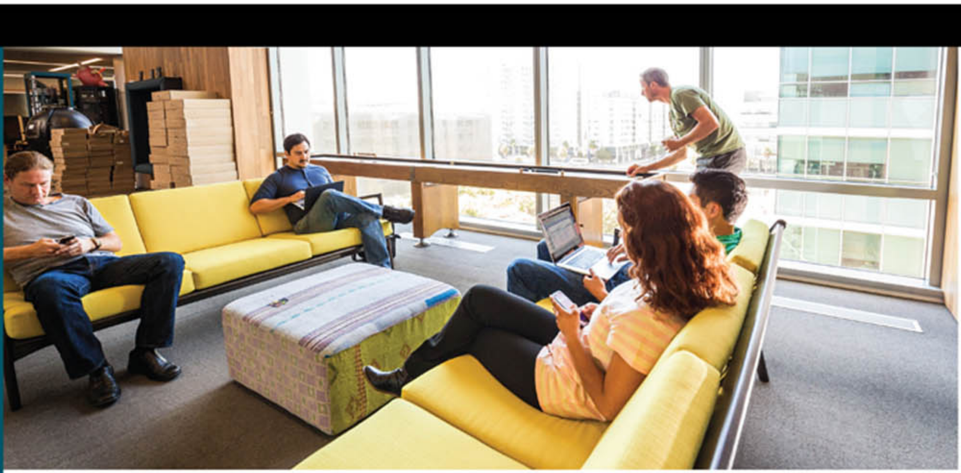

IP Multicast

Volume I: Cisco IP Multicast Networking

Josh Loveless, CCIE No. 16638
Ray Blair, CCIE No. 7050
Arvind Durai, CCIE No. 7016

ciscopress.com

livelessons
video instruction from technology experts
livelessons
video instruction from technology experts
livelessons

```
PIM(0): Add Ethernet0/0/192.168.52.5 to (*, 239.1.1.1), Forward state, by PIM
*G Join
PIM(0): Add Ethernet0/0/192.168.52.5 to (192.168.8.8, 239.1.1.1), Forward
state, by PIM *G Join
```

5. **FHR registers the active source:** The RP already received a register packet from the FHR because it has the (*,G) state with receiver information. The RP sends a (S,G) join message to 224.0.0.13 towards the FHR and then sends a register stop message directly to the FHR. The output from the RP is shown using the following **debug ip pim** message:

```
*May 11 21:36:48.311: PIM(0): Received v2 Register on Ethernet0/2 from
192.168.43.3
*May 11 21:36:48.311:      for 192.168.8.8, group 239.1.1.1
*May 11 21:36:48.311: PIM(0): Adding register decap tunnel (Tunnel1) as
accepting interface of (192.168.8.8, 239.1.1.1).
*May 11 21:36:48.311: PIM(0): Insert (192.168.8.8,239.1.1.1) join in nbr
192.168.42.4's queue
*May 11 21:36:48.311: PIM(0): Building Join/Prune packet for nbr 192.168.42.4
*May 11 21:36:48.311: PIM(0):  Adding v2 (192.168.8.8/32, 239.1.1.1), S-bit
Join
*May 11 21:36:48.311: PIM(0): Send v2 join/prune to 192.168.42.4 (Ethernet0/1)
*May 11 21:36:48.310: PIM(0): Check RP 192.168.0.2 into the (*, 239.1.1.1)
entry
*May 11 21:36:48.310: PIM(0): Building Triggered (*,G) Join / (S,G,RP-bit)
Prune message for 239.1.1.1
*May 11 21:36:48.310: PIM(0): Adding register encap tunnel (Tunnel0) as for-
warding interface of (192.168.8.8, 239.1.1.1).
*May 11 21:36:48.313: PIM(0): Received v2 Join/Prune on Ethernet0/1 from
192.168.43.4, to us
*May 11 21:36:48.313: PIM(0): Join-list: (192.168.8.8/32, 239.1.1.1), S-bit
set
*May 11 21:36:48.313: PIM(0): Add Ethernet0/1/192.168.43.4 to (192.168.8.8,
239.1.1.1), Forward state, by PIM SG Join
```

At the FHR, the output of the **debug ip pim** message at this step is as follows:

```
*May 11 21:45:11.152: PIM(0): Check RP 192.168.0.2 into the (*, 239.1.1.1)
entry
*May 11 21:45:11.152: PIM(0): Building Triggered (*,G) Join / (S,G,RP-bit)
Prune message for 239.1.1.1
*May 11 21:45:11.152: PIM(0): Adding register encap tunnel (Tunnel0) as
forwarding interface of (192.168.8.8, 239.1.1.1).
*May 11 21:45:11.155: PIM(0): Received v2 Join/Prune on Ethernet0/1 from
192.168.43.4, to us
*May 11 21:45:11.155: PIM(0): Join-list: (192.168.8.8/32, 239.1.1.1), S-bit
set
*May 11 21:45:11.155: PIM(0): Add Ethernet0/1/192.168.43.4 to (192.168.8.8,
239.1.1.1), Forward state, by PIM SG Join
```

```
R3#
*May 11 21:45:15.133: PIM(0): Received v2 Register-Stop on Ethernet0/1 from
192.168.0.2
*May 11 21:45:15.133: PIM(0):   for source 192.168.8.8, group 239.1.1.1
*May 11 21:45:15.133: PIM(0): Removing register encap tunnel (Tunnel0) as
forwarding interface of (192.168.8.8, 239.1.1.1).
*May 11 21:45:15.133: PIM(0): Clear Registering flag to 192.168.0.2 for
(192.168.8.8/32, 239.1.1.1)
```

R3 receives the register stop and the (S,G) join from the RP. Now the FHR (R3) sends the multicast traffic towards the RP in a unicast tunnel. The flow of the multicast stream now progresses from the RP down to R7 and ultimately towards the receiver. This is the initial tree build up in PIM sparse-mode. For details on the SPT switch, refer to the earlier section, "The Rendezvous Point and Shared Tree Dynamics."

6. **The multicast flow moves from the shared tree to a source tree:** R3 received the prune message from the RP and added the interface Ethernet0/1 to the OIL, as shown from the following **debug ip pim** message, creating a source tree from the FHR to the LHR:

```
PIM(0): Received v2 Join/Prune on Ethernet0/1 from 192.168.43.4, to us
PIM(0): Join-list: (192.168.8.8/32, 239.1.1.1), S-bit set
PIM(0): Add Ethernet0/1/192.168.43.4 to (192.168.8.8, 239.1.1.1), Forward
state, by PIM SG Join
```

R5 sees the source tree formed from the FHR (R3) and uses the RPF in the unicast routing table to find a better metric. When this occurs, R5 sends a PIM prune toward the RP to remove itself from the shared tree, as shown from the **debug ip pim** output from R5 below.

```
PIM(0): Received v2 Join/Prune on Ethernet0/0 from 192.168.75.7, to us
PIM(0): Join-list: (*, 239.1.1.1), RPT-bit set, WC-bit set, S-bit set
PIM(0): Add Ethernet0/0/192.168.75.7 to (*, 239.1.1.1), Forward state, by PIM
*G Join
PIM(0): Building Triggered (*,G) Join / (S,G,RP-bit) Prune message for
239.1.1.1
PIM(0): Upstream mode for (*, 239.1.1.1) changed from 0 to 1
PIM(0): Insert (*,239.1.1.1) join in nbr 192.168.52.2's queue
PIM(0): Insert (192.168.8.8,239.1.1.1) sgr prune in nbr 192.168.52.2's queue
PIM(0): Update Ethernet0/0/192.168.75.7 to (192.168.8.8, 239.1.1.1), Forward
state, by PIM *G Join
PIM(0): Building Join/Prune packet for nbr 192.168.52.2
PIM(0):  Adding v2 (192.168.0.2/32, 239.1.1.1), WC-bit, RPT-bit, S-bit Join
PIM(0):  Adding v2 (192.168.8.8/32, 239.1.1.1), RPT-bit, S-bit Prune
PIM(0): Send v2 join/prune to 192.168.52.2 (Ethernet0/2)
```

Figure 3-19 shows the traffic flow for the new source tree.

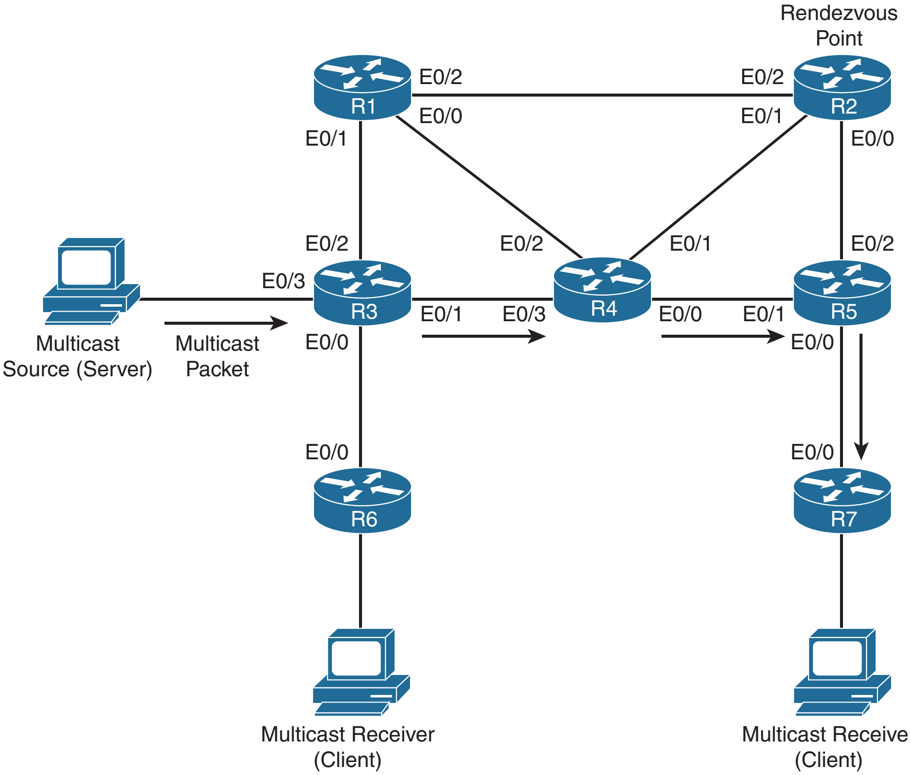

Figure 3-19 *Source Tree Forwarding*

7. **The new source tree is fully activated:** The source tree is now active, as seen using the **show ip mroute 239.1.1.1** command on R3:

```
R3#show ip mroute 239.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
       N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
       Q - Received BGP S-A Route, q - Sent BGP S-A Route,
       V - RD & Vector, v - Vector, p - PIM Joins on route
```

```
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM
Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 239.1.1.1), 00:00:22/stopped, RP 192.168.0.2, flags: SPF
  Incoming interface: Ethernet0/1, RPF nbr 192.168.43.4
  Outgoing interface list: Null

(192.168.8.8, 239.1.1.1), 00:00:22/00:03:07, flags: FT
  Incoming interface: Ethernet0/3, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0/1, Forward/Sparse, 00:00:22/00:03:07
```

The shared tree has not been removed. When another receiver connected to a different router is interested in the multicast stream, the shared tree and the same process of moving from a shared tree to a source tree will occur again, like déjà vu all over again.

Note While there is much going on under the hood in this section, the process of switching between the shared tree and the source tree is relatively quick (less than a second in a robust network). This is exactly what we want! If a sparse-mode network has trees stuck in the shared tree state, eating up RP router resources, this indicates that a group has no source(s), or that something has likely gone wrong in the PIM process. In fact, this is a common symptom that occurs in a network in which PIM communication has broken down. For information on how to find and fix such failures, please see Chapter 7, which covers PIM troubleshooting.

Building the Multicast Routing Information Base

The unicast routing information base (RIB) has been populated by unicast routing protocols. It is not necessary to build another multicast routing table with identical or additional route information that consumes additional memory. As discussed earlier with RPF checks, rather than forward packets to the source, multicast routers forward packets away from the source.

When a router receives a multicast packet, it performs the following actions.

1. Upon receipt of a multicast packet, the router checks the receiving interface to determine whether it is in the path to the source (sending IP address) by performing a lookup in the RIB. This is known as the *RPF check*.
2. If the interface is in the path to the source, the packet is forwarded.
3. If the receiving interface is not in the path to the source, the RPF fails and the packet is dropped.

Multicast takes a unique perspective when it comes to forwarding traffic in the network. Rather than looking for the destination address of where to send the information as with traditional routing, multicast routing does just the opposite. Because multicast routing is the opposite of unicast or traditional routing, you can take advantage of the RIB.

The RIB is a collection of all routing information contained on a router and is a culmination of connected routes, static routes, and routes learned from dynamic routing protocols. It also holds adjacency information for neighboring routers. The RIB is used to populate the forwarding information base (FIB) with the best path to a particular destination. The FIB table is used to forward packets to the appropriate destination.

Multicast Routing Information Base and Multicast Forwarding Information Base

In the earlier section “Two Types of Trees,” we addressed a couple of items that need some further clarification. These are the multicast routing information base (MRIB) and multicast forwarding information base (MFIB) and the way they interact with the other functions of the router.

The MRIB table is a collection of multicast entries, including source, group, group mask, and flags, and may also contain a list of associated interfaces. The entries in the MRIB are created/updated by the control plane when traffic is received from a connected source, a switchover occurs, or there is some other data-driven event. The MRIB table (for non-local multicast groups) can be viewed on older IOS platforms simply by using the **show ip mroute** command as it has been discussed. Newer versions of IOS, including IOS XE, can display the MRIB table using the **show ip mrib route** command, as demonstrated in Example 3-15.

Example 3-15 *Displaying the MRIB Table*

```
ASR1K-2#show ip mrib route
IP Multicast Routing Information Base
Entry flags: L - Domain-Local Source, E - External Source to the Domain,
    C - Directly-Connected Check, S - Signal, IA - Inherit Accept, D - Drop
    ET - Data Rate Exceeds Threshold,K - Keepalive,DDE - Data Driven Event
Interface flags: F - Forward, A - Accept, IC - Internal Copy,
    NS - Negate Signal, DP - Don't Preserve, SP - Signal Present,
    II - Internal Interest, ID - Internal Disinterest, LI - Local Interest,
    LD - Local Disinterest, MD - mCAC Denied

(*,224.64.7.7) RPF nbr: 0.0.0.0 Flags: C
  GigabitEthernet0/0/2 Flags: F NS
  GigabitEthernet0/0/1 Flags: F NS
  GigabitEthernet0/0/0 Flags: F NS

(192.168.8.10,224.64.7.7) RPF nbr: 192.168.32.3 Flags:
  GigabitEthernet0/0/1 Flags: A
  GigabitEthernet0/0/0 Flags: F NS
```

The MFIB is the multicast forwarding information derived from the MRIB and is presented as a unique table, which displays how the router intends to forward multicast messages. Information such as source, group, mask, counts, and rates of received, dropped, and forwarded packets are contained in this table. Example 3-16 shows an abbreviated output from the **show ip mfib** command.

Example 3-16 *MFIB State Table*

```
ASR1K-2#show ip mfib
Entry Flags:    C - Directly Connected, S - Signal, IA - Inherit A flag,
                ET - Data Rate Exceeds Threshold, K - Keepalive
                DDE - Data Driven Event, HW - Hardware Installed
I/O Item Flags: IC - Internal Copy, NP - Not platform switched,
                NS - Negate Signaling, SP - Signal Present,
                A - Accept, F - Forward, RA - MRIB Accept, RF - MRIB Forward,
                MA - MFIB Accept
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kbits per second
Other counts:      Total/RPF failed/Other drops
I/O Item Counts:   FS Pkt Count/PS Pkt Count
Default
 (*,224.64.7.7) Flags: C HW
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   HW Forwarding:   0/0/0/0, Other: 0/0/0
   GigabitEthernet0/0/0 Flags: F NS
     Pkts: 0/0
   GigabitEthernet0/0/1 Flags: F NS
     Pkts: 0/0
   GigabitEthernet0/0/2 Flags: F NS
     Pkts: 0/0
 (192.168.8.10,224.64.7.7) Flags: HW
   SW Forwarding: 0/0/0/0, Other: 15/2/13
   HW Forwarding:   685830/156/1369/1673, Other: 1111/1111/0
   GigabitEthernet0/0/1 Flags: A
   GigabitEthernet0/0/0 Flags: F NS
     Pkts: 0/0
```

Figure 3-20 depicts the interaction between the various components involved in the derivation of multicast forwarding information.