Basic Multicast Troubleshooting Tools

Document ID: 13726

Contents

Introduction

This document explains different tools and techniques for troubleshooting multicast networks. If you understand the various command line interface tools and the key information fields in their output, it helps you troubleshoot multicast networks.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

This document is not restricted to specific software and hardware versions.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

Troubleshooting Strategies

When you troubleshoot multicast networks, it is good to consider the signaling protocol used in the network and packet flow. The signaling protocol is used to setup and tear down the multicast sessions (such as PIM dense mode, PIM sparse mode, and DVMRP), and packet flow is the actual sending, replicating, and receiving of the multicast packets between the source and receiver, based on the forwarding table created by the signaling process.

This table helps verify each piece of troubleshooting information by checking that each section of the table is working correctly:

	Source	**Network**	**Receivers**
Signaling	NA	Check Network Signaling	Check Receiver Signaling
Packet Flow	Check Source Packet Flow	Check Network Packet Flow	Check Receiver Packet Flow

The next subsections detail the troubleshooting tools you can use to check and fix common problems.

Check Source Packet Flow

Complete these steps to determine if the source is actually sourcing the packets and inserting the correct packet fields:

1. Check the interface counters on the host. First, check the interface counters (if you are on a UNIX system, use the **netstat** command) on the source host to see if it is sending packets. If it is not, check for misconfiguration or bugs in the host stack and application.
2. Use the **show ip igmp groups** *interface−name* command to check the upstream router to see if it received a join membership report at the interface directly connected to source.
3. Check the TTL value in the application sourcing packets; it should be greater than 1. If the application sends packets with a TTL value less than 1, you should see the traffic dropped at the first upstream router. To verify, use the **show ip traffic** command and look for an increase in the value of the "bad hop count" counter. Any packet with a TTL value of 1, or less than the TTL threshold set by the interface with the **ip multicast ttl−threshold** command, is dropped and the "bad hop−count" counter is increased by one. Use the **show ip igmp interface** *interface−name* command to see the interface TTL threshold value.
4. Use the **show ip mroute count** and **show ip mroute active** commands to check the first upstream router or switch to see if it sees multicast packets from the source. The command output shows the traffic flow statistics for each (S,G) pair. If you do not observe any traffic, check receiver signaling.

5. Use the **debug ip mpacket** command on the nearest upstream router, with the *detail* or *acl* argument for granularity. Use this command with caution when there is heavy multicast traffic on the network. Only if necessary, use the **debug ip mpacket** command on the route. Use the *detail* argument to show packet headers in the **debug** output, and access lists to check for traffic from specific sources. Remember that this command can have a serious performance impact on other traffic, so use it with caution.

Check Network Signaling

This is the most complex and important piece of troubleshooting in any network. It depends on the network signaling protocol used, such as PIM sparse mode, PIM dense mode, and DVMRP. We recommend the multi−step approach described in this section.

Troubleshooting PIM Sparse Mode

Complete these steps to troubleshoot PIM sparse mode:

1. Check that IP multicast routing is enabled on all multicast routers.
2. Use the **show ip pim neighbor** command to check the expiration timer and mode to ensure sucessful PIM neighbor establishment, and look for any possible connectivity and timer issues that might inhibit the establishment of PIM neighbors. If necessary, use the **ip pim [version] [dense−mode] [sparse−mode] [sparse−dense−mode] interface level** subcommand to set the correct mode and version to successfully establish the PIM neighbors.
3. Use the **show ip pim rp mapping** command to ensure the correct RP−Group mapping and to check the expiration timer if auto−RP is configured. Use the **debug ip pim auto−rp** command to help figure out any auto−RP failures. If you do not see any PIM Group−to−RP Mappings, check the auto−RP configuration, or configure static Group−RP mappings with the **ip pim rp−address** *ip address of RP* **[access−list] [named−accesslist] [override]** command. The auto−RP configuration can be performed with the **ip pim send−rp−announce** *interface−id* **scope** *TTL value* and **ip pim send−rp−discovery** *interface−id* **scope** *TTL value* commands. These commands have to be configured only if there are auto−RP configurations.
4. Use the **show ip rpf** *ip address of source* command to check the RPF failure for the source address. PIM dense mode and PIM sparse mode send Prune messages back to the source if traffic arrives on a non−RPF point−to−point interface. The **debug ip pim** command helps identify possible reasons for a failure in a PIM network it compares the typical output with what you see. Use this output to identify the three discrete stages in PIM sparse mode: joining, registering, and SPT−switchover. The **show ip mroute** command allows you to watch the null entries in the Outgoing Interface lists and pruned entries in the mroute table.

Check Network Packet Flow

Use these commands to check the flow of multicast packets across the network:

- multicast trace hop−by−hop using the **mtrace** command
- **mstat**
- **ping**
- **show ip mroute count**
- **show ip mroute active**
- **debug ip mpacket**

Check Receiver Signaling

Complete these steps to check receiver signaling:

1. Use the **show ip igmp groups** command at the first upstream router connected to the receiver to check that the interface has joined the group.
2. Use the **ping** command to check the reachability of the host and the first upstream router.
3. Use the **show ip igmp interface** command to check the IGMP version of the interface.

   Note: Remember that a router configured with IGMP version 1 considers IGMP version 2 packets received from the host as invalid. These IGMP packets do not join the group until the router receives an IGMP version 1 packet from the host.
4. Use the **debug ip igmp** command to further troubleshoot receiver signaling.

Check Receiver Packet Flow

Complete these steps to check the receiver packet flow:

1. Use the **netstat** command on a UNIX system to check the receiver interface statistics.
2. Check that the TCP/IP stack was installed and configured properly.
3. Check that the Multicast receiver client application was installed and configured properly.
4. Watch for duplicate multicast packets on a multiaccess segment.

Power Tools

The commands in this section can also be useful when troubleshooting, especially when you test the network packet flow and find the points of failure in the multicast network. For more extensive information on multicast tool commands, refer to the IP Multicast Tools Commands.

mstat

This command shows the multicast path in ASCII graphic format. It traces the path between any two points in the network, shows drops and duplicates, TTLs, and delays at each node in the network. It is very useful when you need to locate congestion points in the network, or focus on a router with high drop/duplicate counts. Duplicates are indicated in the output as "negative" drops.

```
Router# mstat lwei-home-ss2 171.69.58.88 224.0.255.255
Type escape sequence to abort
Mtrace from 171.69.143.27 to 171.69.58.88 via group 224.0.255.255
>From source (lwei-home-ss2.cisco.com) to destination (lwei-ss20.cisco.com)
Waiting to accumulate statistics......
Results after 10 seconds:

  Source        Response Dest    Packet Statistics For    Only For Traffic
171.69.143.27    171.69.62.144    All Multicast Traffic     From 171.69.143.27
     |       __/  rtt 48   ms   Lost/Sent = Pct  Rate     To 224.0.255.255
     v      /     hop 48   ms   ---------------------     --------------------
171.69.143.25   lwei-cisco-isdn.cisco.com
     |     ^      ttl   1
     v     |      hop 31   ms    0/12 = 0%      1 pps    0/1 = --%  0 pps
171.69.121.84
171.69.121.45   eng-frmt12-pri.cisco.com
     |     ^      ttl   2
     v     |      hop -17  ms    -735/12 = --%      1 pps    0/1 = --%  0 pps
171.69.121.4
171.69.5.27     eng-cc-4.cisco.com
     |     ^      ttl   3
     v     |      hop -21  ms    -678/23 = --%      2 pps    0/1 = --%  0 pps
171.69.5.21
171.69.62.130   eng-ios-2.cisco.com
     |     ^      ttl   4
     v     |      hop 5    ms    605/639 = 95%      63 pps    1/1 = --%  0 pps
```

```
171.69.62.144
171.69.58.65    eng-ios-f-5.cisco.com
     |      \__   ttl   5
     v         \  hop 0    ms        4         0 pps           0    0 pps
171.69.58.88    171.69.62.144
  Receiver      Query Source
```

mrinfo

This command shows multicast neighbor router information, router capabilities and code version, multicast interface information, TTL thresholds, metrics, protocol, and status. It is useful when you need to verify multicast neighbors, confirm that bi-directional neighbor adjacency exists, and verify that tunnels are up in both directions.

```
Router# mrinfo
  192.1.7.37 (b.cisco.com) [version cisco 11.1] [flags: PMSA]:
  192.1.7.37 -> 192.1.7.34 (s.cisco.com) [1/0/pim]
  192.1.7.37 -> 192.1.7.47 (d.cisco.com) [1/0/pim]
  192.1.7.37 -> 192.1.7.44 (d2.cisco.com) [1/0/pim]
  131.9.26.10 -> 131.9.26.9 (su.bbnplanet.net) [1/32/pim]
```

The flags in the output indicate:

- P = prune-capable
- M = mtrace-capable
- S = SNMP-capable
- A = Auto-RP-capable

mtrace

This command shows the multicast path from the source to the receiver, and it traces the path between points in the networks, which shows TTL thresholds and delay at each node. When troubleshooting, use the **mtrace** command to find where multicast traffic flow stops, to verify the path of multicast traffic, and to identify sub-optimal paths.

```
Router# mtrace 171.69.215.41 171.69.215.67 239.254.254.254
Type escape sequence to abort.
Mtrace from 171.69.215.41 to 171.69.215.67 via group 239.254.254.254
From source (?) to destination (?)
Querying full reverse path...
0  171.69.215.67
-1  171.69.215.67 PIM  thresh^ 0  0 ms
-2  171.69.215.74 PIM  thresh^ 0  2 ms
-3  171.69.215.57 PIM  thresh^ 0  894 ms
-4  171.69.215.41 PIM  thresh^ 0  893 ms
-5  171.69.215.12 PIM  thresh^ 0  894 ms
-6  171.69.215.98 PIM  thresh^ 0  893 ms
```

ping

When troubleshooting, the **ping** command is the easiest way to generate multicast traffic in the lab to test the multicast tree because it pings all members of the group, and all members respond.

```
R3# ping 239.255.0.1
 Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 239.255.0.1, timeout is 2 seconds:
Reply to request 0 from 172.16.12.2, 16 ms
Reply to request 0 from 172.16.7.2, 20 ms
```

show Commands

The commands in this section help you gather useful information when troubleshooting a multicast problem. Refer to the IP Multicast Command Reference Guide for more extensive information on these **show** commands.

Tip: If your **show** command responses are sluggish, the most probable reason is that router currently performs an IP domain lookup for IP addresses in the **show** command. You can disable IP domain lookup You can use the **no ip domain−lookup** command, under the router global configuration mode, to disable IP domain lookup. This stops the IP domain lookup and increases the **show** command output speed.

show ip igmp groups

This command shows which multicast groups are directly connected to the router, and which are learned via Internet Group Management Protocol (IGMP). You can use this command to verify that a source or receiver has actually joined the target group on the router interface. The "Last Reporter" column shows only one IGMP host, which indicates that it has sent either an unsolicited IGMP Join or IGMP Report in response to a IGMP Query from the PIM router for that particular group. You should only see one "Last Reporter" per Group Address.

```
R1# show ip igmp groups
IGMP Connected Group Membership
Group Address    Interface      Uptime          Expires         Last Reporter
239.255.0.1      Ethernet1      00:10:54        00:01:10        192.168.9.1
224.0.1.40       Ethernet0      01:36:27        00:02:45        192.168.10.2
224.0.1.40       Ethernet1      01:48:15        never           192.168.9.3
```

show ip igmp interface

Use this command to display multicast−related information about an interface, and to verify that IGMP is enabled, the correct version is running, the timers, Time To Live (TTL) threshold value, and IGMP querier router are properly set. IGMP does not need to be configured on an interface. It is enabled by default when you configure **ip pim** *dense−mode*|*sparse−mode*|*sparse−dense−mode* .

```
R1# show ip igmp interface
Ethernet1 is up, line protocol is up
  Internet address is 192.168.9.3/24
  IGMP is enabled on interface
  Current IGMP version is 2
  CGMP is disabled on interface
  IGMP query interval is 60 seconds
  IGMP querier timeout is 120 seconds
  IGMP max query response time is 10 seconds
  Last member query response interval is 1000 ms
  Inbound IGMP access group is not set
  IGMP activity: 22 joins, 18 leaves
  Multicast routing is enabled on interface
  Multicast TTL threshold is 0
  Multicast designated router (DR) is 192.168.9.5
  IGMP querying router is 192.168.9.3 (this system)
  Multicast groups joined (number of users):
      224.0.1.40(1)
```

show ip pim neighbor

Use this command to list the Protocol Independent Multicast (PIM) neighbors discovered by the Cisco IOS® Software.

```
R1# show ip pim neighbor
PIM Neighbor Table
Neighbor          Interface               Uptime/Expires    Ver   DR
Address                                                           Prio/Mode
10.10.10.1        Ethernet0/0             02:19:41/00:01:38 v2    1 / DR B S
```

Details of each field are explained here:

- **Neighbor Address** − Specifies a PIM neighbor's IP address
- **Interface** − An interface where a PIM neighbor was discovered
- **Uptime** − The total uptime of neighbor
- **Expires** − The time before a neighbor is timed out and until next PIM hello is received
- **Ver** − The version of PIM on neighbor's interface
- **DR Prio** − The possible values are 0 to 4294967294 or "N"

  This is a new column which tracks the priority of a PIM interface for DR election. The feature to configure a DR based on highest priority versus highest IP address was introduced in Cisco IOS Software Releases 12.1(2)T and 12.2 and Cisco IOS images with Bidir−PIM. You can use the **ip pim dr−priority <0−4294967294>** interface command to set the DR priority. The default DR priority is set to 1. For interoperability, if a PIM neighbor is running an older Cisco IOS version which does not support the DR priority feature, the "DR Prior" column shows as "N". If the neighbor is the only router showing "N" for the interface, it becomes the DR regardless of which router actually has the highest IP address. If there are serveral PIM neighbors with "N" listed under this column, the tie breaker is the highest IP address among them.
- **Mode** − Information about the DR and other PIM capabilities.

  This column lists the DR in addition to any capabilities supported by the PIM neighbor:

  DR − The PIM neighbor is Designated Router

  B − Bidirectional PIM (Bidir−PIM) capable

  S − State refresh capable (applies only for dense mode)

When you troubleshoot, use this command to verify that all neighbors are up and that they use the proper mode, version, and expiration timer. You can also check the router configuration, or use the **show ip pim interface** command to verify the mode (PIM sparse or dense mode). Use the **debug ip pim** command to observe the pim−query message exchange.

show ip pim interface

Use this command to display information about interfaces configured for PIM. In addition, you can use this command to verify that the correct PIM mode (dense or sparse) is configured on the interface, the neighbor count is correct, and the designated router (DR) is correct (which is critical for PIM sparse mode). Multi−access segments (such as Ethernet, Token Ring, FDDI) elect a DR based on highest IP address. Point−to−Point links do not display DR information.

```
R1# show ip pim interface
Address          Interface          Version/Mode    Nbr   Query     DR
                                                    Count Intvl
192.168.10.1     Ethernet0          v2/Sparse−Dense  1    30     192.168.10.2
192.168.9.3      Ethernet1          v2/Sparse−Dense  1    30     192.168.9.5
```

show ip mroute summary

Use this command to display the summarized contents of the IP multicast routing table. You can also use it to verify the active multicast group(s) and which multicast senders are active by looking at the timers and flags.

```
R1## show ip mroute summary
IP Multicast Routing Table
Flags: D − Dense, S − Sparse, 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 − Advertised via MSDP
Outgoing interface flags: H − Hardware switched
Timers: Uptime/Expires
Interface state: Interface, Next−Hop or VCD, State/Mode

(*, 239.255.0.1), 01:57:07/00:02:59, RP 192.168.7.2, flags: SJCF
  (133.33.33.32, 239.255.0.1), 01:56:23/00:02:59, flags: CJT
  (192.168.9.1, 239.255.0.1), 01:57:07/00:03:27, flags: CFT

(*, 224.0.1.40), 1d00h/00:00:00, RP 192.168.7.2, flags: SJPCL
```

show ip mroute

Use this command to display the full contents of the IP multicast routing table. When you troubleshoot, use this command to verify:

- The (S,G) and (*,G) state entries from the flags.
- The incoming interface is correct. If it is not, check the unicast routing table.
- The outgoing interface(s) is correct. If it is incorrectly pruned, check the state in the downstream router.

```
R1# show ip mroute
IP Multicast Routing Table
Flags: D − Dense, S − Sparse, 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 − Advertised via MSDP
Outgoing interface flags: H − Hardware switched
Timers: Uptime/Expires
Interface state: Interface, Next−Hop or VCD, State/Mode

(*, 239.255.0.1), 01:55:27/00:02:59, RP 192.168.7.2, flags: SJCF
  Incoming interface: Ethernet0, RPF nbr 192.168.10.2
  Outgoing interface list:
    Ethernet1, Forward/Sparse, 01:55:27/00:02:52

(133.33.33.32, 239.255.0.1), 01:54:43/00:02:59, flags: CJT
  Incoming interface: Ethernet0, RPF nbr 192.168.10.2
  Outgoing interface list:
    Ethernet1, Forward/Sparse, 01:54:43/00:02:52

(192.168.9.1, 239.255.0.1), 01:55:30/00:03:26, flags: CFT
  Incoming interface: Ethernet1, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0, Forward/Sparse, 01:55:30/00:03:12

(*, 224.0.1.40), 1d00h/00:00:00, RP 192.168.7.2, flags: SJPCL
  Incoming interface: Ethernet0, RPF nbr 192.168.10.2
  Outgoing interface list: Null
```

show ip mroute active

Use this command to display the active traffic sources and groups above the threshold. When you troubleshoot, use it to verify active source groups, the traffic rate for each source group (S,G) pair (you must have switched to Shortest Path Tree (SPT)), and to check if the target group multicast traffic is being received. If the traffic is not being received, look for active traffic starting from the source towards the receiver.

```
R1# show ip mroute active
Active IP Multicast Sources − sending >= 4 kbps

Group: 239.255.0.1, (?)
   Source: 133.33.33.32 (?)
   Rate: 10 pps/115 kbps(1sec), 235 kbps(last 23 secs), 87 kbps(life avg)
```

show ip rpf

Use this command to display how IP multicast routing does Reverse Path Forwarding (RPF). When you troubleshoot, use it to verify that the RPF information is correct. If it is not, check the unicast routing table for the source address. Also use the **ping** and **trace** commands on the source address to verify that unicast routing works. You might need to use Distance Vector Multicast Routing Protocol (DVMRP) routes or static mroutes to fix any unicast−multicast inconsistencies.

```
R1# show ip rpf 133.33.33.32
RPF information for ? (133.33.33.32)
  RPF interface: Ethernet0
  RPF neighbor: ? (192.168.10.2)
  RPF route/mask: 133.33.0.0/16
  RPF type: unicast (eigrp 1)
  RPF recursion count: 0
  Doing distance−preferred lookups across tables
```

show ip mcache

This command can verify the IP multicast fast switching cache and debug fast−switching bugs.

```
R1# show ip mcache
IP Multicast Fast−Switching Cache
(133.33.33.32/32, 239.255.0.1), Ethernet0, Last used: 00:00:00
  Ethernet1       MAC Header: 01005E7F000100000C13DBA90800
(192.168.9.1/32, 239.255.0.1), Ethernet1, Last used: 00:00:00
  Ethernet0       MAC Header: 01005E7F000100000C13DBA80800
```

show ip mroute count

Use this command to verify that multicast traffic is received and to check on its flow rates and drops. If no traffic is received, work from the source to the receiver until you find where the traffic stops. You can also use this command to verify that traffic is being forwarded. If it is not, use the **show ip mroute** command to look for "Null Outgoing interface list" and RPF failures.

```
R1# show ip mroute count
   IP Multicast Statistics
    routes using 2406 bytes of memory
   2 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: 239.255.0.1, Source count: 2, Group pkt count: 11709
  RP−tree: Forwarding: 3/0/431/0, Other: 3/0/0
  Source: 133.33.33.32/32, Forwarding: 11225/6/1401/62, Other: 11225/0/0
  Source: 192.168.9.1/32, Forwarding: 481/0/85/0, Other: 490/0/9
```

```
Group: 224.0.1.40, Source count: 0, Group pkt count:
```

show ip route

Use this command to check the unicast routing table and fix the RPF failures in the mroute table.

```
R2# show ip route
   Codes: C − connected, S − static, I − IGRP, R − RIP, M − mobile, B − BGP
       D − EIGRP, EX − EIGRP external, O − OSPF, IA − OSPF inter area
       N1 − OSPF NSSA external type 1, N2 − OSPF NSSA external type 2
       E1 − OSPF external type 1, E2 − OSPF external type 2, E − EGP
       i − IS−IS, L1 − IS−IS level−1, L2 − IS−IS level−2, ia − IS−IS inter area
       * − candidate default, U − per−user static route, o − ODR
       P − periodic downloaded static route
Gateway of last resort is not set
D    192.168.9.0/24 [90/307200] via 192.168.10.1, 00:59:45,    Ethernet0
   C    192.168.10.0/24 is directly connected, Ethernet0
   D    192.168.4.0/24 [90/11040000] via 192.168.7.1, 23:21:00,    Serial0
   D    192.168.5.0/24 [90/11023872] via 192.168.7.1, 23:21:02,    Serial0
   C    192.168.7.0/24 is directly connected, Serial0
   D    133.33.0.0/16 [90/2195456] via 192.168.7.1, 1d23h, Serial0
   D    192.168.1.0/24 [90/11552000] via 192.168.7.1, 22:41:27,    Serial0
```

show ip pim rp mapping

Use this command to check the RP assignment by multicast group range, and to verify that the source of RP learning (static or auto−RP) and the mapping are correct. If you find an error, check the local router configuration or auto−RP configuration.

```
R1# show ip pim rp mapping
   PIM Group−to−RP Mappings
Group(s) 224.0.1.40/32
     RP 192.168.7.2 (?), v1
       Info source: local, via Auto−RP
            Uptime: 2d00h, expires: never
   Group(s): 224.0.0.0/4, Static
       RP: 192.168.7.2 (?)
```

debug Commands

This section is designed to show you how certain **debug** command outputs should look in a functioning network. When you troubleshoot, you can distinguish between "correct" **debug** output and that which points to a problem in your network. For more extensive information on these **debug** commands, refer to the Cisco IOS Debug Command Reference.

debug ip igmp

Use the **debug ip igmp** command to display IGMP packets received and transmitted, as well as IGMP−host related events. The **no** form of this command disables debug output.

This output helps you discover whether the IGMP processes function. In general, if IGMP does not work, the router process never discovers another host on the network that is configured to receive multicast packets. In PIM dense mode, this means the packets are delivered intermittently (a few every three minutes). In PIM sparse mode, they are never delivered.

```
R1# debug ip igmp
   12:32:51.065: IGMP: Send v2 Query on Ethernet1 to 224.0.0.1
12:32:51.069: IGMP: Set report delay time to 9.4 seconds for 224.0.1.40 on Ethernet1
```

```
12:32:56.909: IGMP: Received v1 Report from 192.168.9.1 (Ethernet1) for 239.255.0.1
12:32:56.917: IGMP: Starting old host present timer for 239.255.0.1 on Ethernet1
12:33:01.065: IGMP: Send v2 Report for 224.0.1.40 on Ethernet1
12:33:01.069: IGMP: Received v2 Report from 192.168.9.4 (Ethernet1) for 224.0.1.40
12:33:51.065: IGMP: Send v2 Query on Ethernet1 to 224.0.0.1
```

The output above shows that the router sends an IGMP version 2 query out interface Ethernet 1 at multicast address 224.0.0.1 (All multicast systems on this subnet). Interface Ethernet 1 itself is a member of group 224.0.1.40 (you can use the **show ip igmp interface** command to determine this), which sets a report delay time of 9.4 seconds (randomly determined). Because it does not receive any report from another system for multicast group 224.0.1.40 for the next 9.4 seconds, it sends a version 2 report of its membership, which is received by the router itself on Ethernet 1. It also receives IGMP report version 1 from host 192.168.9.1, which is directly connected to the interface Ethernet 1 for group 239.255.0.1.

This **debug** output is useful when you verify that the router interface sends queries and to determine the query interval (in the above case, 60 seconds). You can also use the command to determine the version of IGMP used by the clients.

debug ip mpacket

Use the **debug ip mpacket** command to display all received and transmitted IP multicast packets. The **no** form of this command disables debug output.

```
R1# debug ip mpacket 239.255.0.1 detail
   13:09:55.973: IP: MAC sa=0000.0c70.d41e (Ethernet0), IP last−hop=192.168.10.2
   13:09:55.977: IP: IP tos=0x0, len=892, id=0xD3C1, ttl=12, prot=17
   13:09:55.981: IP: s=133.33.33.32 (Ethernet0) d=239.255.0.1 (Ethernet1) len 906, mforwar
```

This command decodes the multicast packet and shows whether the packet is forwarded (mforward) or dropped. It is useful when you debug packet flow problems in the network to look at the TTL value and the reason a packet was dropped.

Caution: Use caution when you turn on packet−level debug output, especially when the router is servicing high multicast packet loads.

debug ip mrouting

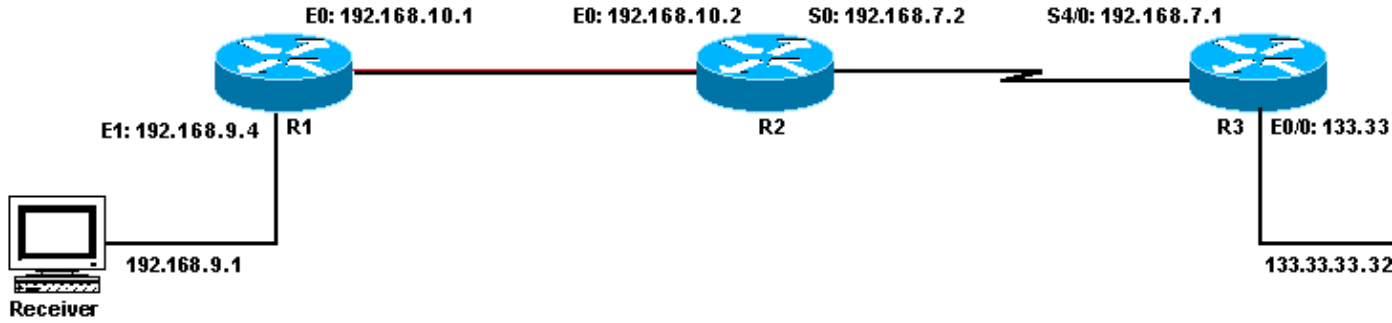
This command is useful for routing table maintenance purposes. Use it to verify that the (S,G) mroute is installed in the mrouting table, or if it is not, why not. The key information in this output is the RPF interface. If there is an RPF check failure, the (S,G) mroute fails to install in the mrouting table.

```
R1# debug ip mrouting 239.255.0.1
13:17:27.821: MRT: Create (*, 239.255.0.1), RPF Null, PC 0x34F16CE
    13:17:27.825: MRT: Create (133.33.33.32/32, 239.255.0.1), RPF Ethernet0/192.168.10.2,
    PC 0x34F181A
    13:17:30.481: MRT: Create (192.168.9.1/32, 239.255.0.1), RPF Ethernet1/0.0.0.0,
    PC 0x34F18
```

debug ip pim

Use the **debug ip pim** command to display PIM packets received and transmitted, as well as PIM related events. The **no** form of this command disables debug output.

This section uses an example to help you understand the debug output of PIM sparse mode, and to show a typical debug output.



Here is the output of **debug ip pim** on R1:

```
R1# debug ip pim
 PIM: Send v2 Hello on Ethernet0
 PIM: Send v2 Hello on Ethernet1
 PIM: Received v2 Hello on Ethernet0 from 192.168.10.2
 PIM: Send v2 Hello on Ethernet0
 PIM: Send v2 Hello on Ethernet1
 PIM: Building Join/Prune message for 239.255.0.1
 PIM: v2, for RP, Join-list: 192.168.7.2/32, RP-bit, WC-bit, S-bit
 PIM: Send v2 periodic Join/Prune to RP via 192.168.10.2 (Ethernet0)
 PIM: Received RP-Reachable on Ethernet0 from 192.168.7.2 for group 239.255.0.1
 PIM: Update RP expiration timer (270 sec) for 239.255.0.1
```

Here is what each line of output denotes: R1 and R2 establish PIM neighbors by exchanging Hello messages. These periodic Hello messages, exchanged at "Query−Interval" seconds between R1 (E0) and R2 (E0), keep track of PIM neighbors.

R1 sends a Join/Prune message to the RP address 192.168.7.2. The RP (R2) replies with a Received RP Reachable message back to R1 for group 239.255.0.1. This in turn updates the RP expiration timer at R1. The expiration timer sets a checkpoint to make sure the RP still exists; otherwise a new RP must be discovered. Use the **show ip pim rp** command to observe the RP expiry time.

Now, look at the **debug** output between R1 and R2 when a multicast receiver for group 239.255.0.1 joins R1.

First, look at the output on R1:

```
1 PIM: Check RP 192.168.7.2 into the (*, 239.255.0.1) entry
 2 PIM: Send v2 Join on Ethernet0 to 192.168.10.2 for (192.16.8.7.2/32, 239.255.0.1), WC-b
 3 PIM: Building batch join message for 239.255.0.1
 4 PIM: Building Join/Prune message for 239.255.0.1
 5 PIM: v2, for RP, Join-list: 192.168.7.2/32, RP-bit, WC-bit, S-bit
 6 PIM: Send v2 periodic Join/Prune to RP via 192.168.10.2 (Ethernet0)
 7 PIM: Received RP-Reachable on Ethernet0 from 192.168.7.2 : for group 239.255.0.1
 8 PIM: Update RP expiration timer (270 sec) for 239.255.0.1
 9 PIM: Building Join/Prune message for 239.255.0.1
 10 PIM: v2, for RP, Join-list: 192.168.7.2/32, RP-bit, WC-bit, S-bit
 11 PIM: Send v2 periodic Join/Prune to RP via 192.168.10.2 (Ethernet0)
```

Now, look at the output on R2:

```
12 PIM: Received v2 Join/Prune on Ethernet0 from 192.168.10.1, to us
13 PIM: Join-list: (*, 239.255.0.1) RP 192.168.7.2
14 PIM: Check RP 192.168.7.2 into the (*, 239.255.0.1) entry, RPT-bit set, WC-bit set, S-b
15 PIM: Add Ethernet0/192.168.10.1 to (*, 239.255.0.1), Forward state
16 PIM: Building Join/Prune message for 239.255.0.1
17 PIM: Received v2 Join/Prune on Ethernet0 from 192.168.10.1, to us
18 PIM: Join-list: (*, 239.255.0.1) RP 192.168.7.2, RPT-bit set, WC-bit set, S-bit set
```

```
19 PIM: Add Ethernet0/192.168.10.1 to (*, 239.255.0.1), Forward state
20 PIM: Building Join/Prune message for 239.255.0.1
21 PIM: Send RP-reachability for 239.255.0.1 on Ethernet0
22 PIM: Received v2 Join/Prune on Ethernet0 from 192.168.10.1, to us
23 PIM: Join-list: (*, 239.255.0.1) RP 192.168.7.2, RPT-bit set, WC-bit set, S-bit set
24 PIM: Add Ethernet0/192.168.10.1 to (*, 239.255.0.1), Forward state
25 PIM: Building Join/Prune message for 239.255.0.1
```

In line 1 above, the multicast receiver for group 239.255.0.1 joins R1. This installs a (*, 239.255.0.1) entry in the mroute table. Then, in line 2, the multicast receiver sends a IGMP Join to R2 (RP) to join the shared tree.

When the IGMP join arrives on R2, R2 installs a (*, 239.255.0.1) mroute, as shown in lines 12 through 15 of the R2 output.

Once R2 installs (*, 239.255.0.1) in its mrouting table, it adds the interface from which it received the Join/Prune message to its Outgoing-interface-list in the forward state. It then sends an RP-reachability message back on the interface on which it received the Join/Prune message. This transaction is shown in lines 15 through 21 of the R2 output.

R1 receives the RP-reachable message for group 239.255.0.1 and updates its expiration timer for RP. This exchange repeats itself once a minute by default and refreshes its multicast forwarding state as shown in lines 7 and 8 of the R1 output.

In the next lines, the **debug** output between R2 (RP) and R3 is seen. The source (directly connected to R3) started to send packets for the group 239.255.0.1.

First, look at the output on R3:

```
1 PIM: Check RP 192.168.7.2 into the (*, 239.255.0.1) entry
2 PIM: Building Join/Prune message for 239.255.0.1
3 PIM: For RP, Join-list: 192.168.7.2/32, RP-bit, WC-bit
4 PIM: Send periodic Join/Prune to RP via 192.168.7.2 (Serial4/0)
5 PIM: Received RP-Reachable on Serial4/0 from 192.168.7.2
6 PIM: Update RP expiration timer (270 sec) for 239.255.0.1
7 PIM: Send Register to 192.168.7.2 for 133.33.33.32, group 239.255.0.1
8 PIM: Send Register to 192.168.7.2 for 133.33.33.32, group 239.255.0.1
9 PIM: Received Join/Prune on Serial4/0 from 192.168.7.2
10 PIM: Join-list: (133.33.33.32/32, 239.255.0.1), S-bit set
11 PIM: Add Serial4/0/192.168.7.2 to (133.33.33.32/32, 239.255.0.1), Forward state
12 PIM: Received Register-Stop on Serial4/0 from 192.168.7.2
13 PIM: Clear register flag to 192.168.7.2 for (133.33.33.32/32, 239.255.0.1)
14 PIM: Received Register-Stop on Serial4/0 from 192.168.7.2
15 PIM: Clear register flag to 192.168.7.2 for (133.33.33.32/32, 239.255.0.1)
```

Here is the output of R2, the RP:

```
16 PIM: Received Join/Prune on Serial0 from 192.168.7.1, to us
17 PIM: Send RP-reachability for 239.255.0.1 on Serial0
18 PIM: Received Register on Serial0 from 192.168.7.1 for 133.33.33.32, group 239.255.0.1
19 PIM: Forward decapsulated data packet for 239.255.0.1 on Ethernet0
10 PIM: Forward decapsulated data packet for 239.255.0.1 on Serial0
21 PIM: Send Join on Serial0 to 192.168.7.1 for (133.33.33.32/32, 239.255.0.1), S-bit
22 PIM: Send Join on Serial0 to 192.168.7.1 for (133.33.33.32/32, 239.255.0.1), S-bit
23 PIM: Send Register-Stop to 192.168.7.1 for 133.33.33.32, group 239.255.0.1
24 PIM: Received Join/Prune on Serial0 from 192.168.7.1, to us
25 PIM: Prune-list: (133.33.33.32/32, 239.255.0.1)
26 PIM: Received v2 Join/Prune on Ethernet0 from 192.168.10.1, to us
27 PIM: Join-list: (*, 239.255.0.1) RP 192.168.7.2, RPT-bit set, WC-bit set, S-bit set
28 PIM: Add Ethernet0/192.168.10.1 to (*, 239.255.0.1), Forward state
29 PIM: Add Ethernet0/192.168.10.1 to (133.33.33.32/32, 239.255.0.1)
30 PIM: Join-list: (133.33.33.32/32, 239.255.0.1), S-bit set
```

```
31 PIM: Add Ethernet0/192.168.10.1 to (133.33.33.32/32, 239.255.0.1), Forward state
32 PIM: Building Join/Prune message for 239.255.0.1
33 PIM: For 192.168.7.1, Join−list: 133.33.33.32/32
34 PIM: For 192.168.10.1, Join−list: 192.168.9.1/32
35 PIM: Send v2 periodic Join/Prune to 192.168.10.1 (Ethernet0)
36 PIM: Send periodic Join/Prune to 192.168.7.1 (Serial0)
37 PIM: Received Join/Prune on Serial0 from 192.168.7.1, to us
38 PIM: Join−list: (*, 239.255.0.1) RP 192.168.7.2, RP−bit set, WC−bit set, S−bit set
39 PIM: Add Serial0/192.168.7.1 to (*, 239.255.0.1), Forward state
40 PIM: Add Serial0/192.168.7.1 to (133.33.33.32/32, 239.255.0.1)
41 PIM: Add Serial0/192.168.7.1 to (192.168.9.1/32, 239.255.0.1)
42 PIM: Join−list: (192.168.9.1/32, 239.255.0.1), S−bit set
43 PIM: Add Serial0/192.168.7.1 to (192.168.9.1/32, 239.255.0.1), Forward state
44 PIM: Join−list: (*, 239.255.0.1) RP 192.168.7.2, RP−bit set, WC−bit set, S−bit set
45 PIM: Add Serial0/192.168.7.1 to (*, 239.255.0.1), Forward state
```

Line 1 above shows that R3, which is directly connected via Ethernet0/0 to the source, receives multicast traffic for group 239.255.0.1. It creates a (* , 239.255.0.1) entry and sends a Join message to the RP.

Lines 16 and 17 show that R2, which is the RP, also receives the Join/Prune message and sends RP reachability information back to R3.

In lines 5 and 6, R3 updates its RP expiration timer after it receives the RP reachable information. Lines 7 and 8 above show that R3 uses its (*,G) entry to send the data to RP encapsulated in a Register packet with the source that initiates transmission to group 239.255.0.1.

Lines 18 to 20 show that R2 received the Register packet, de−encapsulated and forwarded it down the tree with a preexisting (*, 239.255.0.1) entry in route table.

Lines 21 and 29 show that R2 sends a Join message towards R3 and installs an (S,G) (133.33.33.32, 239.255.0.1) entry in the mroute table.

Lines 9 to 11 show that R3 receives the Join message from R2, installs an (S,G) (133.33.33.32,239.255.0.1) entry in mroute table, and puts the interface connected to RP in forward mode, which builds the (S,G) multicast SPT tree toward the source.

In line 23, R2 begins to receive (S,G) traffic down SPT and sends a Register−Stop message (and a Join message) toward the source.

Lines 12 to 15 show that R3 receives the Register−Stop message, clears the register flag, and stops the encapsulation (S,G) traffic.

Periodic Join/Prune messages are exchanged between the RP and R3 to maintain the multicast tree.

Related Information

- **IP Multicast Troubleshooting Guide**
- **Multicast Quick−Start Configuration Guide**
- **IP Multicast Support Page**
- **IP Routed Protocols Support Page**
- **IP Routing Support Page**
- **IP3R: Cisco IOS IP Command Reference, Volume 3 of 3: Multicast, Release 12.2**
- **IPC: Part 3: IP Multicast**
- **Technical Support − Cisco Systems**

Updated: Feb 26, 2008 Document ID: 13726