A Troubleshooting Methodology for IPv4 Multicast Routing

Bill Nickless
Caren Litvanyi
6-Feb-03

General Approach

Internet multicast forwarding follows a source-rooted tree. The tree is built by receivers "joining" or "grafting" on to the tree, following the multicast reachability topology.

Think of a farmer's field that needs water, near a pond with water in it. A farmer will dig a trench starting at the field and working towards the pond. When the trench gets to the pond, water flows back down the trench to the field.

The field is the set of multicast receivers, and the pond is the multicast source. Internet multicast routing protocols "dig a trench"—that is, build forwarding state from the receiver back towards the source. Packets flow from the source to the receivers only when the forwarding state is properly created.

A common problem with multicast forwarding is that a receiver is not getting traffic from a source on a particular group. Here we present a methodology to help network operators solve this problem in a systematic, step-by-step way.

Gathering Information

It is impossible to debug IP multicast problems without an active source and an active receiver. The spreading of the knowledge of active sources, plus the creation of forwarding state, are all operations that happen only in the presence of an active source.

In a typical autonomous system with a single PIM-SM Rendezvous Point and MSDP/M-BGP peerings, gather the following information before starting to debug a multicast problem:

- Receiver's PIM Rendezvous Point IP address
  (not just which router it's on!)
- Active source IP address
- Active receiver IP address
- IP Group address into which the active source is transmitting and the active receiver is requesting traffic

Designated Router on Receiver Subnet

RFC 1112 says that multicast gateways may exist on a subnetwork. It doesn't say anything about how those gateways can be located.

This author traveled to Los Alamos, New Mexico to help debug a multicast problem that had everyone stumped. Everyone was *assuming* the only known router on the subnet was also acting as the multicast gateway. Unfortunately, this wasn't the case. A nominally Layer 2 switch on the subnet was accidentally configured with PIM active, and won the PIM Designated Router election. Of course, this Layer 2 switch had no upstream to anywhere.

Identify and log into the router you think serves the subnet for Internet multicast. Find the interface that serves that subnet:

```
squash# show ip rpf 140.221.34.1
RPF information for ws-video.mcs.anl.gov (140.221.34.1)
  RPF interface: GigabitEthernet5/7
  RPF neighbor: ? (0.0.0.0) - directly connected
  RPF route/mask: 140.221.34.0/28
  RPF type: unicast (connected)
  RPF recursion count: 0
  Doing distance-preferred lookups across tables
squash#
```

Ask the router if it has any PIM neighbors on that subnet, and if so, which is the Designated Router (DR):

```
squash# show ip pim neighbor GigabitEthernet5/7
PIM Neighbor Table
Neighbor Address  Interface                Uptime    Expires   Ver  Mode
squash#
```

If there are any neighbors, and if any of them are the Designated Router (DR) for that subnet, then you should log into that other router and start working from there.

Receiver Join (IGMP Host Membership Report)

Internet multicast routing is receiver-driven. If there isn't interest for group traffic from a receiver, then the routers don't create any forwarding state to actually forward traffic. So, check whether the routers actually know there's an active receiver.

In the previous section you identified the Designated Router (DR) for the subnet of the active receiver. Verify that router believes there are interested receivers for the group on the interface that serves the active receiver:

```
squash# show ip igmp groups GigabitEthernet5/7
IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter
224.255.222.239  GigabitEthernet5/7       22:52:12  00:02:15  140.221.34.1
233.2.171.1      GigabitEthernet5/7       20:12:07  00:02:14  140.221.34.1
233.2.171.34     GigabitEthernet5/7       00:54:13  00:02:16  140.221.34.1
squash#
```

If the group isn't in the active list, then you have to stop and solve that problem. It might be a problem with the host, the host operating system, or the application. Broken IGMP snooping might also keep the IGMP Host Membership reports from reaching the Designated Router. Do not move any further until you've seen the group interest on the subnet of the active receiver at the Designated Router (DR).

Knowledge of Active Source

In a perfect world you would not have to worry about knowledge of active sources; all of your customer applications would use SSM and IGMPv3. Forwarding state would be created directly by the customer application's IGMPv3 source-specific Host Membership Report. If you find yourself in this desirable situation, you can skip this section entirely, going directly to 'Reachability of Active Source' below.

Most of us deal with receivers that simply send IGMPv2 Group Membership Reports. In that situation, you'll have to see whether the receiver's PIM-SM Designated Router (DR) has knowledge of the active source.

```
squash# show ip mroute 233.2.171.1 141.142.64.104
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, 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 - Advertised via MSDP, U - URD, I - Received Source Specific Host
           Report
Outgoing interface flags: H - Hardware switched
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(141.142.64.104, 233.2.171.1), 1w0d/00:02:59, flags: CJT
  Incoming interface: Vlan669, RPF nbr 130.202.222.74
  Outgoing interface list:
    GigabitEthernet5/7, Forward/Sparse, 20:19:14/00:02:08
    Vlan1, Forward/Sparse, 1w0d/00:01:56

squash#
```

If that state exists, the receiver's Designated Router (DR) knows about the active source. Skip ahead to 'Reachability of Active Source' below.

Here's an example of a lack of active source knowledge in the DR. Note that the source address is bogus, and different from above:

```
squash# show ip mroute 233.2.171.1 10.11.12.13
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, 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 - Advertised via MSDP, U - URD, I - Received Source Specific Host
           Report
Outgoing interface flags: H - Hardware switched
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 233.2.171.1), 7w0d/00:02:59, RP 192.5.170.2, flags: SJCF
  Incoming interface: Vlan29, RPF nbr 140.221.20.97
  Outgoing interface list:
    GigabitEthernet5/7, Forward/Sparse, 20:22:27/00:02:52
    Vlan1, Forward/Sparse, 7w0d/00:02:45

squash#
```

Knowledge of active sources is spread through a PIM domain by per-group Rendezvous Point (RP)-rooted shared distribution trees. Best current practice is to set the Source Path Tree (SPT) threshold to zero at the Rendezvous Point, so that (S,G) state is created on the first packet sent through the Rendezvous Point.

First, verify that the receiver's Designated Router (DR) has the right Rendezvous Point address:

```
squash# show ip pim rp mapping 233.2.171.1
PIM Group-to-RP Mappings

Group(s) 224.0.0.0/4
  RP 192.5.170.2 (kiwi-loop.anchor.anl.gov), v2v1
    Info source: 140.221.20.97 (kiwi.anchor.anl.gov), via Auto-RP, via bootstrap
         Uptime: 7w0d, expires: 00:02:47
Group(s): 224.0.0.0/4, Static
    RP: 192.5.170.2 (kiwi-loop.anchor.anl.gov)
```

Next, ask the DR what the RPF path is towards the Rendezvous Point. This information is already available to you in the (*,G) result from 'show ip mroute', but the following will tell you why it's the way it is:

```
squash# show ip rpf 192.5.170.2
RPF information for kiwi-loop.anchor.anl.gov (192.5.170.2)
  RPF interface: Vlan29
  RPF neighbor: kiwi.anchor.anl.gov (140.221.20.97)
  RPF route/mask: 192.5.170.2/32
  RPF type: unicast (ospf 683)
  RPF recursion count: 0
  Doing distance-preferred lookups across tables
squash#
```

If there is no RPF interface defined, the "upstream" interface may not have PIM configured and operational. Here's an example of what a good PIM adjacency looks like:

```
squash# show ip pim neighbor Vlan669
PIM Neighbor Table
Neighbor Address  Interface                Uptime    Expires   Ver  Mode
130.202.222.74    Vlan669                  7w0d      00:01:35  v2    (DR)
squash#
```

Repeat this process for any intermediate routers between the receiver's DR and the receiver's RP. That is, log into the intermediate routers and verify that each one has the proper RP address, the proper RPF towards the RP, and a PIM adjacency on the interface towards the RP.

Now, log in to the Rendezvous Point router. Again, verify that the RP router knows that it is the rendezvous point (by using the `show ip pim rp mapping` command.

Next, on the RP, look to see if there is knowledge of the active source. If the RP doesn't have knowledge of the active source, then it's impossible for the receiver's DR to learn of the active source. Here's what it looks like when the RP does not know about an active source, but does know that there are interested receivers for that group:

```
Kiwi# show ip mroute 233.2.171.1 10.11.12.13
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir 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
Outgoing interface flags: H - Hardware switched
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 233.2.171.1), 7w0d/stopped, RP 192.5.170.2, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet5/0, Forward/Sparse, 3w0d/00:03:28
```

Here's what it looks like when the RP doesn't even know that there are interested receivers for a group. (This can happen when the DR for a host hasn't properly joined the shared tree.)

```
Kiwi# show ip mroute 233.2.2.2
Group 233.2.2.2 not found
Kiwi#
```

There are two ways an RP can know about an active source. The traditional way is through the PIM-SM Register process. This is used when the source is within the PIM domain (in other words, the DR uses this RP). The second way is via MSDP, when the source is in a different domain (in other words, the receiver's DR and the source's DR use different RPs.) Let's take those two cases in order.

Source and Receiver DRs use the same RP

Here's an example of what it looks like at the RP when both the source and receiver's DR uses the same RP:

```
Kiwi# show ip mroute 233.2.171.1 140.221.34.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir 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
Outgoing interface flags: H - Hardware switched
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(140.221.34.1, 233.2.171.1), 5w6d/00:03:23, flags: TA
  Incoming interface: GigabitEthernet5/0, RPF nbr 140.221.20.124
  Outgoing interface list:
    ATM3/0.100103, Forward/Sparse, 1w5d/00:02:41 (ttl-threshold 32)

Kiwi#
```

Let's take a look at the flags. The T flag shows that the SPT bit is set; in other words, the RP wants all the DRs to join the source-specific tree rather than getting traffic from the

RP-rooted shared tree. The A flag says that this source is a candidate for MSDP advertisement—if this flag doesn't show up, then no MSDP peer will learn that this source is active.

If the A flag doesn't exist when it should, it may be a DR problem at the active source. Let's say that the DR knows about an active source, but the RP does not. We've already checked that the DR knows the right address for the RP. Perhaps the DR isn't properly sending PIM Register messages to the RP, or those RP messages aren't making it through. Here's what it should look like on the DR when Register messages are being sent properly:

```
squash# show ip mroute 233.2.171.1 140.221.34.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, 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 - Advertised via MSDP, U - URD, I - Received Source Specific Host
           Report
Outgoing interface flags: H - Hardware switched
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(140.221.34.1, 233.2.171.1), 5w6d/00:03:29, flags: CFT
  Incoming interface: GigabitEthernet5/7, RPF nbr 0.0.0.0
  Outgoing interface list:
    Vlan669, Forward/Sparse, 5w1d/00:03:09
    Vlan29, Forward/Sparse, 5w6d/00:02:33
    Vlan1, Forward/Sparse, 5w6d/00:02:53

squash#
```

The key here is the F flag, which indicates the RP is properly sending Register packets the RP. If this F flag doesn't show up, that explains why the RP doesn't know about the active source. Cisco IOS 12.0(10) fixes a major problem with the DR code. If you're running Cisco IOS on your DR, and it's earlier than 12.0(10), then you must upgrade to a more recent version. Otherwise, there's a random failure when the DR will silently stop sending Register packets to the RP.

Source and Receiver DRs use different RPs

The other way for an RP to know about an active source is through MSDP. Here's an example of how that sort of state looks on an RP:

```
Kiwi# show ip mroute 233.2.171.1 141.142.64.104
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir 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
Outgoing interface flags: H - Hardware switched
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(141.142.64.104, 233.2.171.1), 1w0d/00:02:19, flags: PMT
  Incoming interface: GigabitEthernet1/0.600, RPF nbr 130.202.222.82, Mbgp
  Outgoing interface list: Null

Kiwi#
```

The M flag is the key indicator that MSDP created the entry.

Troubleshooting MSDP

If you think you should be getting Source Active (SA) messages through MSDP to your RP, but the RP doesn't have knowledge [(S,G) state] of that active source, then there's a problem with MSDP.

MSDP uses so-called Peer-RPF rules to determine from where to accept MSDP Source Active notifications. Peer-RPF operates on the IP address of the source's Rendezvous Point. So to debug MSDP problems, you need the IP address of the source's Rendezvous Point. One way to determine this is to contact the operators of the network of the sender and ask them what their RP address should be.

Once you know the source's Rendezvous Point IP address, you can determine which active MSDP session (if any) to expect Source Active messages for that source. You should use your vendor's documentation and/or the MSDP specification for the exact Peer-RPF selection rules. But in general, MSDP will select the session associated with the multicast reachability towards the sender's RP.

A common mistake is for a BGP-speaking Autonomous System to not advertise their Rendezvous Point IP address as being multicast reachable. (In other words, not advertising the RP's IP address in the IPv4 multicast address family.) This can cause MSDP SAs from that RP to be rejected by all of their peers, or some downstream peers.

Once you have determined the Peer-RPF session associated with sender's RP address, you can notify the MSDP peer operator of the problem and work with them to start getting the required MSDP SA to your RP.

When you contact the appropriate MSDP Peer-RPF peer, you should give them the following information about the missing MSDP SA:

- The active source's IP address
- The active group's IP address
- The active source's Rendezvous Point IP address

Ask them whether they have the desired MSDP SA in their caches. If they don't have it, that's a problem they have to work on and fix. If they do have it, you should work with them to get the MSDP SA from their MSDP speaker to your MSDP speaker.

What If MSDP Works But PIM Doesn't?

MSDP encapsulates data in the Source Active messages. This encapsulated data is supposed to be sent down the RP-rooted shared tree for that group, creating source-specific state in the DRs with interested receivers. The RP is also supposed to join the group itself and send traffic down the RP-rooted shared tree. If, for some reason, MSDP is operating but PIM isn't, you can get a small trickle of traffic through the MSDP control-plane channel. This can lead users of SDR to believe IP Multicast is working correctly, when in fact there are serious PIM failures.

If you think this is going on, check the PIM forwarding state from the RP towards the source, following the procedures in the 'Reachability of Active Source' and 'Forwarding State' sections below. But remember, you're solving the problem of why the RP isn't getting traffic, not the question of why the interested source isn't getting traffic. Once the RP is getting the active source traffic properly, you have to go back to the receiver's DR to verify that the receiver's DR knows about the active source.

Knowledge of Active Source to the Receiver's DR

Keep working on all of this until the DR associated with the RP has knowledge of the active source. There may be multiple problems; you may fix MSDP, only to discover that there's a problem with the RP-rooted distribution tree that must also be fixed.

Only when the receiver's DR has knowledge of the active source can you move ahead to the next section.

Reachability of Active Source

You may have noticed we used the 'show ip rpf' command above, rather than the more familiar 'show ip route'. That's because 'show ip rpf' shows the multicast reachability directly—that's how PIM will decide where to send Join messages towards the source.

Within a PIM Domain, the multicast reachability topology often follows the unicast reachability topology. This is true even if some of the inter-router links don't have PIM forwarding enabled. If 'show ip rpf' points towards a router interface that doesn't have PIM enabled, or if there's no PIM adjacency, then forwarding state can't be created.

On the Designated Router (DR) with known group interest, verify the RPF next hop towards the active source:

```
squash# show ip rpf 141.142.64.104
RPF information for ag-nl-video.ncsa.uiuc.edu (141.142.64.104)
  RPF interface: Vlan669
  RPF neighbor: guava-stardust.anchor.anl.gov (130.202.222.74)
  RPF route/mask: 0.0.0.0/0
  RPF type: unicast (ospf 683)
  RPF recursion count: 0
  Doing distance-preferred lookups across tables
```

Stop and think. Is this the router that should be upstream towards the source?

Next, verify that there's a matching PIM Sparse Mode adjacency for that RPF next hop:

```
squash# show ip pim neighbor Vlan669
PIM Neighbor Table
Neighbor Address  Interface               Uptime    Expires   Ver  Mode
130.202.222.74    Vlan669                 7w0d      00:01:35  v2    (DR)
squash#
```

Now repeat this process from the next router—the one indicated in the 'show ip rpf' results at the beginning of this section. Eventually you'll come to the subnet with the active source, or to the edge of your area of responsibility.

If you come to two routers that point to each other in the results of `show ip rpf`, then you have a multicast reachability loop. The "trench" can't be built to the source, because the routers don't actually have a valid path towards the source. Use the information provided by the `show ip rpf` command to identify why the RPF is being chosen at each router. Once you've identified the root cause, adjust the underlying reachability (routing) to remove the loop.

If you come up against the edge of your area of responsibility, make a note to yourself where your routers will be asking for traffic from that source.

Forwarding State

Forwarding state is created in routers to actually move multicast packets from sources to receivers. It is created upon request by receivers. SSM distribution trees are created directly from the source to the interested receiver; the network can do this directly because the receiver tells the internetwork about the existence of the source. The same kind of distribution tree is created in the ASM model, but that requires all the machinery for spreading the knowledge of active sources.

The basic assumption of this section is that the Designated Router for the subnet of the receiver knows both the interest of the receiver and the existence of the active source, but for some reason the source's packets aren't being delivered to the receiver.

The first thing to check is whether the receiver's Designated Router is receiving packets from the active source. In other words, verify that the problem is "upstream" of the receiver's DR.

```
squash# show ip mroute 233.2.171.1 141.142.64.104 count
IP Multicast Statistics
229 routes using 104850 bytes of memory
42 groups, 4.45 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: 233.2.171.1, Source count: 100, Group pkt count: 987285640
  Source: 141.142.64.104/32, Forwarding: 19113351/27/94/20, Other: 19123422/10070/1
squash#
```

In this example, the Designated Router (DR) is indeed receiving 27 packets per second from the source. Next, double-check that the DR has the receiver's interface in the output interface list for that source and group:

```
squash#show ip mroute 233.2.171.1 141.142.64.104
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, 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 - Advertised via MSDP, U - URD, I - Received Source Specific Host
           Report
Outgoing interface flags: H - Hardware switched
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(141.142.64.104, 233.2.171.1), 1w1d/00:02:59, flags: CJT
  Incoming interface: Vlan669, RPF nbr 130.202.222.74
  Outgoing interface list:
    GigabitEthernet5/7, Forward/Sparse, 1d07h/00:02:12
    Vlan1, Forward/Sparse, 1w1d/00:02:01

squash#
```

Assuming the receiver is attached to GigabitEthernet5/7, it looks like the Designated Router is forwarding correctly.

If we get to this point and the receiver isn't getting the packets, there's a problem between the Designated Router and the receiver. IGMP-snooping Ethernet switched networks can fail in this way.

Unfortunately, debugging IGMP-snooping is outside the scope of this paper. IGMP-snooping behaviors are not defined well enough to lend themselves to a step-by-step debugging strategy. In the author's opinion, Ethernets doing IP multicast should almost never be bridged or switched; instead, each Ethernet segment should be individually routed.

Here's an example of a source for which the Designated Router is not receiving packets:

```
squash# show ip mroute 233.2.171.1 204.121.50.22 count
IP Multicast Statistics
226 routes using 103842 bytes of memory
42 groups, 4.38 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: 233.2.171.1, Source count: 100, Group pkt count: 987910557
  Source: 204.121.50.22/32, Forwarding: 0/0/0/0, Other: 6/0/6
squash#
```

The next step is to see where the packets should have come from:

```
squash# show ip rpf 204.121.50.22
RPF information for agaudio2.acl.lanl.gov (204.121.50.22)
  RPF interface: Vlan669
  RPF neighbor: guava-stardust.anchor.anl.gov (130.202.222.74)
  RPF route/mask: 0.0.0.0/0
  RPF type: unicast (ospf 683)
  RPF recursion count: 0
  Doing distance-preferred lookups across tables
squash#
```

Stop and think. Is that the right upstream for this source? Is the RPF resolving the multicast reachability topology properly? It should be, if we've followed the process in the previous section "Reachability of Active Source".

Again, verify that there's an active PIM adjacency for this upstream:

```
squash# show ip pim neighbor Vlan669
PIM Neighbor Table
Neighbor Address  Interface                Uptime    Expires   Ver  Mode
130.202.222.74    Vlan669                  7w0d      00:01:38  v2    (DR)
squash#
```

And double-check that the Designated Router has sent a PIM (S,G) Join towards the correct upstream neighbor:

```
squash# show ip mroute 233.2.171.1 204.121.50.22
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, 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 - Advertised via MSDP, U - URD, I - Received Source Specific Host
           Report
Outgoing interface flags: H - Hardware switched
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(204.121.50.22, 233.2.171.1), 00:00:41/00:02:18, flags: CJ
  Incoming interface: Vlan669, RPF nbr 130.202.222.74
  Outgoing interface list:
    Vlan1, Forward/Sparse, 00:00:41/00:02:18
    GigabitEthernet5/7, Forward/Sparse, 00:00:41/00:02:20
```

Now log in to the upstream router and check that the forwarding state exists, and that the outgoing interface list includes the interface towards the previous router checked:

```
guava# show ip mroute 233.2.171.1 204.121.50.22
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, 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 - Advertised via MSDP, U - URD, I - Received Source Specific Host
           Report
Outgoing interface flags: H - Hardware switched
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(204.121.50.22, 233.2.171.1), 00:00:18/00:03:11, flags:
  Incoming interface: Vlan800, RPF nbr 192.5.170.78
  Outgoing interface list:
    Vlan669, Forward/Sparse, 00:00:18/00:03:11

guava#
```

Next, check to see whether this router is receiving packets:

```
guava# show ip mroute 233.2.171.1 204.121.50.22 count
IP Multicast Statistics
508 routes using 180048 bytes of memory
41 groups, 11.39 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: 233.2.171.1, Source count: 114, Group pkt count: 931903200
  Source: 204.121.50.22/32, Forwarding: 0/0/0/0, Other: 0/0/0
guava#
```

In this example, the intermediate router still isn't receiving packets from the source. In such cases, repeat the process of checking whether the RPF is being calculated correctly, that the upstream interface is correct in the state, what peer should be sending the traffic, and whether there's a PIM adjacency.

If you find that a router isn't receiving packets from a source, but the upstream router is receiving those packets, you need to track down how those packets are being lost in the intermediate link. Check any access control lists associated with those interfaces, on both routers. Also see if there's a broken IGMP-snooping Ethernet bridge between them.

You may follow the trail towards a source to the edge of your area of responsibility. At that point you need to contact the operator of the neighboring router. You'll need to ask them to continue the process towards the source. Information you'll need to give them will include:

- The active source's IP address (S)
- The multicast group IP address (G)
- The shared link (circuit ID, ATM PVC, Frame Relay DLCI) towards which your router has sent the PIM (S,G) Join
- The fact that you're not receiving packets from them for that source and group on that shared link

Once you've passed that information to the upstream, they will have to follow the same process through their network towards the source.

Good luck!

In summary, here is a recap of the methodology:

- Gather information (IP addresses of the active source, group, and receiver)

- Identify the Designated Router (DR) associated with the active receiver, and verify that the DR knows of the interest of the active receiver in the group traffic. Check that the receiver's DR is not receiving traffic before trying to fix anything!

- Get the receiver's DR to know of the existence of the active (source,group), which might mean fixing the multicast reachability topology, PIM state, and/or MSDP.

- Trace the forwarding state through the network from the receiver's DR towards the source. At each router, verify the multicast reachability, the PIM state, and whether traffic is being received properly.

By following this methodology, you should be able to find and fix the majority of situations where IP multicast traffic is not being passed from a sender to a receiver. And if you can't solve the end-to-end problem yourself, you know what information to give to your peer networks so they can continue the troubleshooting process towards the source.