default MDT Upstream--Root 2

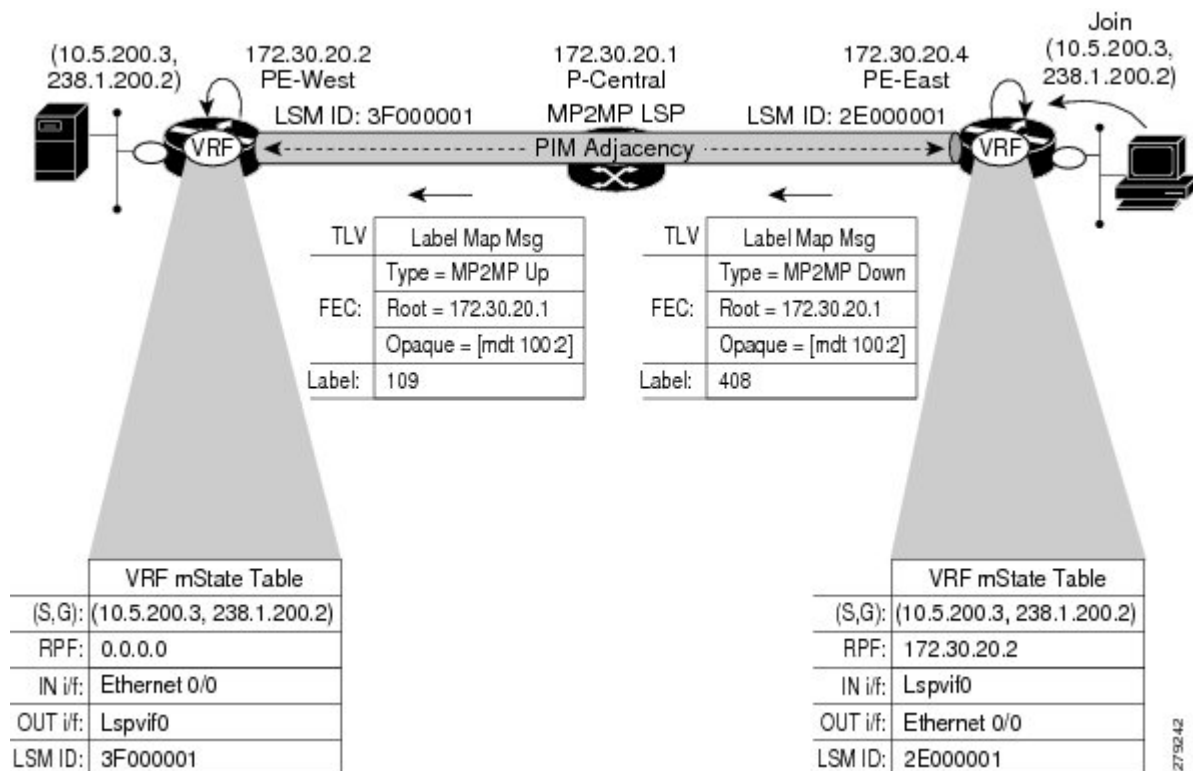


279241

## PIM Overlay Signaling of VPN Multicast State

The signaling of the multicast state within a VPN is via PIM. It is called overlay signaling because the PIM session runs over the multipoint LSP and maps the VPN multicast flow to the LSP. In an MVPN, the operation of PIM is independent of the underlying tunnel technology. In the MVPN solution, a PIM adjacency is created between PE devices, and the multicast states within a VRF are populated over the PIM sessions. When using MLDP, the PIM session runs over an LSP-VIF interface. The figure shows PIM signaling running over the default MDT MP2MP LSP. Access to the MP2MP LSP is via the LSP-VIF, which can see all the leaf PE devices at the end of branches, much like a LAN interface. In the figure, PE-East sends a downstream label mapping message to the root, P-Central, which in turn sends an upstream label mapping message to PE-West. These messages result in the creation of the LSP between the two leaf PE devices. A PIM session can then be activated over the top of the LSP allowing the (S, G) states and control messages to be signaled between PE-West and PE-East. In this case, PE-East receives a Join TLV message for (10.5.200.3, 238.1.200.2) within VRF, which it inserts into the mroute table. The Join TLV message is then sent via the PIM session to PE-West (BGP next-hop of 10.5.200.3), which populates its VRF mroute table. This procedure is identical to the procedure using an mGRE tunnel.

Figure 38: PIM Signaling over LSP



## Data MDT Scenario

In an MVPN, traffic that exceeds a certain threshold can move off the default MDT onto a data MDT.

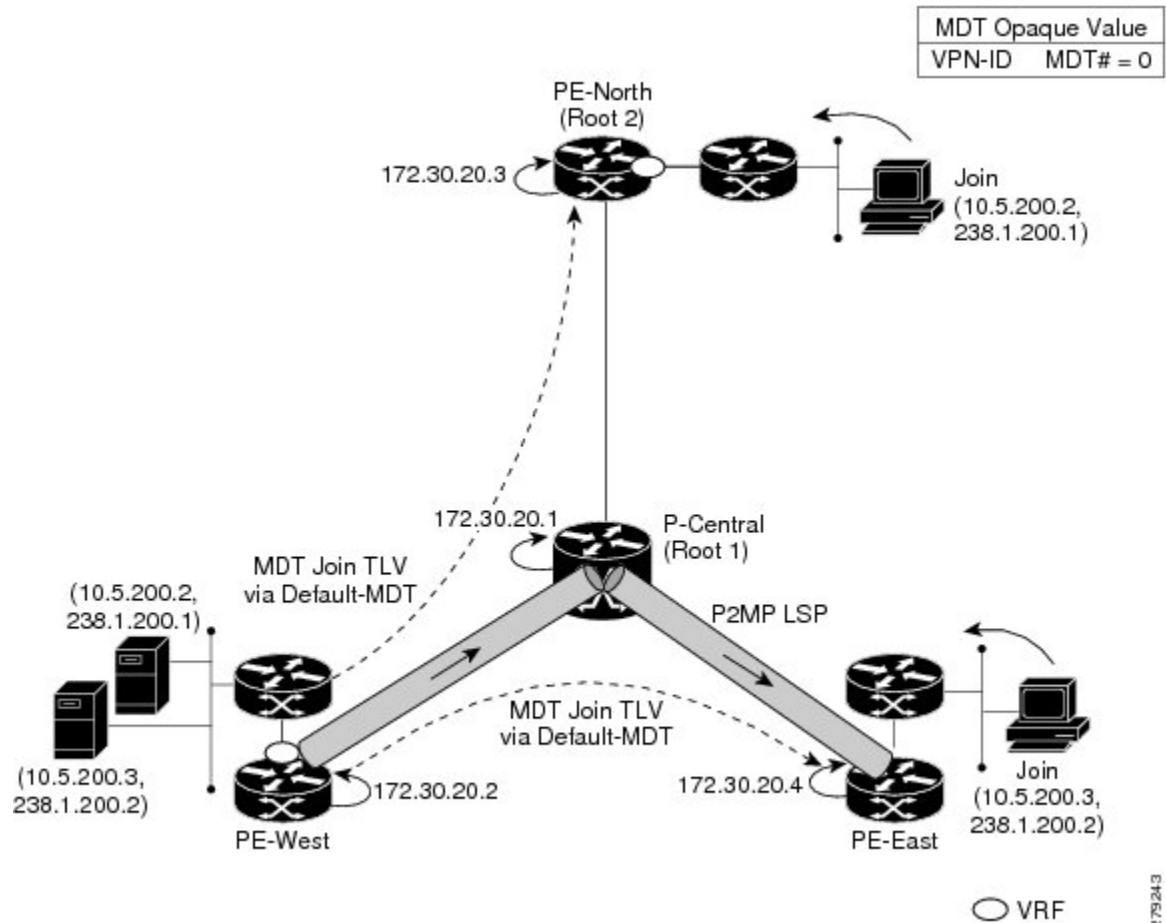
The figure shows the data MDT scenario. The Opaque value used to signal a data MDT consists of two parameters: the VPN ID and the MDT number in the format (vpn-id, MDT# > 0) where vpn-id is a manually configured 7-byte number that uniquely identifies this VPN. The second parameter is the unique data MDT number for this VPN, which is a number greater than zero.

In the scenario, two receivers at PE-North and PE-East are interested in two sources at PE-West. If the source 10.5.200.3 exceeds the threshold on the default MDT, PE-West will issue an MDT Join TLV message over the default MDT MP2MP LSP advising all PE devices that a new data MDT is being created.

Because PE-East has an interested receiver in VRF, it will build a multipoint LSP using P2MP back to PE-West, which will be the root of the tree. PE-North does not have a receiver for 10.5.200.3, therefore it will just cache the Join TLV message.



Figure 39: Data MDT Scenario



## P2MP and MP2MP Label Switched Paths

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

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

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

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

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



**Note** We recommend that you configure one P2MP MDT tree per prefix. For example, if 500 multicast routes are needed, then you should configure at least 500 P2MP MDT trees.

## Packet Flow in MLDP-based MVPN

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

## Realizing an MLDP-based MVPN

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

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

The device performs the following important functions for the implementation of MLDP:

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

## Overview of MVPN MLDP Partitioned MDT

MVPN allows a service provider to configure and support multicast traffic in an MPLS VPN environment. This type supports routing and forwarding of multicast packets for each individual VPN routing and forwarding (VRF) instance, and it also provides a mechanism to transport VPN multicast packets across the service provider backbone. In the MLDP case, the regular label switch path forwarding is used, so core does not need

to run PIM protocol. In this scenario, the c-packets are encapsulated in the MPLS labels and forwarding is based on the MPLS Label Switched Paths (LSPs).

The MVPN MLDP service allows you to build a Protocol Independent Multicast (PIM) domain that has sources and receivers located in different sites.

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

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

Separate multicast routing and forwarding tables are maintained for each VPN routing and forwarding (VRF) instance, with traffic being sent through VPN tunnels across the service provider backbone.

In the Rosen MVPN MLDP solution, a multipoint-to-multipoint (MP2MP) default MDT is setup to carry control plane and data traffic. A disadvantage with this solution is that all PE routers that are part of the MVPN need to join this default MDT tree. Setting up a MP2MP tree between all PE routers of a MVPN is equivalent to creating N P2MP trees rooted at each PE (Where N is the number of PE routers). In an Inter-AS (Option A) solution this problem is exacerbated since all PE routers across all AS'es need to join the default MDT. Another disadvantage of this solution is that any packet sent through a default MDT reaches all the PE routers even if there is no requirement.

In the partitioned MDT approach, only those egress PE routers that receive traffic requests from a particular ingress PE join the PMSI configured at that ingress PE. This makes the number of ingress PE routers in a network to be low resulting in a limited number of trees in the core.

## Supported MLDP Profiles

| Profile Name                                                            | Supported on MLDP |
|-------------------------------------------------------------------------|-------------------|
| Profile 1 Default MDT - MLDP MP2MP - PIM C-mcast Signaling              | Yes               |
| Profile 2 Partitioned MDT - MLDP MP2MP - PIM C-mcast Signaling          | No                |
| Profile 4 Partitioned MDT - MLDP MP2MP - BGP-AD - PIM C-mcast Signaling | No                |
| Profile 5 Partitioned MDT - MLDP P2MP - BGP-AD - PIM C-mcast Signaling  | No                |
| Profile 6 VRF MLDP - In-band Signaling                                  | No                |
| Profile 7 Global MLDP In-band Signaling                                 | No                |
| Profile 9 Default MDT - MLDP - MP2MP - BGP-AD - PIM C-mcast Signaling   | No                |

| Profile Name                                                            | Supported on MLDP |
|-------------------------------------------------------------------------|-------------------|
| Profile 12 Default MDT - MLDP - P2MP - BGP-AD - BGP C-mcast Signaling   | No                |
| Profile 13 Default MDT - MLDP - MP2MP - BGP-AD - BGP C-mcast Signaling  | Yes               |
| Profile 14 Partitioned MDT - MLDP P2MP - BGP-AD - BGP C-mast Signaling  | Yes               |
| Profile 15 Partitioned MDT - MLDP MP2MP - BGP-AD - BGP C-mast Signaling | No                |
| Profile 17 Default MDT - MLDP - P2MP - BGP-AD - PIM C-mcast Signaling   | No                |

# How to Configure MLDP-Based MVPN

## Configuring Initial MLDP Settings

Perform this task to configure the initial MLDP settings.

### Procedure

|               | Command or Action                                                                                                       | Purpose                                                                                                               |
|---------------|-------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>enable</b><br><b>Example:</b><br><pre>Device&gt; enable</pre>                                                        | Enables privileged EXEC mode.<br><ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul> |
| <b>Step 2</b> | <b>configure terminal</b><br><b>Example:</b><br><pre>Device# configure terminal</pre>                                   | Enters global configuration mode.                                                                                     |
| <b>Step 3</b> | <b>mpls mldp logging notifications</b><br><b>Example:</b><br><pre>Device(config)# mpls mldp logging notifications</pre> | Enables MLDP logging notifications.                                                                                   |
| <b>Step 4</b> | <b>end</b><br><b>Example:</b><br><pre>Device(config)# end</pre>                                                         | Ends the current configuration session and returns to privileged EXEC mode.                                           |

## Configuring an MLDP-Based MVPN

Perform this task to configure an MLDP-based MVPN.

### Procedure

|               | Command or Action                                                                                           | Purpose                                                                                                                                                                                               |
|---------------|-------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>enable</b><br><b>Example:</b><br>Device> enable                                                          | Enables privileged EXEC mode.<br>• Enter your password if prompted.                                                                                                                                   |
| <b>Step 2</b> | <b>configure terminal</b><br><b>Example:</b><br>Device# configure terminal                                  | Enters global configuration mode.                                                                                                                                                                     |
| <b>Step 3</b> | <b>ip multicast-routing</b><br><b>Example:</b><br>Device(config)# ip multicast-routing                      | Enables IP multicast routing.                                                                                                                                                                         |
| <b>Step 4</b> | <b>ip multicast-routing vrf vrf-name</b><br><b>Example:</b><br>Device(config)# ip multicast-routing vrf VRF | Enables IP multicast routing for the MVPN VRF specified for the <i>vrf-name</i> argument.                                                                                                             |
| <b>Step 5</b> | <b>vrf definition vrf-name</b><br><b>Example:</b><br>Device(config)# vrf definition VRF                     | Enters VRF configuration mode and defines the VPN routing instance by assigning a VRF name.                                                                                                           |
| <b>Step 6</b> | <b>rd route-distinguisher</b><br><b>Example:</b><br>Device(config-vrf)# rd 50:11                            | Creates a route distinguisher (RD) (in order to make the VRF functional). Creates the routing and forwarding tables, associates the RD with the VRF instance, and specifies the default RD for a VPN. |
| <b>Step 7</b> | <b>vpn id oui : vpn-index</b><br><b>Example:</b><br>Device(config-vrf)# vpn id 50:10                        | Sets or updates the VPN ID on a VRF instance.                                                                                                                                                         |
| <b>Step 8</b> | <b>address family ipv4</b><br><b>Example:</b><br>Device(config-vrf)# address family ipv4                    | Enters VRF address family configuration mode to specify an address family for a VRF.<br>• The <b>ipv4</b> keyword specifies an IPv4 address family for a VRF.                                         |

|                | Command or Action                                                                                                                         | Purpose                                                                     |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| <b>Step 9</b>  | <b>mdt preference { mldp   pim }</b><br><b>Example:</b><br><pre>Device(config-vrf-af)# mdt preference mldp</pre>                          | Specifies a preference for a particular MDT type (MLDP or PIM).             |
| <b>Step 10</b> | <b>mdt default mpls mldp group-address</b><br><b>Example:</b><br><pre>Device(config-vrf-af)# mdt default mpls mldp 172.30.20.1</pre>      | Configures a default MDT group for a VPN VRF instance.                      |
| <b>Step 11</b> | <b>mdt data mpls mldp number-of-data-mdt</b><br><b>Example:</b><br><pre>Device(config-vrf-af)# mdt data mpls mldp 255</pre>               | Specifies a range of addresses to be used in the data MDT pool.             |
| <b>Step 12</b> | <b>mdt data threshold kb/s list access-list</b><br><b>Example:</b><br><pre>Device(config-vrf-af)# mdt data threshold 40 list 1</pre>      | Defines the bandwidth threshold value in kilobits per second.               |
| <b>Step 13</b> | <b>route target export route-target-ext-community</b><br><b>Example:</b><br><pre>Device(config-vrf-af)# route target export 100:100</pre> | Creates an export route target extended community for the specified VRF.    |
| <b>Step 14</b> | <b>route target import route-target-ext-community</b><br><b>Example:</b><br><pre>Device(config-vrf-af)# route target import 100:100</pre> | Creates an import route target extended community for the specified VRF.    |
| <b>Step 15</b> | <b>end</b><br><b>Example:</b><br><pre>Device(config-vrf-af)# end</pre>                                                                    | Ends the current configuration session and returns to privileged EXEC mode. |

## Verifying the Configuration of an MLDP-Based MVPN

Perform this task in privileged EXEC mode to verify the configuration of an MLDP-based MVPN.

## Procedure

### Step 1 show mpls mldp database

Enter the **show mpls mldp database** command to display information in the MLDP database. It shows the FEC, the Opaque value of the FEC decoded, and the replication clients associated with it:

#### Example:

```
Device# show mpls mldp database
* For interface indicates MLDP recursive forwarding is enabled
* For RPF-ID indicates wildcard value
> Indicates it is a Primary MLDP MDT Branch

LSM ID : CB (RNR LSM ID: CC) Type: MP2MP Uptime : 00:01:38
FEC Root : 2.2.2.2 (we are the root)
Opaque decoded : [mdt 3001:1 0]
Opaque length : 11 bytes
Opaque value : 02 000B 003001000000001000000000
RNR active LSP : (this entry)
Upstream client(s) :
 None
 Expires : N/A Path Set ID : D5
Replication client(s):
> MDT (VRF vrf3001)
 Uptime : 00:01:38 Path Set ID : D6
 Interface : Lspvif101 RPF-ID : *
 33.33.33.33:0
 Uptime : 00:01:22 Path Set ID : D7
 Out label (D) : 2343 Interface : Vlan2222*
 Local label (U) : 466 Next Hop : 26.1.3.2
```

### Step 2 show ip pim neighbor [vrf vrf-name] neighbor [interface-type interface-number]

Enter the **show ip pim neighbor** command to display PIM adjacencies information:

#### Example:

```
Device# show ip pim vrf vrf3001 neighbor
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
 P - Proxy Capable, S - State Refresh Capable, G - GenID Capable,
 L - DR Load-balancing Capable
Neighbor Interface Uptime/Expires Ver DR
Address
192.168.1.2 Port-channel122.3001 3d19h/00:01:30 v2 1 / DR B S P G
5.5.5.5 Lspvif101 00:01:48/00:01:25 v2 1 / B S P G
7.7.7.7 Lspvif101 00:01:48/00:01:25 v2 1 / DR S P G
```

### Step 3 show ip mroute [vrf vrf-name] [[active [kpbs] [interface type number] | bidirectional | count [terse] | dense | interface type number | proxy | pruned | sparse | ssm | static | summary] | [group-address [source-address]] [count [terse] | interface type number | proxy | pruned | summary] | [source-address group-address] [count [terse] | interface type number | proxy | pruned | summary] | [group-address] active [kpbs] [interface type number | verbose]]

Enter the **show ip mroute** command to display the contents of the multicast routing (mroute) table:

#### Example:

```

Device# show ip mroute vrf vrf3001 225.1.1.1 30.22.1.10
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
 L - Local, P - Pruned, R - RP-bit set, F - Register flag,
 T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
 X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
 U - URD, I - Received Source Specific Host Report,
 Z - Multicast Tunnel, z - MDT-data group sender,
 Y - Joined MDT-data group, y - Sending to MDT-data group,
 G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
 N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
 Q - Received BGP S-A Route, q - Sent BGP S-A Route,
 V - RD & Vector, v - Vector, p - PIM Joins on route,
 x - VxLAN group, c - PFP-SA cache created entry,
 * - determined by Assert, # - iif-starg configured on rpf intf,
 e - encap-helper tunnel flag
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(30.22.1.10, 225.1.1.1), 00:31:08/00:02:14, flags: JTY
Incoming interface: Lspvif101, RPF nbr 2.2.2.2, MDT: [2, 2.2.2.2]/00:02:51
Outgoing interface list:
Vlan3001, Forward/Sparse, 00:31:08/00:02:35

```

**Step 4** **show mpls forwarding-table** [*network {mask | length}*] | **labels** *label* [- *label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*] [**vrf** *vrf-name*] [**detail**]

Enter the **show mpls forwarding-table** command to display the contents of the MPLS Label Forwarding Information Base (LFIB):

**Example:**

```

Device# show mpls forwarding-table vrf vrf3001
Local Outgoing Prefix Bytes Label Outgoing Next Hop
Label Label or Tunnel Id Switched interface
150 No Label 192.168.1.0/24[V] \
 0 aggregate/vrf3001
356 No Label 30.1.30.2/32[V] 0 Po122.3001 192.168.1.2
357 No Label 30.1.30.1/32[V] 0 Po122.3001 192.168.1.2
358 No Label 30.22.1.0/24[V] 0 Po122.3001 192.168.1.2
466 [T] No Label [mdt 3001:1 0][V] \
 65660 aggregate/vrf3001

[T] Forwarding through a LSP tunnel.
 View additional labelling info with the 'detail' option

```

## Configuration Examples for MLDP-Based MVPN

### Example: Initial Deployment of an MLDP-Based MVPN

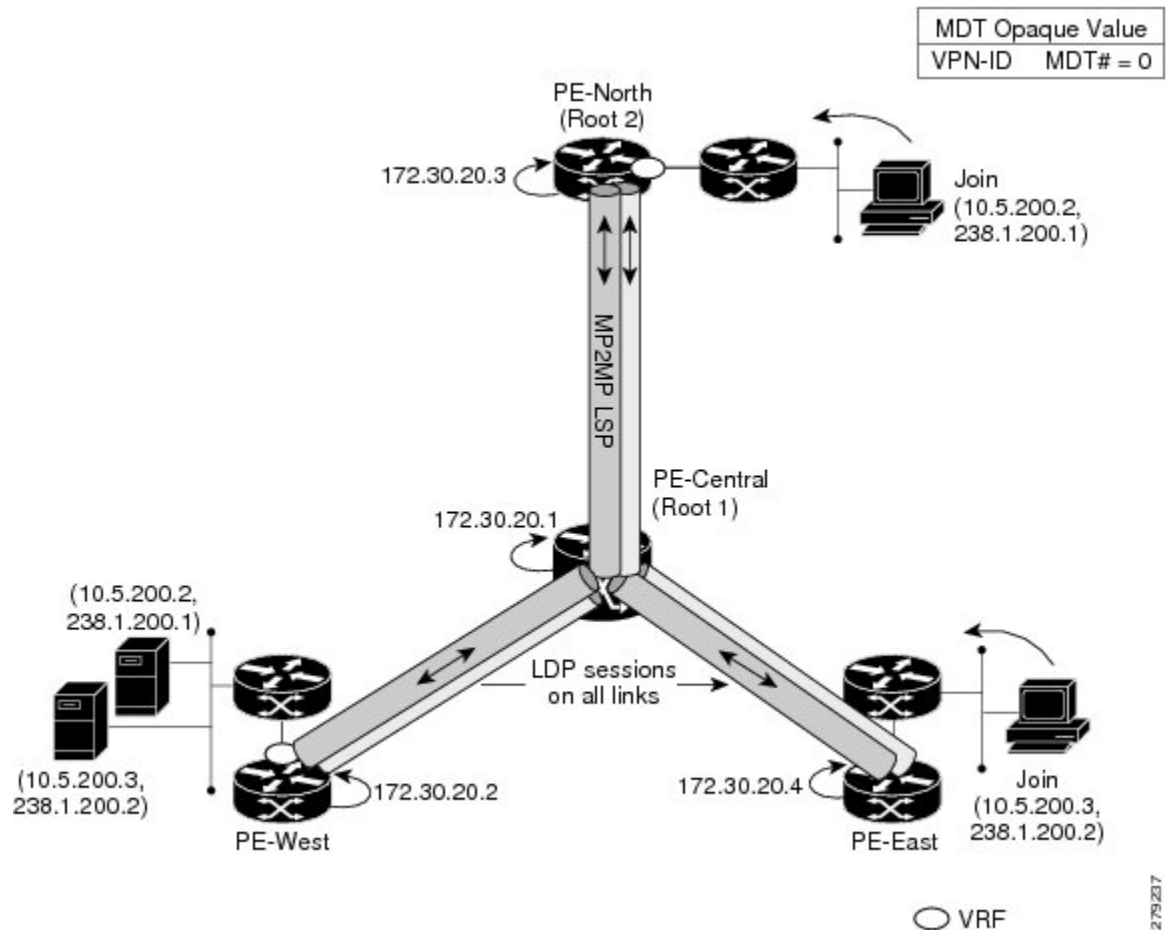
Initial deployment of an MLDP-based MVPN involves the configuration of a default MDT and one or more data MDTs.



## Default MDT Configuration

The following example shows how to configure the default MDT for an MLDP-based MVPN. This configuration is based on the sample topology illustrated in the figure.

**Figure 40: Default MDT Example**



This configuration is consistent for every PE device participating in the same VPN ID. The **vpn id 100:2** command replaces the MDT group address used with the mGRE transport method. To provide redundancy, two default MDT trees are statically configured, rooted at P-Central and PE-North. The selection as to which MP2MP tree the default MDT will use at a particular PE device is determined by Interior Gateway Protocol (IGP) metrics. An MP2MP LSP is implicit for the default MDT.

```
ip pim mpls source Loopback0
ip multicast-routing
ip multicast-routing vrf VRF
!
ip vrf VRF
rd 100:2
vpn id 100:2
route-target export 200:2
route-target import 200:2
mdt default mpls mldp 172.30.20.1 (P-Central)
mdt default mpls mldp 172.30.20.3 (PE-North)
```

## PIM Adjacencies

PIM operates over the LSP-VIF as if it were a regular tunnel interface. That means PIM hellos are exchanged over the LSP-VIF to establish PIM adjacencies over the default MDT. The sample output in this section displays the three PIM adjacencies in VRF of PE-East. The listed entries here are the adjacencies to PE-West and PE-North over the MP2MP LSP via LSP-VIF interface 101.

```
PE-East# show ip pim vrf vrf3001 neighbor
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
 P - Proxy Capable, S - State Refresh Capable, G - GenID Capable,
 L - DR Load-balancing Capable
Neighbor Interface Uptime/Expires Ver DR
Address
5.5.5.5 Lspvif0 00:18:54/00:01:33 v2 1 / S P G
2.2.2.2 Lspvif0 1d00h/00:01:34 v2 1 / S P G
22.22.22.22 Lspvif0 1d00h/00:01:34 v2 1 / DR S P G
```

The output from the **show ip mroute** command also shows the (S, G) entry for VRF. The stream 225.1.1.1 has the Reverse Path Forwarding (RPF) interface of LSP-VIF interface 101 and the neighbor 2.2.2.2, which is PE-West.

```
PE-East# show ip mroute vrf vrf3001 225.1.1.1 30.22.1.10
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
 L - Local, P - Pruned, R - RP-bit set, F - Register flag,
 T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
 X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
 U - URD, I - Received Source Specific Host Report,
 Z - Multicast Tunnel, z - MDT-data group sender,
 Y - Joined MDT-data group, y - Sending to MDT-data group,
 G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
 N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
 Q - Received BGP S-A Route, q - Sent BGP S-A Route,
 V - RD & Vector, v - Vector, p - PIM Joins on route,
 x - VxLAN group, c - PFP-SA cache created entry,
 * - determined by Assert, # - iif-starg configured on rpf intf,
 e - encap-helper tunnel flag
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(30.22.1.10, 225.1.1.1), 00:31:08/00:02:14, flags: JTY
Incoming interface: Lspvif101, RPF nbr 2.2.2.2, MDT: [2, 2.2.2.2]/00:02:51
Outgoing interface list:
 Vlan3001, Forward/Sparse, 00:31:08/00:02:35
```

## MLDP Database Entry--PE-East

The sample output in this section displays the database entries for the MP2MP trees supporting the default MDT at PE-East. The database is searched by Opaque value MDT 3001:1, which results in information for two MP2MP trees (one for each root) being returned. Both trees have different system IDs and use the same Opaque value ([mdt 3001:1 1]), but with different roots. Entry 3E0 shows it is the primary MP2MP tree, therefore PE-East will transmit all source multicast traffic on this LSP, and 21C will be the backup root. The interface LSP-VIF interface 101 represents both MP2MP LSPs. The Local Label (D) is the downstream label allocated by PE-East for this tree. In other words, traffic from the root will be received with either the label of the primary tree or the backup tree. The Out Label (U) is the label that PE-East will use to send traffic into

the tree; upstream towards the root, either 361 for the Primary Tree or 363 for the Backup Tree. Both these labels were received from P-Central.

```

PE-East# show mpls mldp database opaque_type mdt 3001:1
LSM ID : 3E0 Type: P2MP Uptime : 00:34:24
 FEC Root : 2.2.2.2
 Opaque decoded : [mdt 3001:1 1]
 Opaque length : 11 bytes
 Opaque value : 02 000B 003001000000001000000001
 Upstream client(s) :
 33.33.33.33:0 [Active]
 Expires : Never Path Set ID : 1C0
 Out Label (U) : None Interface : Port-channel23*
 Local Label (D): 361 Next Hop : 104.2.3.2
 Replication client(s):
 MDT (VRF vrf3001)
 Uptime : 00:34:24 Path Set ID : None
 Interface : Lspvif101 RPF-ID : *

LSM ID : 21C Type: P2MP Uptime : 00:34:16
 FEC Root : 2.2.2.2
 Opaque decoded : [mdt 3001:1 2]
 Opaque length : 11 bytes
 Opaque value : 02 000B 003001000000001000000002
 Upstream client(s) :
 33.33.33.33:0 [Active]
 Expires : Never Path Set ID : 17D
 Out Label (U) : None Interface : Port-channel23*
 Local Label (D): 363 Next Hop : 104.2.3.2
 Replication client(s):
 MDT (VRF vrf3001)
 Uptime : 00:34:16 Path Set ID : None
 Interface : Lspvif101 RPF-ID : *

```

### Label Forwarding Entry--P-Central (Root 1)

The sample output shown in this section displays the VRF (MDT 3001:1) MLDP database entry 7035A for the primary MP2MP LSP, which is P-Central. Because the local device P-Central is the root, there is no upstream peer ID, therefore no labels are allocated locally. However there are three replication clients, representing each of the three PE devices: PE-North, PE-West, and PE-East. These replication clients are the downstream nodes of the P2MP LSP. These clients receive multipoint replicated traffic.

In the replication entry looking from the perspective of the root, there are two types of labels:

- Out label (D)--These are labels received from remote peers that are downstream to the root (remember traffic flows downstream away from the root).
- Local label (U)--These are labels provided by P-Central to its neighbors to be used as upstream labels (sending traffic to the root). It is easy to identify these labels as they all start in the 100 range, which we have configured for P-Central to use. P-Central sends these labels out when it receives a FEC with the type as MP2MP Down.

From the labels received and sent in the replication entries, the Label Forwarding Information Base (LFIB) is created. The LFIB has one entry per upstream path and one entry per downstream path. In this case because P-Central is the root, there are only upstream entries in the LFIB that have been merged with the corresponding downstream labels. For example, label 105 is the label P-Central sent to PE-East to send source traffic upstream. Traffic received from PE-East will then be replicated using the downstream labels 307 to PE-West and 208 to PE-North.

```

P-Central# show mpls mldp database opaque_type mdt 3001:1
LSM ID : 7035A Type: P2MP Uptime : 00:01:13
FEC Root : 2.2.2.2
Opaque decoded : [mdt 3001:1 1]
Opaque length : 11 bytes
Opaque value : 02 000B 003001000000001000000001
Upstream client(s) :
 33.33.33.33:0 [Active]
 Expires : Never Path Set ID : 501A2
 Out Label (U) : None Interface : Vlan31*
 Local Label (D) : 997 Next Hop : 104.3.1.2
Replication client(s):
 MDT (VRF vrf3001)
 Uptime : 00:01:13 Path Set ID : None
 Interface : Lspvif1 RPF-ID : *

```

The sample output shown in this section displays the entry on P-Central for the P2MP LSP rooted at PE-North (backup root). In this tree P-Central is a branch of the tree, not a root, therefore there are some minor differences to note:

- The upstream peer ID is PE-North, therefore P-Central has allocated label 915 in the downstream direction towards PE-North and subsequently PE-North has responded with an upstream label.
- Two replication entries representing PE-East and PE-West are displayed.
- The merged LFIB shows three entries:
  - One downstream entry label 915 receiving traffic from Root 2 (PE-North), which is then directed further downstream using out labels of PE-West and PE-East.
  - Two upstream entries receiving traffic from the leaves and directing it either downstream or upstream using label out label.

```

Central_P# show mpls mldp database opaque_type mdt 3001:1
LSM ID : 3024C (RNR LSM ID: 1026F) Type: MP2MP Uptime : 2w3d
FEC Root : 2.2.2.2
Opaque decoded : [mdt 3001:1 0]
Opaque length : 11 bytes
Opaque value : 02 000B 003001000000001000000000
RNR active LSP : 101F6 (root: 22.22.22.22)
Upstream client(s) :
 33.33.33.33:0 [Active]
 Expires : Never Path Set ID : D0157
 Out Label (U) : 4069 Interface : Port-channel31*
 Local Label (D) : 915 Next Hop : 104.3.1.2
Replication client(s) :
 > MDT (VRF vrf3001)
 Uptime : 2w3d Path Set ID : F0036
 Interface : Lspvif1 RPF-ID : *
 7.7.7.7:0
 Uptime : 1d20h Path Set ID : B01ED
 Out label (D) : 25 Interface : Port-channel71.1*
 Local label (U) : 941 Next Hop : 104.71.1.1

LSM ID : 101F6 (RNR LSM ID: 1026F) Type: MP2MP Uptime : 21:17:45
FEC Root : 22.22.22.22 (we are the root)
Opaque decoded : [mdt 3001:1 0]
Opaque length : 11 bytes
Opaque value : 02 000B 003001000000001000000000

```

```

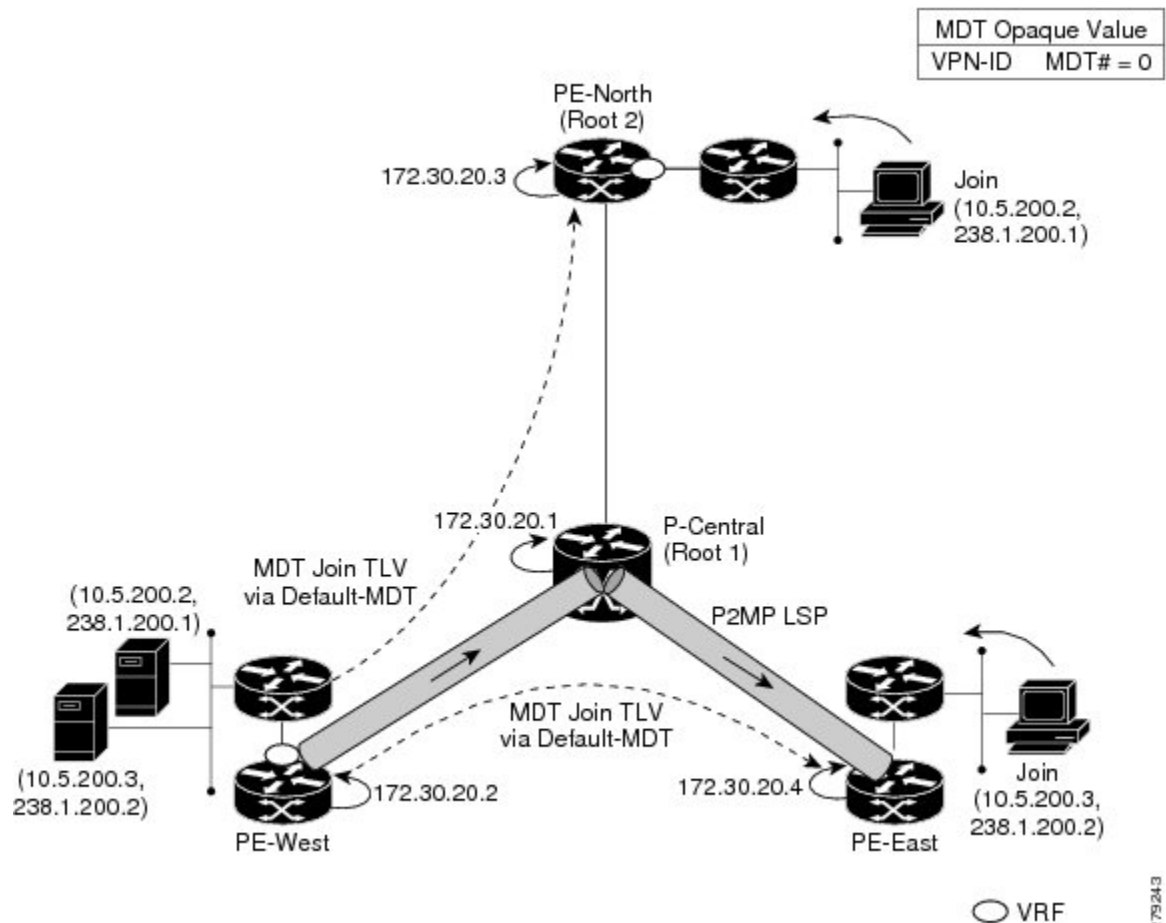
RNR active LSP : (this entry)
Candidate RNR ID(s) : 3024C
Upstream client(s) :
None
Expires : N/A Path Set ID : F007B
Replication client(s) :
> MDT (VRF vrf3001)
Uptime : 20:51:46 Path Set ID : C001F
Interface : Lspvif1 RPF-ID : *
7.7.7.7:0
Uptime : 20:51:43 Path Set ID : C0020
Out label (D) : 44 Interface : Port-channel71.1*
Local label (U) : 1191 Next Hop : 104.71.1.1
33.33.33.33:0
Uptime : 00:00:34 Path Set ID : 100049
Out label (D) : 3109 Interface : Port-channel31*
Local label (U) : 1340 Next Hop : 104.3.1.2

```

## Data MDT Configuration

The following example shows how to configure the data MDT for an MLDP-based MVPN. This configuration is based on the sample topology illustrated in the figure.

**Figure 41: Data MDT Example**



The sample output in this section displays the data MDT configuration for all the PE devices. The **mdt data** commands are the only additional commands necessary. The first **mdt data** command allows a maximum of 60 data MDTs to be created, and the second **mdt data** command sets the threshold. If the number of data MDTs exceeds 60, then the data MDTs will be reused in the same way as they are for the mGRE tunnel method (the one with the lowest reference count).

```
ip pim vrf VRF mpls source Loopback0
!
ip vrf VRF
 rd 100:2
 vpn id 100:2
 route-target export 200:2
 route-target import 200:2
 mdt default mpls mldp 172.30.20.1 (P-Central)
 mdt default mpls mldp 172.30.20.3 (PE-North)
 mdt data mpls mldp 60
 mdt data threshold 1
```

### VRF mroute Table--PE-West

The sample output in this section displays the VRF mroute table on PE-West before the high-bandwidth source exceeds the threshold. At this point there are two streams, representing each of the two VPN sources at PE-West, on a single MP2MP LSP (System ID 2). The LSP represents the default MDT accessed via LSP-VIF interface 0.

```
PE-West# show ip mroute vrf vrf3001 verbose.
.
.

(30.0.5.10, 228.1.1.1), 16:08:00/00:02:21, flags: FTAp
Incoming interface: Vlan3001, RPF nbr 0.0.0.0
Outgoing interface list:
Lspvif0, LSM MDT: 2 (default), Forward/Sparse, 16:08:00/00:03:25, Pkts:0, p
.
.
.

(30.0.5.10, 228.1.1.3), 15:55:20/00:01:38, flags: FTAp
Incoming interface: Vlan3001, RPF nbr 0.0.0.0
Outgoing interface list:
Lspvif0, LSM MDT: 2 (default), Forward/Sparse, 15:55:13/00:02:44, Pkts:0, p
```

The sample output in this section displays the output after the source transmission exceeds the threshold. PE-West sends an MDT Join TLV message to signal the creation of a data MDT. In this case, the data MDT number is 8, therefore PE-East will send a label mapping message back to PE-West with a FEC TLV containing root=PE-West, Opaque value=(mdt vpn-id 8). The System ID is now changed to D signaling a different LSP; however, the LSP-VIF is still LSP-VIF interface 0. The (S, G) entry also has the “y” flag set indicating this stream has switched to a data MDT.

```
PE-West# show ip mroute vrf vrf3001 228.1.1.3 30.0.5.10 verbose
.
.
.

(30.0.5.10, 228.1.1.3), 16:00:17/00:02:49, flags: FTApp
Incoming interface: Vlan3001, RPF nbr 0.0.0.0
MDT TX nr: 8 LSM-ID: 0xD
Outgoing interface list:
```

```
Lspvif0, LSM MDT: D (data), Forward/Sparse, 16:00:10/00:02:43, Pkts:0, p
```

## MLDP Database Entries

The sample output in this section displays the MLDLP entry for the data MDT (F) on the ingress device PE-West. The following points about this entry should be noted:

- The tree type is P2MP with PE-West (5.5.5.5) as the root.
- The Opaque value is [mdt 3001:1 10] denoting the first data MDT.
- There are no labels allocated as it is the root.
- There is one replication client entry on this tree.
- The MDT entry is an internal construct.

```
PE-West# show mpls mldp database id F
LSM ID : F Type: P2MP Uptime : 00:02:37
 FEC Root : 5.5.5.5 (we are the root)
 Opaque decoded : [mdt 3001:1 10]
 Opaque length : 11 bytes
 Opaque value : 02 000B 0030010000000010000000A
 Upstream client(s) :
 None
 Expires : N/A Path Set ID : 10
 Replication client(s):
> MDT (VRF vrf3001)
 Uptime : 00:02:37 Path Set ID : None
 Interface : Lspvif0 RPF-ID : *
 33.33.33.33:0
 Uptime : 00:02:37 Path Set ID : None
 Out label (D) : 3326 Interface : Port-channel23*
 Local label (U): None Next Hop : 104.2.3.2
```

The sample output in this section displays the database entry for the data MDT on PE-East, the egress device. Also shown is the MDT Join TLV message that was sent from PE-West over the default MDT. The MDT Join TLV message contains all the necessary information to allow PE-East to create a label mapping message P2MP LSP back to the root of PE-West.

```
PE-East# show mpls mldp database opaque_type mdt 3001:1
LSM ID : CD Type: P2MP Uptime : 00:33:46
 FEC Root : 2.2.2.2 (we are the root)
 Opaque decoded : [mdt 3001:1 1]
 Opaque length : 11 bytes
 Opaque value : 02 000B 003001000000001000000001
 Upstream client(s) :
 None
 Expires : N/A Path Set ID : D8
 Replication client(s):
> MDT (VRF vrf3001)
 Uptime : 00:33:46 Path Set ID : None
 Interface : Lspvif101 RPF-ID : *
 33.33.33.33:0
 Uptime : 00:33:46 Path Set ID : None
 Out label (D) : 348 Interface : Vlan2222*
 Local label (U): None Next Hop : 26.1.3.2

LSM ID : CE Type: P2MP Uptime : 00:33:38
```

```

FEC Root : 2.2.2.2 (we are the root)
Opaque decoded : [mdt 3001:1 2]
Opaque length : 11 bytes
Opaque value : 02 000B 003001000000001000000002
Upstream client(s) :
 None
 Expires : N/A Path Set ID : D9
Replication client(s):
> MDT (VRF vrf3001)
 Uptime : 00:33:38 Path Set ID : None
 Interface : Lspvif101 RPF-ID : *
 33.33.33.33:0
 Uptime : 00:33:38 Path Set ID : None
 Out label (D) : 2399 Interface : Vlan2222*
 Local label (U) : None Next Hop : 26.1.3.2

```

## LFIB Entry for the Data MDT

The sample output in this section displays the LFIB entry for the data MDT as it passes through P-Central and PE-East. The Tunnel ID used for the LSP is the Opaque value [mdt 3001:1 0].

```

P-Central# show mpls forwarding-table labels 1191
Local Outgoing Prefix Bytes Label Outgoing Next Hop
Label Label or Tunnel Id Switched interface
1191 2602 [mdt 3001:1 0][V] \
 156663076 Po31 104.3.1.2
 [T] No Label [mdt 3001:1 0][V] \
 45279264 aggregate/vrf3001

[T] Forwarding through a LSP tunnel.
 View additional labelling info with the 'detail' option

PE-East# show mpls forwarding-table vrf vrf3001
Local Outgoing Prefix Bytes Label Outgoing Next Hop
Label Label or Tunnel Id Switched interface
132 No Label 30.0.1.0/24[V] 0 drop
133 Pop Label 30.30.1.1/32[V] 0 aggregate/vrf3001
137 Pop Label 30.1.30.1/32[V] 0 aggregate/vrf3001
138 No Label 30.0.5.0/24[V] 0 aggregate/vrf3001
142 [T] No Label [mdt 3001:1 0][V] \
 905056 aggregate/vrf3001
145 [T] No Label [mdt 3001:1 0][V] \
 7448 aggregate/vrf3001

[T] Forwarding through a LSP tunnel.
 View additional labelling info with the 'detail' option

```

## Example: Configuring MVPN Profile 1 - Default MDT - MLDP MP2MP - PIM C-mcast Signaling

The following example shows how to configure MVPN Profile 1:

```

vrf definition one
rd 1:2
vpn id 1000:2000
!
address-family ipv4
mdt default mpls mldp 10.100.1.1
route-target export 1:1

```



```

 route-target import 1:1
 exit-address-family
!

ip multicast-routing vrf one

mpls mldp logging notifications

router bgp 1
 bgp log-neighbor-changes
 neighbor 10.100.1.7 remote-as 1
 neighbor 10.100.1.7 update-source Loopback0
 !
 address-family vpnv4
 neighbor 10.100.1.7 activate
 neighbor 10.100.1.7 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf one
 redistribute connected
 neighbor 10.2.2.9 remote-as 65002
 neighbor 10.2.2.9 activate
 exit-address-family

```

## Example: Configuring MVPN Profile 13 - Default MDT - MLDP - MP2MP - BGP-AD - BGP C-mcast Signaling

The following example shows how to configure MVPN Profile 13:

```

vrf definition one
 rd 1:1
 vpn id 1000:2000
 !
 address-family ipv4
 mdt auto-discovery mldp
 mdt default mpls mldp 10.100.1.3
 mdt overlay use-bgp
 route-target export 1:1
 route-target import 1:1
 exit-address-family
!

interface Ethernet2/0
 vrf forwarding one
 ip address 10.2.1.1 255.255.255.0
 ip pim sparse-mode

router bgp 1
 neighbor 10.100.1.7 remote-as 1
 neighbor 10.100.1.7 update-source Loopback0
 !
 address-family ipv4 mvpn
 neighbor 10.100.1.7 activate
 neighbor 10.100.1.7 send-community extended
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.100.1.7 activate
 neighbor 10.100.1.7 send-community extended
 exit-address-family

```

!

## Example: Configuring MVPN Profile 14 - Partitioned MDT - MLDP P2MP - BGP-AD - BGP C-mast Signaling

The following example shows how to configure MVPN Profile 14:

```
vrf definition one
 rd 1:1
 !
 address-family ipv4
 mdt auto-discovery mldp
 mdt strict-rpf interface
 mdt partitioned mldp p2mp
 mdt overlay use-bgp
 route-target export 1:1
 route-target import 1:1
 exit-address-family

!
interface Ethernet2/0
 vrf forwarding one
 ip address 10.2.1.1 255.255.255.0
 ip pim sparse-mode
!

router bgp 1
 neighbor 10.100.1.7 remote-as 1
 neighbor 10.100.1.7 update-source Loopback0
 !
 address-family ipv4 mvpn
 neighbor 10.100.1.7 activate
 neighbor 10.100.1.7 send-community extended
 exit-address-family
 !
 address-family vpv4
 neighbor 10.100.1.7 activate
 neighbor 10.100.1.7 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf one
 redistribute connected
 neighbor 10.2.1.8 remote-as 65001
 neighbor 10.2.1.8 activate
 exit-address-family
!
```

## Feature History for MLDP-Based MVPN

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

| Release                          | Feature            | Feature Information                                                                                                                                                                                                                                                           |
|----------------------------------|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cisco IOS XE<br>Amsterdam 17.3.3 | MLDP-Based<br>MVPN | The MLDP-based MVPN feature provides extensions to Label Distribution Protocol (LDP) for the setup of point-to-multipoint (P2MP) and multipoint-to-multipoint (MP2MP) label switched paths (LSPs) for transport in the Multicast Virtual Private Network (MVPN) core network. |

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>.





## CHAPTER 18

# IP Multicast Optimization: Optimizing PIM Sparse Mode in a Large IP Multicast Deployment

- [Prerequisites for Optimizing PIM Sparse Mode in a Large IP Multicast Deployment, on page 367](#)
- [Information About Optimizing PIM Sparse Mode in a Large IP Multicast Deployment, on page 367](#)
- [How to Optimize PIM Sparse Mode in a Large IP Multicast Deployment, on page 370](#)
- [Configuration Examples for Optimizing PIM Sparse Mode in a Large Multicast Deployment, on page 372](#)
- [Additional References for IP Multicast Optimization: Optimizing PIM Sparse Mode in a Large IP Multicast Deployment, on page 373](#)
- [Feature History for IP Multicast Optimization: Optimizing PIM Sparse Mode in a Large IP Multicast Deployment, on page 373](#)

## 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

IP multicast sources do not use a signaling mechanism to announce their presence. Sources just send their data into the attached network, as opposed to receivers that use Internet Group Management Protocol (IGMP) to announce their presence. If a source sends traffic to a multicast group configured in PIM sparse mode (PIM-SM), the Designated Router (DR) leading toward the source must inform the rendezvous point (RP) about the presence of this source. If the RP has downstream receivers that want to receive the multicast traffic

(natively) from this source and has not joined the shortest path leading toward the source, then the DR must send the traffic from the source to the RP. The PIM registering process, which is individually run for each (S, G) entry, accomplishes these tasks between the DR and RP.

The registering process begins when a DR creates a new (S, G) state. The DR encapsulates all the data packets that match the (S, G) state into PIM register messages and unicasts those register messages to the RP.

If an RP has downstream receivers that want to receive register messages from a new source, the RP can either continue to receive the register messages through the DR or join the shortest path leading toward the source. By default, the RP will join the shortest path, because delivery of native multicast traffic provides the highest throughput. Upon receipt of the first packet that arrives natively through the shortest path, the RP will send a register-stop message back to the DR. When the DR receives this register-stop message, it will stop sending register messages to the RP.

If an RP has no downstream receivers that want to receive register messages from a new source, the RP will not join the shortest path. Instead, the RP will immediately send a register-stop message back to the DR. When the DR receives this register-stop message, it will stop sending register messages to the RP.

Once a routing entry is established for a source, a periodic reregistering takes place between the DR and RP. One minute before the multicast routing table state times out, the DR will send one dataless register message to the RP each second that the source is active until the DR receives a register-stop message from the RP. This action restarts the timeout time of the multicast routing table entry, typically resulting in one reregistering exchange every 2 minutes. Reregistering is necessary to maintain state, to recover from lost state, and to keep track of sources on the RP. It will take place independently of the RP joining the shortest path.

## PIM Version 1 Compatibility

If an RP is running PIM Version 1, it will not understand dataless register messages. In this case, the DR will not send dataless register messages to the RP. Instead, approximately every 3 minutes after receipt of a register-stop message from the RP, the DR encapsulates the incoming data packets from the source into register messages and sends them to the RP. The DR continues to send register messages until it receives another register-stop message from the RP. The same behavior occurs if the DR is running PIM Version 1.

When a DR running PIM Version 1 encapsulates data packets into register messages for a specific (S, G) entry, the entry is process-switched, not fast-switched or hardware-switched. On platforms that support these faster paths, the PIM registering process for an RP or DR running PIM Version 1 may lead to periodic out-of-order packet delivery. For this reason, we recommend upgrading your network from PIM Version 1 to PIM Version 2.

## PIM Designated Router

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.

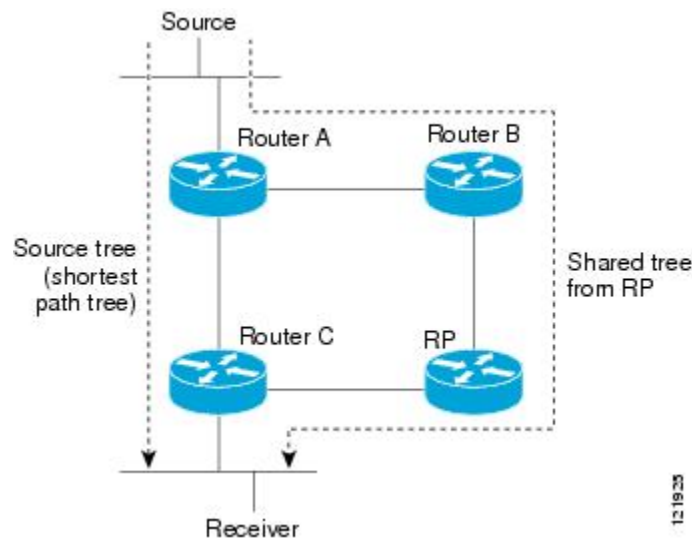
## Preventing Use of Shortest-Path Tree to Reduce Memory Requirement

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

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 42: 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 [PIM Shared Tree and Source Tree - Shortest-Path Tree, on page 369](#)). 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

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.



## Procedure

|               | Command or Action                                                                                                                                                              | Purpose                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>enable</b><br><b>Example:</b><br><pre>Router&gt; enable</pre>                                                                                                               | Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Step 2</b> | <b>configure terminal</b><br><b>Example:</b><br><pre>Router# configure terminal</pre>                                                                                          | Enters global configuration mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>Step 3</b> | <b>ip pim register-rate-limit <i>rate</i></b><br><b>Example:</b><br><pre>Router(config)# ip pim register-rate-limit 10</pre>                                                   | (Optional) Sets a limit on the maximum number of PIM sparse mode register messages sent per second for each (S, G) routing entry. <ul style="list-style-type: none"> <li>• Use this command to limit the number of register messages that the designated router (DR) will allow for each (S, G) entry.</li> <li>• By default, there is no maximum rate set.</li> <li>• 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.</li> <li>• Receivers may experience data packet loss within the first second in which register messages are sent from bursty sources.</li> </ul>                                                        |
| <b>Step 4</b> | <b>ip pim spt-threshold {<i>kbits</i> infinity}[<i>group-list access-list</i>]</b><br><b>Example:</b><br><pre>Router(config)# ip pim spt-threshold infinity group-list 5</pre> | (Optional) Specifies the threshold that must be reached before moving to the shortest-path tree. <ul style="list-style-type: none"> <li>• The default value is <b>0</b>, which causes the router to join the SPT immediately upon the first data packet it receives.</li> <li>• Specifying the <b>infinity</b> 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.</li> <li>• 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.</li> </ul> |

|               | Command or Action                                                                                                             | Purpose                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------------|-------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               |                                                                                                                               | <ul style="list-style-type: none"> <li>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</li> </ul>                                                                                                                                                                                                                                                                                                           |
| <b>Step 5</b> | <b>interface</b> <i>type number</i><br><b>Example:</b><br><pre>Router(config)# interface GigabitEthernet 1/0/1</pre>          | Configures an interface. <ul style="list-style-type: none"> <li>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.</li> </ul>                                                                                                                                                                                                                                                                                                                            |
| <b>Step 6</b> | <b>ip pim query-interval</b> <i>period</i> [msec]<br><b>Example:</b><br><pre>Router(config-if)# ip pim query-interval 1</pre> | (Optional) Configures the frequency at which multicast routers send PIM router query messages. <ul style="list-style-type: none"> <li>Perform this step only on redundant routers on the edge of a PIM domain.</li> <li>The default query interval is 30 seconds.</li> <li>The <i>period</i> argument is in seconds unless the <b>msec</b> keyword is specified.</li> <li>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.</li> </ul> |

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

### 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 GigabitEthernet 1/0/1
 ip pim query-interval 1
.
```

```
.
!
ip pim spt-threshold infinity
ip pim register-rate-limit 10
!
```

## Additional References for IP Multicast Optimization: Optimizing PIM Sparse Mode in a Large IP Multicast Deployment

### Related Documents

| Related Topic                                                                    | Document Title                                                                                                |
|----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| For complete syntax and usage information for the commands used in this chapter. | See the IP Multicast Routing Commands section of the <i>Command Reference (Catalyst 9400 Series Switches)</i> |

## Feature History for IP Multicast Optimization: Optimizing PIM Sparse Mode in a Large IP Multicast Deployment

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

| Release                     | Feature                                                                                  | Feature Information                                                                                                                                                      |
|-----------------------------|------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cisco IOS XE Everest 16.6.1 | IP Multicast Optimization: Optimizing PIM Sparse Mode in a Large IP Multicast Deployment | Protocol Independent Multicast (PIM) has two basic operating modes: sparse-mode and dense-mode, and is suitable for large networks with heterogeneous links and devices. |

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <https://cfngng.cisco.com/>





## CHAPTER 19

# IP Multicast Optimization: Multicast Subsecond Convergence

---

- [Prerequisites for Multicast Subsecond Convergence, on page 375](#)
- [Restrictions for Multicast Subsecond Convergence, on page 375](#)
- [Information About Multicast Subsecond Convergence, on page 375](#)
- [How to Configure Multicast Subsecond Convergence, on page 377](#)
- [Configuration Examples for Multicast Subsecond Convergence, on page 378](#)
- [Additional References for IP Multicast Optimization: Multicast Subsecond Convergence, on page 379](#)
- [Feature History for IP Multicast Optimization Multicast Subsecond Convergence, on page 379](#)

## 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

- 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.

## 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 PIM Router Query Message Interval

Perform this task to modify the PIM router query message interval.

#### Procedure

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

### Verifying Multicast Subsecond Convergence Configurations

Perform this task to display detailed information about and to verify information regarding the Multicast Subsecond Convergence feature.

**Procedure****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

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



## Additional References for IP Multicast Optimization: Multicast Subsecond Convergence

### Related Documents

| Related Topic                                                                    | Document Title                                                                                                |
|----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| For complete syntax and usage information for the commands used in this chapter. | See the IP Multicast Routing Commands section of the <i>Command Reference (Catalyst 9400 Series Switches)</i> |

## Feature History for IP Multicast Optimization Multicast Subsecond Convergence

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

| Release                     | Feature                                                    | Feature Information                                                                                                                                                                                                   |
|-----------------------------|------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cisco IOS XE Everest 16.6.1 | IP Multicast Optimization: Multicast Subsecond Convergence | 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). |

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <https://cfng.cisco.com/>





## CHAPTER 20

# IP Multicast Optimization: IP Multicast Load Splitting across Equal-Cost Paths

- [Prerequisites for IP Multicast Load Splitting across Equal-Cost Paths, on page 381](#)
- [Information About IP Multicast Load Splitting across Equal-Cost Paths, on page 381](#)
- [How to Load Split IP Multicast Traffic over ECMP, on page 389](#)
- [Configuration Examples for Load Splitting IP Multicast Traffic over ECMP, on page 395](#)
- [Additional References for IP Multicast Optimization: IP Multicast Load Splitting across Equal-Cost Paths, on page 396](#)
- [Feature History for IP Multicast Optimization: IP Multicast Load Splitting across Equal-Cost Paths, on page 396](#)

## 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 Routing Configuration Guide*.

## Information About IP Multicast Load Splitting across Equal-Cost Paths

### 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

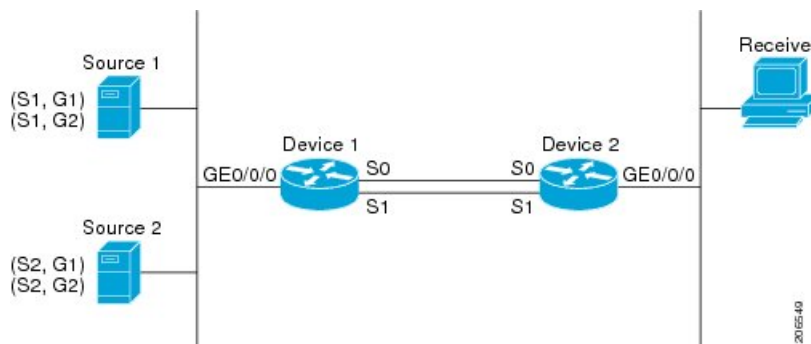
By default, for Protocol Independent Multicast sparse mode (PIM-SM), Source Specific Multicast (PIM-SSM), groups, if multiple equal-cost paths are available, Reverse Path Forwarding (RPF) for IPv4 multicast traffic is based on the PIM neighbor with the highest IP address. This method is referred to as the highest PIM neighbor behavior. This behavior is in accordance with RFC 2362 for PIM-SM, but also applies to PIM-SSM, 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.



**Note** Although the following illustration and example uses routers in the configuration, any device (router or controller) can be used.

**Figure 43: 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 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.

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), the root of the multicast tree is the RP address for the group G; for (S, G) trees (PIM-SM, PIM-SSM), 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* 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* 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* 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

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.

### 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

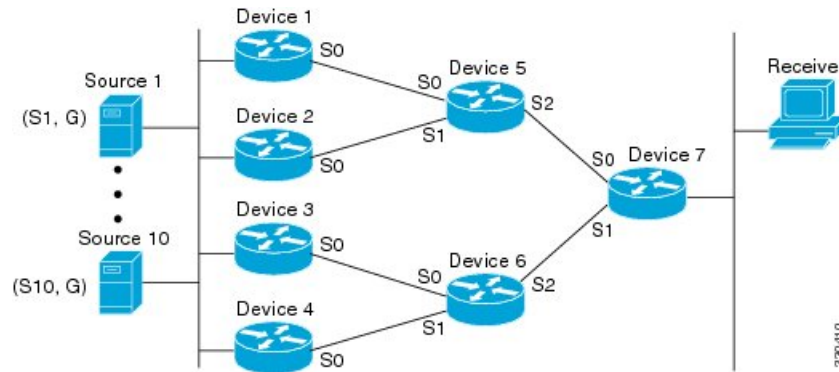
The hash mechanism used in IPv4 multicast to load split multicast traffic by source address or by source and group address is subject to a problem usually referred to as polarization. A by-product of ECMP multicast load splitting based on source address or on source and group address, polarization is 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.



**Note** Although the following illustration and example uses routers in the configuration, any device (router or switch) can be used.

**Figure 44: 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.

## 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* 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 Load Splitting on the PIM Assert Process in PIM-SM and PIM-SSM

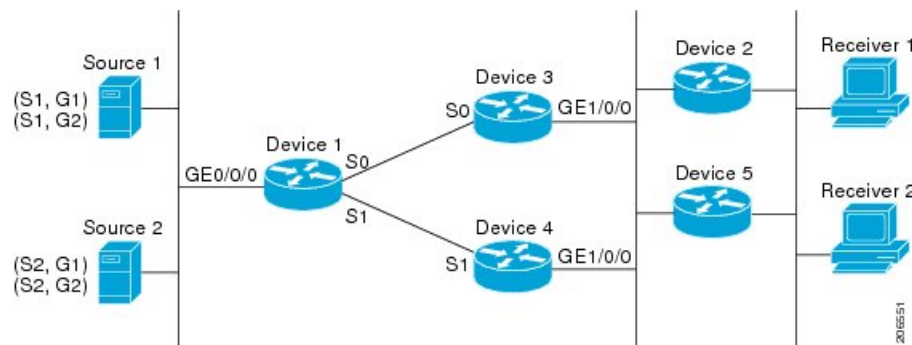
There are also cases where ECMP multicast load splitting with the **ip multicast multipath** command can become ineffective due to the PIM assert process taking over, even when using PIM-SM with (\*, G) or (S, G) forwarding or PIM-SSM with (S, G) forwarding.

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.



**Note** Although the following illustration and example uses routers in the configuration, any device (router or controller) can be used.

**Figure 45: 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* configuration note on the Cisco IOS IP multicast FTP site at <ftp://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

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

- 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, on page 388](#) Use of BGP with ECMP Multicast Load Splitting section.

## Restrictions for IP Multicast Load Splitting -ECMP

- 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.

### Before you begin

- 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* section.

### Procedure

|               | Command or Action                                                                                     | Purpose                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------------|-------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>enable</b><br><b>Example:</b><br><pre>Device&gt; enable</pre>                                      | Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Step 2</b> | <b>configure terminal</b><br><b>Example:</b><br><pre>Device# configure terminal</pre>                 | Enters global configuration mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Step 3</b> | <b>ip multicast multipath</b><br><b>Example:</b><br><pre>Device(config)# ip multicast multipath</pre> | Enables ECMP multicast load splitting based on source address using the S-hash algorithm. <ul style="list-style-type: none"> <li>• 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.</li> <li>• 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.</li> <li>• 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.</li> </ul> |
| <b>Step 4</b> | Repeat step 3 on all the devices in a redundant topology.                                             | --                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Step 5</b> | <b>exit</b><br><b>Example:</b><br><pre>Device(config)# exit</pre>                                     | Exits global configuration mode and returns to privileged EXEC mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

|               | Command or Action                                                                                                               | Purpose                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|---------------|---------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 6</b> | <b>show ip rpf</b> <i>source-address</i> [ <i>group-address</i> ]<br><b>Example:</b><br><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"> <li>• Use this command to verify RPF selection so as to ensure that IP multicast traffic is being properly load split.</li> </ul>                                                                                                                                                                                                                                                                   |
| <b>Step 7</b> | <b>show ip route</b> <i>ip-address</i><br><b>Example:</b><br><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"> <li>• Use this command to verify that there are multiple paths available to a source or RP for ECMP multicast load splitting.</li> <li>• 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).</li> </ul> |

## 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 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.

### Procedure

|               | Command or Action                                                | Purpose                                                                                                            |
|---------------|------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>enable</b><br><b>Example:</b><br><pre>Device&gt; enable</pre> | Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul> |

|               | Command or Action                                                                                                                | Purpose                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------|----------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 2</b> | <b>configure terminal</b><br><br><b>Example:</b><br><br>Device# configure terminal                                               | Enters global configuration mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Step 3</b> | <b>ip multicast multipath s-g-hash basic</b><br><br><b>Example:</b><br><br>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.<br><br><ul style="list-style-type: none"> <li>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.</li> </ul>                                                                                                                                                                                                     |
| <b>Step 4</b> | Repeat Step 3 on all the devices in a redundant topology.                                                                        | --                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Step 5</b> | <b>exit</b><br><br><b>Example:</b><br><br>Device(config)# exit                                                                   | Exits global configuration mode and returns to privileged EXEC mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>Step 6</b> | <b>show ip rpf source-address [group-address]</b><br><br><b>Example:</b><br><br>Device# show ip rpf 10.1.1.2                     | (Optional) Displays the information that IP multicast routing uses to perform the RPF check.<br><br><ul style="list-style-type: none"> <li>Use this command to verify RPF selection so as to ensure that IP multicast traffic is being properly load split.</li> </ul>                                                                                                                                                                                                                                                                 |
| <b>Step 7</b> | <b>show ip route ip-address</b><br><br><b>Example:</b><br><br>Device# show ip route 10.1.1.2                                     | (Optional) Displays the current state of the IP routing table.<br><br><ul style="list-style-type: none"> <li>Use this command to verify that there are multiple paths available to a source or RP for ECMP multicast load splitting.</li> <li>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).</li> </ul> |

## 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.

### Procedure

|               | Command or Action                                                                                                                                     | Purpose                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>enable</b><br><b>Example:</b><br><pre>Device&gt; enable</pre>                                                                                      | Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Step 2</b> | <b>configure terminal</b><br><b>Example:</b><br><pre>Device# configure terminal</pre>                                                                 | Enters global configuration mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Step 3</b> | <b>ip multicast multipath s-g-hash next-hop-based</b><br><b>Example:</b><br><pre>Device(config)# ip multicast multipath s-g-hash next-hop-based</pre> | Enables ECMP multicast load splitting based on source, group, and next-hop-address using the next-hop-based S-G-hash algorithm. <ul style="list-style-type: none"> <li>• 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.</li> </ul> <p><b>Note</b> Be sure to enable the <b>ip multicast multipath</b> 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> |
| <b>Step 4</b> | Repeat Steps 1 through 3 on all the routers in a redundant topology.                                                                                  | --                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Step 5</b> | <b>end</b><br><b>Example:</b><br><pre>Device(config)# end</pre>                                                                                       | Exits global configuration mode and returns to privileged EXEC mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |



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

## Configuration Examples for Load Splitting IP Multicast Traffic over ECMP

### 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 for IP Multicast Optimization: IP Multicast Load Splitting across Equal-Cost Paths

### Related Documents

| Related Topic                                                                    | Document Title                                                                                                |
|----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| For complete syntax and usage information for the commands used in this chapter. | See the IP Multicast Routing Commands section of the <i>Command Reference (Catalyst 9400 Series Switches)</i> |

### Standards and RFCs

| Standard/RFC    | Title                                                                                       |
|-----------------|---------------------------------------------------------------------------------------------|
| <i>RFC 4601</i> | <a href="#">Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification</a> |

## Feature History for IP Multicast Optimization: IP Multicast Load Splitting across Equal-Cost Paths

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

| Release                     | Feature                                                                  | Feature Information                                                                                                                                                                                                                                                                                                   |
|-----------------------------|--------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cisco IOS XE Everest 16.6.1 | IP Multicast Optimization: Load Splitting IP Multicast Traffic over ECMP | 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. |

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <https://cfng.cisco.com/>





## CHAPTER 21

# IP Multicast Optimization: SSM Channel Based Filtering for Multicast

- [Prerequisites for SSM Channel Based Filtering for Multicast Boundaries, on page 399](#)
- [Information About the SSM Channel Based Filtering for Multicast Boundaries, on page 399](#)
- [How to Configure SSM Channel Based Filtering for Multicast Boundaries, on page 400](#)
- [Configuration Examples for SSM Channel Based Filtering for Multicast Boundaries, on page 401](#)
- [Additional References for IP Multicast Optimization: SSM Channel-Based Filtering for Multicast, on page 402](#)
- [Feature History for IP Multicast Optimization: SSM Channel Based Filtering for Multicast, on page 403](#)

## Prerequisites for SSM Channel Based Filtering for Multicast Boundaries

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 the SSM Channel Based Filtering for Multicast Boundaries

This section provides information about the SSM channel based filtering for multicast boundaries feature.

### Rules for Multicast Boundaries

The SSM Channel Based Filtering for Multicast Boundaries feature expands the **ip multicast boundary** command for control plane filtering support. More than one **ip multicast boundary** command can be applied to an interface.

The following rules govern the **ip multicast boundary** command:

- One instance of the **in** and **out** keywords can be configured on an interface.
- The **in** and **out** keywords can be used for standard or extended access lists.

- Only standard access lists are permitted with the use of the **filter-autorp** keyword or no keyword.
- A maximum of three instances of a command will be allowed on an interface: one instance of **in**, one instance of **out**, and one instance of **filter-autorp** or no keyword.
- When multiple instances of the command are used, the filtering will be cumulative. If a boundary statement with no keyword exists with a boundary statement with the **in** keyword, both access lists will be applied on the in direction and a match on either one will be sufficient.
- All instances of the command apply to both control and data plane traffic.
- Protocol information on the extended access list is parsed to allow reuse and filtering for consistency. An (S,G) operation will be filtered by an extended access list under all conditions stated above for keywords if the access list filters (S,G) traffic for all protocols.

## Benefits of SSM Channel Based Filtering for Multicast Boundaries

- This feature allows input on the source interface.
- The access control capabilities are the same for SSM and Any Source Multicast (ASM).

## How to Configure SSM Channel Based Filtering for Multicast Boundaries

This section provides steps for configuring SSM channel based filtering for multicast boundaries.

### Configuring Multicast Boundaries

#### Procedure

|               | Command or Action                                                                                                               | Purpose                                          |
|---------------|---------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| <b>Step 1</b> | <b>enable</b><br><b>Example:</b><br><pre>Device&gt; enable</pre>                                                                | Enables privileged EXEC mode.                    |
| <b>Step 2</b> | <b>configure terminal</b><br><b>Example:</b><br><pre>Device# configure terminal</pre>                                           | Enters global configuration mode.                |
| <b>Step 3</b> | <b>ip access-list {standard  extended} access-list-name</b><br><b>Example:</b><br><pre>Device(config)# ip access-list 101</pre> | Configures the standard or extended access list. |

|               | Command or Action                                                                                                                                                                  | Purpose                                                                                                                    |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| <b>Step 4</b> | <b>permit</b> <i>protocol</i> <b>host</b> <i>address</i> <b>host</b> <i>address</i><br><b>Example:</b><br>Device(config-ext-nacl)# permit ip host 181.1.2.201 host 232.1.1.11      | Permits specified ip host traffic.                                                                                         |
| <b>Step 5</b> | <b>deny</b> <i>protocol</i> <b>host</b> <i>address</i> <b>host</b> <i>address</i><br><b>Example:</b><br>Device(config-acl-nacl)# deny ip host 181.1.2.203 host 232.1.1.1           | Denies specified multicast ip group and source traffic.                                                                    |
| <b>Step 6</b> | Repeat Step 4 or Step 5 as needed.                                                                                                                                                 | Permits and denies specified host and source traffic.                                                                      |
| <b>Step 7</b> | <b>interface</b> <i>type</i> <b>interface-number</b> <i>port</i> <i>-number</i><br><b>Example:</b><br>Device(config)# interface gigabitethernet 2/3/0                              | Enables interface configuration mode.                                                                                      |
| <b>Step 8</b> | <b>ip multicast boundary</b> <i>access-list-name</i> [ <b>in</b>   <b>out</b>   <b>filter-autorp</b> ]<br><b>Example:</b><br>Device(config-if)# ip multicast boundary acc_grp1 out | Configures the multicast boundary.<br><b>Note</b> The <b>filter-autorp</b> keyword does not support extended access lists. |

## Configuration Examples for SSM Channel Based Filtering for Multicast Boundaries

This section provides configuration examples of SSM Channel Based filtering for multicast boundaries.

### Configuring the Multicast Boundaries Permitting and Denying Traffic Example

The following example permits outgoing traffic for (181.1.2.201, 232.1.1.1) and (181.1.2.202, 232.1.1.1) and denies all other (S,G)s.

```
configure terminal
ip access-list extended acc_grp1
permit ip host 0.0.0.0 232.1.1.1 0.0.0.255
permit ip host 181.1.2.201 host 232.1.1.1
permit udp host 181.1.2.202 host 232.1.1.1
permit ip host 181.1.2.202 host 232.1.1.1
```

```
deny igmp host 181.2.3.303 host 232.1.1.1
interface gigabitethernet 1/0/1
ip multicast boundary acc_grp1 out
```

## Configuring the Multicast Boundaries Permitting Traffic Example

The following example permits outgoing traffic for (192.168.2.201, 232.1.1.5) and 192.168.2.202, 232.1.1.5).

```
configure terminal
ip access-list extended acc_grp6
permit ip host 0.0.0.0 232.1.1.1 5.0.0.255
deny udp host 192.168.2.201 host 232.1.1.5
permit ip host 192.168.2.201 host 232.1.1.5
deny pim host 192.168.2.201 host 232.1.1.5
permit ip host 192.168.2.202 host 232.1.1.5
deny igmp host 192.2.3.303 host 232.1.1.1
interface gigabitethernet 1/0/1
ip multicast boundary acc_grp6 out
```

## Configuring the Multicast Boundaries Denying Traffic Example

The following example denies a group-range that is announced by the candidate RP. Because the group range is denied, no pim auto-rp mappings are created.

```
configure terminal
ip access-list standard acc_grp10
deny 225.0.0.0 0.255.255.255
permit any
access-list extended acc_grp12
permit pim host 181.1.2.201 host 232.1.1.8
deny udp host 181.1.2.201 host 232.1.1.8
permit pim host 181.1.2.203 0.0.0.255 host 227.7.7.7
permit ip host 0.0.0.0 host 227.7.7.7
permit ip 181.1.2.203 0.0.0.255 host 227.7.7.7
permit ip host 181.1.2.201 host 232.1.1.7
ip access-list extended acc_grp13
deny ip host 181.1.2.201 host 232.1.1.8
permit ip any any
interface gigabitethernet 1/0/1
ip multicast boundary acc_grp10 filter-autorp
ip multicast boundary acc_grp12 out
ip multicast boundary acc_grp13 in
```

## Additional References for IP Multicast Optimization: SSM Channel-Based Filtering for Multicast

### Related Documents

| Related Topic                                                                    | Document Title                                                                                                  |
|----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| For complete syntax and usage information for the commands used in this chapter. | See the IP Multicast Routing Commands section of the <i>Command Reference (Catalyst 9400 Series Switches)</i> . |



## Feature History for IP Multicast Optimization: SSM Channel Based Filtering for Multicast

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

| Release                     | Feature                                                              | Feature Information                                                                                                                                                                                                         |
|-----------------------------|----------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cisco IOS XE Everest 16.6.1 | IP Multicast Optimization: SSM Channel Based Filtering for Multicast | The SSM Channel Based Filtering for Multicast Boundaries feature expands the ip multicast boundary command for control plane filtering support. More than one ip multicast boundary command can be applied to an interface. |

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <https://cfng.cisco.com/>





## CHAPTER 22

# IP Multicast Optimization: IGMP State Limit

- [Prerequisites for IGMP State Limit, on page 405](#)
- [Restrictions for IGMP State Limit, on page 405](#)
- [Information About IGMP State Limit, on page 405](#)
- [How to Configure IGMP State Limit, on page 407](#)
- [Configuration examples for IGMP State Limit, on page 408](#)
- [Additional References, on page 410](#)
- [Feature History for IP Multicast Optimization: IGMP State Limit, on page 410](#)

## Prerequisites for IGMP State Limit

- IP multicast is enabled and the Protocol Independent Multicast (PIM) interfaces are configured using the tasks described in the "Configuring Basic IP Multicast" module of the *IP Multicast: PIM Configuration Guide*.
- ALL ACLs must be configured. For information, see the "Creating an IP Access List and Applying It to an Interface" module of the *Security Configuration Guide: Access Control Lists* guide.

## Restrictions for IGMP State Limit

You can configure only one global limit per device and one limit per interface.

## Information About IGMP State Limit

This section provides information about IGMP state limit.

## 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

- 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.

# How to Configure IGMP State Limit

This section describes how to configure IGMP state limit.

## Configuring IGMP State Limiters

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.

### Configuring Global IGMP State Limiters

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

#### Procedure

|               | Command or Action                                                                          | Purpose                                                                                                                               |
|---------------|--------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>enable</b><br><b>Example:</b><br>Device> enable                                         | Enables privileged EXEC mode.<br>• Enter your password if prompted.                                                                   |
| <b>Step 2</b> | <b>configure terminal</b><br><b>Example:</b><br>Device# configure terminal                 | Enters global configuration mode.                                                                                                     |
| <b>Step 3</b> | <b>ip igmp limit <i>number</i></b><br><b>Example:</b><br>Device(config)# ip igmp limit 150 | Configures a global limit on the number of mroute states resulting from IGMP membership reports (IGMP joins).                         |
| <b>Step 4</b> | <b>end</b><br><b>Example:</b><br>Device(config-if)# end                                    | Ends the current configuration session and returns to privileged EXEC mode.                                                           |
| <b>Step 5</b> | <b>show ip igmp groups</b><br><b>Example:</b><br>Device# show ip igmp groups               | (Optional) Displays the multicast groups with receivers that are directly connected to the device and that were learned through IGMP. |

### Configuring Per Interface IGMP State Limiters

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

**Procedure**

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

## Configuration examples for IGMP State Limit

This section show configuration examples of IGMP state limit.

## Configuring IGMP State Limiters Example

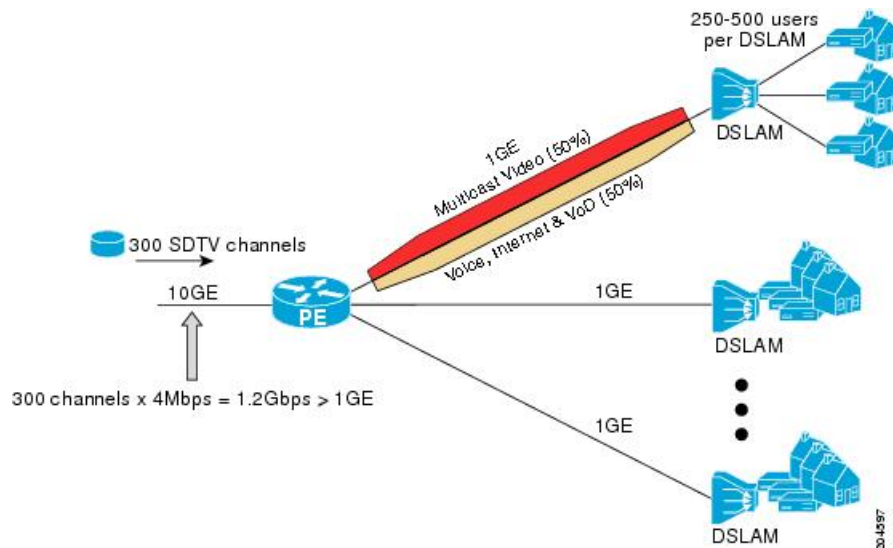
The following example shows how to configure IGMP state limiters to provide multicast CAC in a network environment where all the multicast flows roughly utilize the same amount of bandwidth.

This example uses the topology illustrated in the figure.



**Note** Although the following illustration and example uses routers in the configuration, any device (router or switch) can be used.

**Figure 46: IGMP State Limit Example Topology**



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 router 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 router. 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 0/0/0 for the SD channels and Internet, Voice, and VoD services being offered to subscribers:

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

## Additional References

### Related Documents

| Related Topic                                                                    | Document Title                                                                                                |
|----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| For complete syntax and usage information for the commands used in this chapter. | See the IP Multicast Routing Commands section of the <i>Command Reference (Catalyst 9400 Series Switches)</i> |

## Feature History for IP Multicast Optimization: IGMP State Limit

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

| Release                     | Feature                                     | Feature Information                                                                                                                                                                                                                                                                                    |
|-----------------------------|---------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cisco IOS XE Everest 16.6.1 | IP Multicast Optimization: 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. |

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <https://cfnng.cisco.com/>