Troubleshoot Multicast Networks with CLI Tools

Contents

Introduction

Prerequisites

- Requirements
- Components Used
- Conventions
- Troubleshoot Strategies
- Check Source Packet Flow
- Check Network Signaling
- Troubleshoot PIM Sparse Mode
- Check Network Packet Flow
- Check Receiver Signaling
- Check Receiver Packet Flow
- Power CLI Tools
- Mstat
- Mrinfo
- Mtrace
- Ping
- Show Commands
- show ip igmp groups
- show ip igmp interface
- show ip pim neighbor
- show ip pim interface
- show ip mroute summary
- show ip mroute
- show ip mroute active
- show ip rpf
- show ip mcache
- show ip mroute count
- show ip route
- show ip pim rp mapping

Debug Commands

- debug ip igmp
- debug ip mpacket
- debug ip mrouting
- debug ip pim

Related Information

Introduction

This document describes tools and techniques for multicast network troubleshooting.

Prerequisites

Requirements

This document requires familiarity with IP multicast, Internet Group Management Protocol (IGMP), Protocol Independent Multicast (PIM), and Reverse Path Forwarding (RPF).

Components Used

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



Note: The command availability, syntax, and output in this document vary by Cisco platform and software release. The examples use Cisco IOS® and Cisco IOS XE® command output.

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, ensure that you understand the potential impact of any command.

Conventions

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

Troubleshoot Strategies

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

This table organizes the signaling and packet-flow checks for the source, network, and receivers:

	Source	**Network**	**Receivers**
Signaling	N/A	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 resolve common problems.

Check Source Packet Flow

Complete these steps to determine if the source has sent the packets and the correct packet fields are inserted:

1. Check the interface counters on the host. First, check the interface counters (if you are on a UNIX system, run the **netstat** command) on the source host to see if it sends packets. If it is not, check for misconfiguration or bugs in the host stack and application.

2. Run the **show ip igmp groups** **<interface-name>** command to verify the router received an IGMP membership report on that interface.

3. Check the TTL value for the packets in the multicast application. It must be at least 2 to cross one routed hop. If the application sends packets with a TTL value of 1, the first upstream router drops the packets when the TTL expires. To verify, run 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. Run the **show ip igmp interface** **<interface-name>** command to view the interface TTL threshold value.

4. Run the **show ip mroute count** and **show ip mroute active** commands to check the first upstream router or switch to view 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, continue checking the source application, host stack, packet capture, and first-hop router interface.

5. Run the **debug ip mpacket** command on the nearest upstream router, with a multicast group or supported access-list (ACL) filter and the **detail** keyword for additional packet information and granularity.



Note: Run this command with caution when there is heavy multicast traffic on the network. Only if necessary, run the **debug ip mpacket** command on the router. Run the **detail** argument to show

---

packet headers in the **debug** output, and access lists to check for traffic from specific sources. Remember this command can have serious performance impacts on other traffic.

Check Network Signaling

This is the most complex and important piece to troubleshoot in any network. It depends on the network signaling protocol used, such as PIM sparse mode, PIM dense mode, and DVMRP. Cisco recommends the multi-step approach described in this section.

Troubleshoot PIM Sparse Mode

Complete these steps to troubleshoot PIM sparse mode:

1. Check that IP multicast routing is enabled on all multicast routers.

2. Run the **show ip pim neighbor** command to check the expiration timer and mode to ensure successful PIM neighbor establishment, and look for any possible connectivity and timer issues that can inhibit the establishment of PIM neighbors. If necessary, run the interface commands **ip pim version {1 | 2}** **and ip pim {dense-mode | sparse-mode | sparse-dense-mode}** to configure the required PIM version and mode.

3. Run 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. Run the **debug ip pim auto-rp** command to identify any Auto-RP failures. If you do not see any PIM Group-to-RP Mappings, check the auto-RP configuration, or configure static group-to-RP mappings with the **ip pim rp-address <rp-address> [<access-list>] [override]** command. Configure a candidate RP with **ip pim send-rp-announce <interface> scope <ttl> group-list <access-list>** and configure the mapping agent with **ip pim send-rp-discovery <interface> scope <ttl>**. These commands apply only to Auto-RP: configure **send-rp-announce** on a candidate RP and **send-rp-discovery** on a mapping agent.

4. Run the **show ip rpf <isource-address>** command to verify the RPF interface, neighbor, and route for the source address. PIM drops multicast packets that arrive on a non-RPF interface. Additional Assert or Prune processing depends on the PIM mode and interface type. The **debug ip pim** command helps identify possible reasons for a failure in a PIM network. Compare the observed output with the typical output in this document. Run 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:

- Run the **mtrace** command to check multicast trace hop-by-hop
- **mstat**
- **ping**
- **show ip mroute count**
- **show ip mroute active**
- **debug ip mpacket**

Check Receiver Signaling

Complete these steps to check receiver signaling:

1. Run the **show ip igmp groups** command at the first upstream router connected to the receiver to verify the router learned group membership on the receiver-facing interface.

2. Run the **ping** command to check the reachability of the host and the first upstream router.

3. Run the **show ip igmp interface** command to check the IGMP version of the interface.

Note: Ensure the router and receivers use compatible IGMP versions. An IGMPv1 router does not process IGMPv2 membership reports. When an IGMPv2 receiver detects an IGMPv1 querier, it uses IGMPv1 compatibility mode and sends IGMPv1 membership reports.

---

4. Run the **debug ip igmp** command to further troubleshoot receiver signaling.

Check Receiver Packet Flow

Complete these steps to check the receiver packet flow:

1. Run the **netstat** command on a UNIX system to check the receiver interface statistics.

2. Check the TCP/IP stack was installed and configured properly.

3. Check the multicast receiver client application was installed and configured properly.

4. Use a packet capture on the receiver facing interface in order to identify duplicate multicast packets on a multi-access segment.

Power CLI Tools

The commands in this section can also be useful when you troubleshoot, especially when you test the network packet flow and find the points of failure in the multicast network.

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.

This is useful when you need to locate congestion points in the network or focus on a router with high drop/duplicate counts. A negative calculated drop value indicates the downstream interface reports more packets than the upstream interface, which can indicate duplicate packets or counter-sampling differences.

<#root>

```
Router#
```

mstat lwei-home-ss2 172.16.58.88 224.0.255.255

```
Type escape sequence to abort
Mtrace from 172.16.143.27 to 172.16.58.88 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
172.16.143.27    172.16.62.144    All Multicast Traffic     From 172.16.143.27
     |       __/  rtt 48   ms   Lost/Sent = Pct  Rate     To 224.0.255.255
     v      /     hop 48   ms   ---------------------     --------------------
172.16.143.25   lwei-cisco-isdn.cisco.com
     |     ^      ttl   1
     v     |      hop 31   ms    0/12 = 0%      1 pps    0/1 = --%  0 pps
172.16.121.84
172.16.121.45   eng-frmt12-pri.cisco.com
     |     ^      ttl   2
     v     |      hop -17  ms    -735/12 = --%      1 pps    0/1 = --%  0 pps
172.16.121.4
172.16.5.27     eng-cc-4.cisco.com
     |     ^      ttl   3
     v     |      hop -21  ms    -678/23 = --%      2 pps    0/1 = --%  0 pps
172.16.5.21
172.16.62.130   eng-ios-2.cisco.com
     |     ^      ttl   4
     v     |      hop 5    ms    605/639 = 95%      63 pps    1/1 = --%  0 pps
172.16.62.144
172.16.58.65    eng-ios-f-5.cisco.com
     |      \__   ttl   5
     v         \  hop 0    ms        4         0 pps           0    0 pps
172.16.58.88    172.16.62.144
```

```
  Receiver      Query Source
```

Mrinfo

This command displays multicast neighbor router information, router capabilities and code version, multicast interface information, TTL thresholds, metrics, protocol, and status. This is useful when you must verify multicast neighbors, confirm the bidirectional neighbor adjacency exists, and verify tunnels are up in both directions.

<#root>

```
Router#
```

mrinfo

```
  192.168.7.37 (b.cisco.com) [version cisco 11.1] [flags: PMSA]:
  192.168.7.37 -> 192.168.7.34 (s.cisco.com) [1/0/pim]
  192.168.7.37 -> 192.168.7.47 (d.cisco.com) [1/0/pim]
  192.168.7.37 -> 192.168.7.44 (d2.cisco.com) [1/0/pim]
  192.168.9.26  -> 192.168.9.29 (su.bbnplanet.net) [1/32/pim]
```

The flags in the output indicates:

- P = Prune-capable
- M = Mtrace-capable
- S = SNMP-capable
- A = Auto-RP-capable

Mtrace

This command displays 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, run the **mtrace** command to find where multicast traffic flow stops, to verify the path of multicast traffic, and to identify sub-optimal paths.

<#root>

```
Router#

mtrace 192.168.215.41 192.168.215.67 239.254.254.254


Type escape sequence to abort.
Mtrace from 192.168.215.41 to 192.168.215.67 via group 239.254.254.254
From source (?) to destination (?)
Querying full reverse path...
0  192.168.215.67
-1  192.168.215.67 PIM  thresh^ 0  0 ms
-2  192.168.215.74 PIM  thresh^ 0  2 ms
-3  192.168.215.57 PIM  thresh^ 0  894 ms
-4  192.168.215.41 PIM  thresh^ 0  893 ms
-5  192.168.215.12 PIM  thresh^ 0  894 ms
-6  192.168.215.98 PIM  thresh^ 0  893 ms
```

Ping

When you troubleshoot, the **ping** command is a simple way to generate multicast traffic in the lab to test the multicast tree because reachable group members that process multicast ICMP Echo Requests can respond.

```
<#root>

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 you troubleshoot a multicast problem. Refer to the IP Multicast Configuration Guide, Cisco IOS XE 17.x for additional information.



Tip: If your **show** command responses are sluggish, the router can be performing IP domain lookups for addresses in the command output. You can disable the IP domain lookup by running the **no ip domain lookup** command in global configuration mode. 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 the Internet Group Management Protocol (IGMP). You can use this command to verify that a receiver or a locally joined router interface has joined the target group on the router interface. The **Last Reporter** column shows only one IGMP host, which indicates it has sent either an unsolicited IGMP Join or IGMP Report in response to an IGMP Query from the PIM router for that particular group. You can see one Last Reporter for each group and interface entry.

<#root>

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

Run this command to display multicast-related information on an interface, and to verify IGMP is enabled, the correct version runs, 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}**.

<#root>

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

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

<#root>

```
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 IP address.
- **Interface**: An interface where a PIM neighbor was discovered.
- **Uptime**: The total uptime of a neighbor.
- **Expires**: The time before a neighbor is timed out and until the next PIM hello is received.
- **Ver**: The version of PIM on neighbor interface.
- **DR Prio**: The possible values are 0 to 4294967294 or N.
    - This is a new column that 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 run 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 runs an older Cisco IOS® version which does not support the DR priority feature, the "DR Prior" column shows as N. If any router on the multiaccess network does not advertise the DR Priority option, the DR election uses the highest primary IP address among all PIM routers on that network.

- **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

Run this command to display information on interfaces configured for PIM. In addition, you can run this command to verify 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, and FDDI) elect a DR based on the highest advertised DR priority, with the highest primary IP address used as a tie-breaker. If a router does not advertise DR priority, the election uses the highest primary IP address. Point-to-Point links do not display DR information.

<#root>

```
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

Run this command to display the summarized contents of the IP multicast routing table. You can also use this to verify the active multicast group(s) and which multicast senders are active when you look at the timers and flags.

<#root>

```
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
  (192.168.33.32, 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

Run this command to display the full contents of the IP multicast routing table. When you troubleshoot, run 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.

<#root>

```
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


(192.168.33.32 , 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

Run this command to display the active traffic sources and groups over the threshold. When you troubleshoot, use it to verify active source-group pairs, the traffic rate for each (S,G) entry that exceeds the configured threshold, and whether the target group multicast traffic is received. If the traffic is not received, look for active traffic that starts from the source towards the receiver.

<#root>

```
R1#

show ip mroute active


Active IP Multicast Sources - sending >= 4 kbps

Group: 239.255.0.1, (?)
   Source: 192.168.33.32 (?)
   Rate: 10 pps/115 kbps(1sec), 235 kbps(last 23 secs), 87 kbps(life avg)
```

show ip rpf

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

<#root>

R1#

show ip rpf 192.168.33.32

RPF information for ? (192.168.33.32)

RPF interface: Ethernet0

  RPF neighbor: ? (192.168.10.2)
  RPF route/mask: 192.168.33.0/16
  RPF type: unicast (eigrp 1)
  RPF recursion count: 0
  Doing distance-preferred lookups across tables

---

show ip mcache

On Cisco IOS® releases that support the multicast fast-switching cache, this command displays cache entries and helps isolate fast-switching problems. On newer Cisco IOS XE® platforms, run the **show ip mfib** command to display multicast forwarding information.

<#root>

R1#

show ip mcache

IP Multicast Fast-Switching Cache
(192.168.33.32/32, 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

Run 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 run this command to verify traffic is forwarded. If it is not, run the **show ip mroute** command to look for "Null Outgoing interface list" and RPF failures.

<#root>

```
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: 192.168.33.32/32, 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

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

<#root>

```
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    192.168.33.0/16 [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

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

<#root>

```
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 must look in a functioning network. When you troubleshoot, you can distinguish between correct **debug** output and what 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

Run 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 does not learn receiver group membership on the connected interface. In PIM dense mode, initial flooding can send traffic onto the segment until pruning removes it. In PIM sparse mode, the router does not add the receiver-facing interface to the outgoing interface list without receiver membership or another local forwarding requirement.

<#root>

```
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 previous output shows 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 run 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 previous case, 60 seconds). You can also use the command to determine the version of IGMP used by the clients.

---

debug ip mpacket

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

<#root>

```
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=192.168.33.32 (Ethernet0) d=239.255.0.1 (Ethernet1) len 906, mforward
```

This command decodes the multicast packet and shows whether the packet is forwarded (mforward) or dropped. This 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 services 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 IP multicast routing table, or if it is not, why. 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.

<#root>

```
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 (192.168.33.32/32, 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

Run 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.

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

<#root>

```
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
```

- This is what each line of output denotes: R1 and R2 establish PIM neighbors when Hello messages are exchanged. 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 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. Run 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:

<#root>

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 (192.168.7.2/32, 239.255.0.1), WC-bit, RPT-bit, S-bit
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:

```
<#root>
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-bit set

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 previous, 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 an IGMP Join to R2 (RP) to join the shared tree.

When the PIM Join/Prune 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 (OIL) in the forward state. It then sends an RP-reachability message back on the interface 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 to group 239.255.0.1.

First, look at the output on R3:

<#root>

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 192.168.33.32, group 239.255.0.1

8

PIM: Send Register to 192.168.7.2 for 192.168.33.32, group 239.255.0.1

9

PIM: Received Join/Prune on Serial4/0 from 192.168.7.2

10

PIM: Join-list: (192.168.33.32/32, 239.255.0.1), S-bit set

11

PIM: Add Serial4/0/192.168.7.2 to (192.168.33.32/32, 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 (192.168.33.32/32, 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 (192.168.33.32/32, 239.255.0.1)
```

This is the output of R2, the RP:

```
<#root>

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 192.168.33.32, group 239.255.0.1

19

 PIM: Forward decapsulated data packet for 239.255.0.1 on Ethernet0

20

 PIM: Forward decapsulated data packet for 239.255.0.1 on Serial0

21

 PIM: Send Join on Serial0 to 192.168.7.1 for (192.168.33.32/32, 239.255.0.1), S-bit

22

 PIM: Send Join on Serial0 to 192.168.7.1 for (192.168.33.32/32, 239.255.0.1), S-bit

23

 PIM:

Send Register-Stop to 192.168.7.1 for 192.168.33.32, group 239.255.0.1
```

```
24
 PIM: Received Join/Prune on Serial0 from 192.168.7.1, to us
25
 PIM: Prune-list: (192.168.33.32/32, 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 (192.168.33.32/32, 239.255.0.1)
30
 PIM: Join-list: (192.168.33.32/32, 239.255.0.1), S-bit set
31
 PIM: Add Ethernet0/192.168.10.1 to (192.168.33.32/32, 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: 192.168.33.32/32
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 (192.168.33.32/32, 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 shows that R3, which is directly connected via Ethernet0/0 to the source, associates RP 192.168.7.2 with the (*, 239.255.0.1) entry, and lines 2 through 4 show that R3 sends a Join toward the RP. Lines 7 and 8 show that R3 receives source traffic and sends PIM Register messages 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 show that R3, the source DR, encapsulates packets from source 192.168.33.32 in PIM Register messages and sends them to the RP for 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 the route table.
- Lines 21 and 29 show that R2 sends a Join message towards R3 and installs an (S,G) (192.168.33.32, 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) (192.168.33.32,239.255.0.1) entry in mroute table, and sets 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 Routing Support Page
- Cisco Technical Support & Downloads