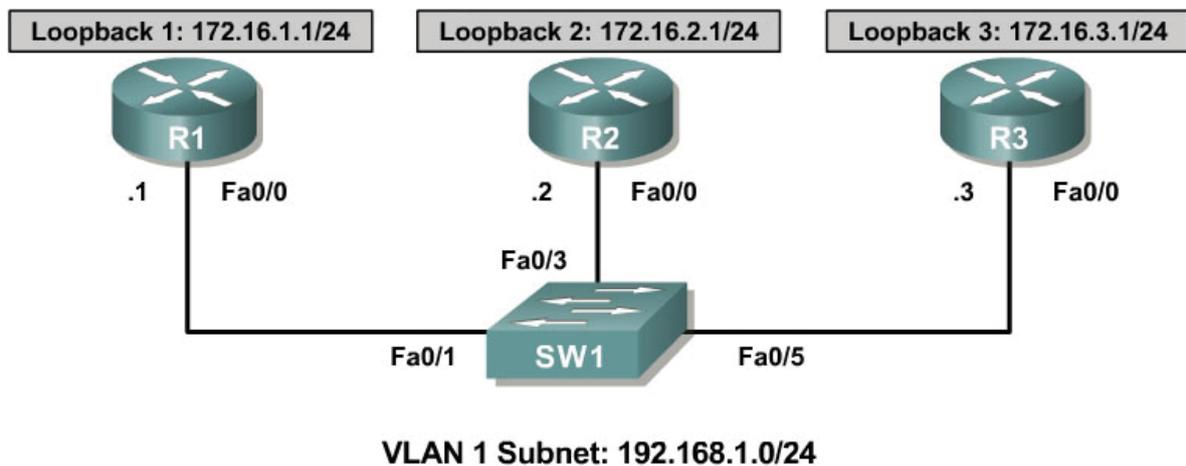


Lab 7-1 Implementing IGMP and IGMP Snooping

Learning Objectives

- Configure IGMP to join interfaces to a multicast group
- Verify the operation of IGMP at Layer 3
- Analyze IGMP packets and packets sent to multicast groups
- Enable PIM-DM
- Verify the operation of IGMP snooping on a Catalyst switch

Topology Diagram



Scenario

Configure IGMP to listen to the multicast group 229.7.7.7 on R2 and R3. Send multicast traffic from R1 to the LAN segment. Configure IGMP snooping to efficiently send multicast traffic through the switch.

Review IGMP, PIM-DM, and IGMP snooping in your course materials before completing this lab.

Overview

This set of multicast labs builds skills in working with multicast features and routing protocols so that the student can confidently configure PIM in sparse-dense mode. Sparse-dense mode is simply PIM sparse mode (PIM-SM) with a dense mode fallback control mechanism.

To grasp the principles involved in sparse-dense mode, students must understand the principles involved with IGMP and PIM routing protocols at Layer 3. Thus, considerable lab time will be spent understanding IGMP as well as each of the PIM protocols. IGMP snooping greatly improves multicast efficiency at Layer 2 and therefore receives thorough treatment.

Step 1: Configure Hosts on a LAN

Configure the basic addressing scheme in the diagram on each router. Disable IP routing using the **no ip routing** command. This command forces a router to act as a host connected to a specific VLAN via the switch. Routing between networks is not performed.

```
R1# conf t
R1(config)# no ip routing
R1(config)# interface fastethernet 0/0
R1(config-if)# ip address 192.168.1.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# interface loopback 1
R1(config-if)# ip address 172.16.1.1 255.255.255.0

R2# conf t
R2(config)# no ip routing
R2(config)# interface fastethernet 0/0
R2(config-if)# ip address 192.168.1.2 255.255.255.0
R2(config-if)# no shutdown
R2(config-if)# interface loopback 2
R2(config-if)# ip address 172.16.2.1 255.255.255.0

R3# conf t
R3(config)# no ip routing
R3(config)# interface fastethernet 0/0
R3(config-if)# ip address 192.168.1.3 255.255.255.0
R3(config-if)# no shutdown
R3(config-if)# interface loopback 3
R3(config-if)# ip address 172.16.3.1 255.255.255.0
```

With routing disabled, will the routers redirect packets between remote addresses on connected subnets?

No, with routing disabled, no packets will be forwarded between subnets.

Reload the switch with a blank configuration file and erase the vlan.dat file in flash memory. By default, all connected ports are assigned to VLAN 1.

Step 2: Subscribe Interfaces to Multicast Groups with IGMP

In IP multicast, hosts use the Internet Group Management Protocol (IGMP) to join and leave groups and to respond to membership queries.

Debug IP IGMP and all IP packets on all routers. This allows you to see any IGMP messages sent or received by any of the routers.

```

R1# debug ip igmp
R1# debug ip packet

R2# debug ip igmp
R2# debug ip packet

R3# debug ip igmp
R3# debug ip packet

```

On R1, issue the **ip igmp join-group** *group_address* command in interface configuration mode for FastEthernet0/0.

```

R1# conf t
R1(config)# interface fastethernet 0/0
R1(config-if)# ip igmp join-group 229.7.7.7

```

```

*Nov  3 20:54:57.114: IGMP(0): WAVL Insert group: 229.7.7.7 interface:
FastEthernet0/0Successful
*Nov  3 20:54:57.114: IGMP(0): Send v2 Report for 229.7.7.7 on FastEthernet0/0
*Nov  3 20:54:57.114: IP: s=192.168.1.1 (local), d=229.7.7.7
(FastEthernet0/0), len 28, sending broad/multicast

```

The **ip igmp join-group** command sends an IGMPv2 join message to the group-specific address, which is then propagated to all reachable members of the specified group. This message states that multicast traffic destined for the group should be forwarded to the local router's IP address. Because none of the other routers are listening to the 229.7.7.7 group at this point, R2 and R3 do not receive a broadcast sent to the group.

Issue the **show ip igmp interface** *interface_type interface_number* command on R1.

```

R1# show ip igmp interface fastethernet 0/0
FastEthernet0/0 is up, line protocol is up
  Internet address is 192.168.1.1/24
  IGMP is enabled on interface
  Current IGMP host version is 2
  Current IGMP router version is 2
  IGMP query interval is 60 seconds
  IGMP querier timeout is 120 seconds
  IGMP max query response time is 10 seconds
  Last member query count is 2
  Last member query response interval is 1000 ms
  Inbound IGMP access group is not set
  IGMP activity: 1 joins, 0 leaves
  Multicast routing is disabled on interface
  Multicast TTL threshold is 0
  Multicast groups joined by this system (number of users):
    229.7.7.7(1)

```

Which version of IGMP is R1's Fast Ethernet interface using?

R1's Fast Ethernet interface is using IGMP version 2.

Remove the **ip igmp join-group** command from R1's FastEthernet0/0 interface because it is not used for the remainder of this lab.

```
R1(config)# interface fastethernet 0/0
R1(config-if)# no ip igmp join-group 229.7.7.7
```

```
*Nov  3 20:59:10.582: IGMP(0): IGMP delete group 229.7.7.7 on FastEthernet0/0
*Nov  3 20:59:10.582: IGMP(0): Send Leave for 229.7.7.7 on FastEthernet0/0
*Nov  3 20:59:10.582: IP: s=192.168.1.1 (local), d=224.0.0.2
(FastEthernet0/0), len 28, sending broad/multicast
```

Note that the IGMPv2 leave messages are sent to 224.0.0.2, the link-local multicast group representing all routers on the subnet.

Use R1 as the originator of multicast traffic to the group 229.7.7.7, but do not subscribe it to the group. Use IGMP to subscribe R2 and R3 to the 229.7.7.7 group. If you turned debugging off on R2 and R3, re-enable debugging on both routers to watch the IGMP messages. Keep debugging on for IGMP and IP packets through the end of Step 4.

```
R2# debug ip igmp
R2# debug ip packet
```

```
R3# debug ip igmp
R3# debug ip packet
```

```
R2(config-if)# ip igmp join-group 229.7.7.7
*Nov  4 04:59:59.340: IGMP(0): WAVL Insert group: 229.7.7.7 interface:
FastEthernet0/0Successful
*Nov  4 04:59:59.340: IGMP(0): Send v2 Report for 229.7.7.7 on FastEthernet0/0
*Nov  4 04:59:59.340: IP: s=192.168.1.2 (local), d=229.7.7.7
(FastEthernet0/0), len 28, sending broad/multicast
```

```
R3(config-if)# ip igmp join-group 229.7.7.7
*Nov  4 05:02:11.696: IGMP(0): WAVL Insert group: 229.7.7.7 interface:
FastEthernet0/0Successful
*Nov  4 05:02:11.700: IGMP(0): Send v2 Report for 229.7.7.7 on FastEthernet0/0
*Nov  4 05:02:11.700: IP: s=192.168.1.3 (local), d=229.7.7.7
(FastEthernet0/0), len 28, sending broad/multicast
```

Display the IGMP groups to which R2 and R3 have subscribed with the **show ip igmp groups** and **show ip igmp membership** commands. Verify that the specified groups are being received.

```
R2# show ip igmp groups
IGMP Connected Group Membership
Group Address Interface      Uptime    Expires  Last Reporter  Group Accounted
229.7.7.7    FastEthernet0/0 00:07:56  stopped  192.168.1.2
```

```
R2# show ip igmp membership
Flags: A - aggregate, T - tracked
       L - Local, S - static, V - virtual, R - Reported through v3
       I - v3lite, U - Urd, M - SSM (S,G) channel
       1,2,3 - The version of IGMP the group is in
Channel/Group-Flags:
/ - Filtering entry (Exclude mode (S,G), Include mode (*,G))
Reporter:
<mac-or-ip-address> - last reporter if group is not explicitly tracked
```

```

    <n>/<m>          - <n> reporter in include mode, <m> reporter in exclude

Channel/Group      Reporter      Uptime  Exp.  Flags  Interface
*,229.7.7.7       192.168.1.2  00:08:27 stop  2LA   Fa0/0

R3# show ip igmp groups
IGMP Connected Group Membership
Group Address Interface      Uptime  Expires  Last Reporter Group Accounted
229.7.7.7   FastEthernet0/0  00:09:00 stopped  192.168.1.3

R3# show ip igmp membership
Flags: A - aggregate, T - tracked
       L - Local, S - static, V - virtual, R - Reported through v3
       I - v3lite, U - Urd, M - SSM (S,G) channel
       1,2,3 - The version of IGMP the group is in
Channel/Group-Flags:
 / - Filtering entry (Exclude mode (S,G), Include mode (*,G))
Reporter:
 <mac-or-ip-address> - last reporter if group is not explicitly tracked
 <n>/<m>          - <n> reporter in include mode, <m> reporter in exclude

Channel/Group      Reporter      Uptime  Exp.  Flags  Interface
*,229.7.7.7       192.168.1.3  00:09:27 stop  2LA   Fa0/0

```

From R1, ping the multicast group 229.7.7.7. The output should appear similar to the following with debugging turned on:

```

R1# ping 229.7.7.7

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 229.7.7.7, timeout is 2 seconds:

*Nov  4 16:10:18.413: IP: s=192.168.1.1 (local), d=229.7.7.7
(FastEthernet0/0), len 100, sending broad/multicast
*Nov  4 16:10:18.413: IP: s=172.16.1.1 (local), d=229.7.7.7 (Loopback1), len
100, sending broad/multicast
*Nov  4 16:10:18.413: IP: s=192.168.1.3 (FastEthernet0/0), d=192.168.1.1, len
100, rcvd 1
*Nov  4 16:10:18.413: IP: s=192.168.1.2 (FastEthernet0/0), d=192.168.1.1, len
100, rcvd 1
Reply to request 0 from 192.168.1.3, 1 ms
Reply to request 0 from 192.168.1.2, 1 ms

```

The output above indicates the expected result of a successful multicast ping: echo replies from all members of the multicast group. Notice that the ping is sent out all interfaces, including the loopback interface on R1. Because there could technically be multicast listeners to that group on other connected subnets or even remote networks, multicasts must be flooded out all interfaces, unless a multicast routing protocol informs the router that this is unnecessary.

If R2 and R3 received the multicast ping, how did SW1 treat the ping at OSI Layer 2?

SW1 treated the multicast ping as a frame with an all-ones destination address, which will act as a broadcast. SW1 forwarded the frame to all devices connected to VLAN 1.

To which address did R2 and R3 send their replies?

R2 and R3 replied to the unicast source IP address of R1's Fast Ethernet interface. At Layer 2, they used the unicast MAC address of R1's Fast Ethernet interface as the destination address of their replies.

Is this a unicast or multicast address?

This is a unicast IP address.

Ping from R2 to the multicast group. From which hosts do you expect replies?

R2 should receive ICMP echo replies from all of the members of the 229.7.7.7 multicast group, R2 and R3.

If the output of your ping has only one reply, as shown below, re-issue the ping. With unicast pings, the first ping usually times out due to the delay at Layer 2 caused by the ARP request from the source to the destination.

```
R2# ping 229.7.7.7
```

```
Type escape sequence to abort.  
Sending 1, 100-byte ICMP Echos to 229.7.7.7, timeout is 2 seconds:
```

```
Reply to request 0 from 192.168.1.2, 1 ms
```

For R3 to send the ICMP echo reply to 192.168.1.2, it needs to locate a MAC address for that IP address via ARP. Thus, the ARP request from the multicast listener to the multicast receiver may cause the ICMP echo to time out at the source. Try to ping again with the same command. You should now receive all ICMP echo replies, as follows:

```
R2# ping 229.7.7.7
```

```
Type escape sequence to abort.  
Sending 1, 100-byte ICMP Echos to 229.7.7.7, timeout is 2 seconds:
```

```
*Nov  4 16:19:28.117: IP: s=192.168.1.2 (local), d=229.7.7.7  
(FastEthernet0/0), len 100, sending broad/multicast  
*Nov  4 16:19:28.117: IP: s=172.16.2.1 (local), d=229.7.7.7 (Loopback2), len  
100, sending broad/multicast  
*Nov  4 16:19:28.117: IP: s=192.168.1.2 (FastEthernet0/0), d=229.7.7.7, len  
100, rcvd 1  
*Nov  4 16:19:28.117: IP: tableid=0, s=192.168.1.2 (local), d=192.168.1.2  
(FastEthernet0/0), routed via RIB
```

```
*Nov  4 16:19:28.117: IP: s=192.168.1.2 (local), d=192.168.1.2
(FastEthernet0/0), len 100, sending
*Nov  4 16:19:28.121: IP: s=192.168.1.3 (FastEthernet0/0), d=192.168.1.2, len
100, rcvd 1
*Nov  4 16:19:28.121: IP: s=192.168.1.2 (FastEthernet0/0), d=192.168.1.2, len
100, rcvd 1
Reply to request 0 from 192.168.1.3, 4 ms
Reply to request 0 from 192.168.1.2, 4 ms
```

For more information on IP multicast and IGMP, see RFC 1112 and RFC 2236:
<http://www.ietf.org/rfc/rfc1112.txt> <http://www.ietf.org/rfc/rfc2236.txt>.

Step 3: Verify IGMP Snooping on the Switch

Up to this point, multicast has been simplified to be multicast hosts on a broadcast, multi-access medium of a switched VLAN. The switch treated these multicast packets as broadcasts at Layer 2 and forwarded them to every host on the VLAN, regardless of whether that host was subscribed to the group. Sending traffic to every host on the subnet effectively reduces multicasting to broadcasting with a different destination address. Multicast switching in an enterprise network should be designed in a more intelligent way at Layer 2.

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

The multicast router sends out periodic IGMP general queries to all VLANs. When IGMP snooping is enabled, the switch responds to the router queries with only one join request per MAC multicast group, and the switch creates one entry per VLAN in the Layer 2 forwarding table for each MAC group from which it receives an IGMP join request. All hosts interested in this multicast traffic send join requests and are added to the forwarding table entry.

Layer 2 multicast groups learned through IGMP snooping are dynamic. However, you can statically configure MAC multicast groups by using the **ip igmp snooping vlan static** global configuration command. If you specify group membership for a multicast group address statically, your setting supersedes any automatic manipulation by IGMP snooping. Multicast group membership lists can consist of both user-defined and IGMP snooping-learned settings.

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

IGMP snooping is enabled on Catalyst switches by default.

Issue the **show ip igmp snooping** command on SW1.

```
SW1# show ip igmp snooping
vlan 1
-----
IGMP snooping is globally enabled
IGMP snooping is enabled on this Vlan
IGMP snooping immediate-leave is disabled on this Vlan
IGMP snooping mrouter learn mode is pim-dvmrp on this Vlan
IGMP snooping is running in IGMP_ONLY mode on this Vlan
```

Notice that IGMP snooping is enabled by default globally and on a per-VLAN basis on SW1. In this case, IGMP snooping identifies a switchport as a multicast router port only if it see PIM or DVMRP messages sent toward the switch on that port.

IGMP snooping only subscribes other switchports to multicast groups at Layer 2 if it sees IGMP messages sent to the multicast router. Until you configure a supported multicast routing protocol (PIM or DVMRP) on the multicast router R1, IGMP snooping does not create MAC address table entries for the multicast groups. Verify this with the **show mac address-table multicast** command on SW1.

```
SW1# show mac address-table multicast
Vlan      Mac Address      Type      Ports
----      -
SW1#
```

Step 4: Configure a Multicast-Enabled Router on the VLAN

First, enable IP routing on R1 with the **ip routing** command.

```
R1(config)# ip routing
```

Enable multicast routing on R1 using the **ip multicast-routing** command in global configuration mode, which is similar to the **ip routing** command. The **ip multicast-routing** command allows R1 to start obtaining current information on multicast groups, sources, destination, and routing patterns.

```
R1(config)# ip multicast-routing
```

Enable PIM-DM on R1's FastEthernet0/0 interface with the **ip pim dense-mode** command. Identify the output with debugging turned on.

```
R1(config-if)# ip pim dense-mode
R1(config-if)#
*Nov  5 00:28:24.687: IP: s=192.168.1.1 (local), d=224.0.0.13
(FastEthernet0/0), len 54, sending broad/multicast
```

PIM attempts to make adjacencies out interfaces on which it is enabled. The multicast packet to 224.0.0.13 equates with a PIM Hello packet to all PIM-enabled routers on the subnet.

```
R1(config-if)#
*Nov 5 00:28:24.687: IGMP(0): Send v2 init Query on FastEthernet0/0
*Nov 5 00:28:24.687: IP: s=192.168.1.1 (local), d=224.0.0.1
(FastEthernet0/0), len 28, sending broad/multicast
```

Next, IGMP sends the initial IGMP query to the multicast group referencing all devices on the subnet 224.0.0.1. Each IGMP-enabled interface connected to the VLAN should now send an IGMP join message to 192.168.1.1 indicating which multicast groups it wishes to join.

```
*Nov 5 00:28:24.691: IGMP(0): WAVL Insert group: 224.0.1.40 interface:
FastEthernet0/0Successful
*Nov 5 00:28:24.691: IGMP(0): Send v2 Report for 224.0.1.40 on
FastEthernet0/0
*Nov 5 00:28:24.691: IGMP(0): Received v2 Report on FastEthernet0/0 from
192.168.1.1 for 224.0.1.40
R1(config-if)#
*Nov 5 00:28:24.691: IGMP(0): Received Group record for group 224.0.1.40,
mode 2 from 192.168.1.1 for 0 sources
*Nov 5 00:28:24.691: IGMP(0): Switching to EXCLUDE mode for 224.0.1.40 on
FastEthernet0/0
*Nov 5 00:28:24.691: IGMP(0): Updating EXCLUDE group timer for 224.0.1.40
*Nov 5 00:28:25.331: IGMP(0): MRT Add/Update FastEthernet0/0 for
(*,224.0.1.40) by 0
*Nov 5 00:28:24.691: IP: s=192.168.1.1 (local), d=224.0.1.40
(FastEthernet0/0), len 28, sending broad/multicast
```

All PIM routers automatically subscribe to the 224.0.1.40 group. Therefore, R1 sends an IGMP join message on its Fast Ethernet interface for the 224.0.1.40 group. Then, R1 receives its own message and implicitly joins its Fast Ethernet interface to the group. You will learn more about the 224.0.1.40 group in Lab 7.4.

The relevant parts of the remaining output are highlighted.

```
*Nov 5 00:28:25.331: IP: s=192.168.1.2 (FastEthernet0/0), d=229.7.7.7, len
28, rcvd 0
*Nov 5 00:28:25.331: IGMP(0): Received v2 Report on FastEthernet0/0 from
192.168.1.2 for 229.7.7.7
*Nov 5 00:28:25.331: IGMP(0): Received Group record for group 229.7.7.7, mode
2 from 192.168.1.2 for 0 sources
*Nov 5 00:28:25.331: IGMP(0): WAVL Insert group: 229.7.7.7 interface:
FastEthernet0/0Successful
*Nov 5 00:28:25.331: IGMP(0): Switching to EXCLUDE mode for 229.7.7.7 on
FastEthernet0/0
*Nov 5 00:28:25.331: IGMP(0): Updating EXCLUDE group timer for 229.7.7.7
*Nov 5 00:28:25.331: IGMP(0): MRT Add/Update FastEthernet0/0 for
(*,229.7.7.7) by 0
*Nov 5 00:28:25.687: %PIM-5-DRCHG: DR change from neighbor 0.0.0.0 to
192.168.1.1 on interface FastEthernet0/0 (vrf default)
*Nov 5 00:28:53.979: IP: s=192.168.1.1 (local), d=224.0.0.13
(FastEthernet0/0), len 54, sending broad/multicast
*Nov 5 00:29:23.415: IP: s=192.168.1.1 (local), d=224.0.0.13
(FastEthernet0/0), len 54, sending broad/multicast
*Nov 5 00:29:23.695: IGMP(0): Send v2 general Query on FastEthernet0/0
*Nov 5 00:29:23.695: IGMP(0): Set report delay time to 4.6 seconds for
224.0.1.40 on FastEthernet0/0
```

```

*Nov 5 00:29:23.695: IP: s=192.168.1.1 (local), d=224.0.0.1
(FastEthernet0/0), len 28, sending broad/multicast
*Nov 5 00:29:28.695: IGMP(0): Send v2 Report for 224.0.1.40 on
FastEthernet0/0
*Nov 5 00:29:28.695: IGMP(0): Received v2 Report on FastEthernet0/0 from
192.168.1.1 for 224.0.1.40
*Nov 5 00:29:28.695: IGMP(0): Received Group record for group 224.0.1.40,
mode 2 from 192.168.1.1 for 0 sources
*Nov 5 00:29:28.695: IGMP(0): Updating EXCLUDE group timer for 224.0.1.40
*Nov 5 00:29:28.695: IP: s=192.168.1.1 (local), d=224.0.1.40
(FastEthernet0/0), len 28, sending broad/multicast
*Nov 5 00:29:29.331: IP: s=192.168.1.2 (FastEthernet0/0), d=229.7.7.7, len
28, rcvd 0
*Nov 5 00:29:29.331: IGMP(0): Received v2 Report on FastEthernet0/0 from
192.168.1.2 for 229.7.7.7
*Nov 5 00:29:29.331: IGMP(0): Received Group record for group 229.7.7.7, mode
2 from 192.168.1.2 for 0 sources
*Nov 5 00:29:29.331: IGMP(0): Updating EXCLUDE group timer for 229.7.7.7
*Nov 5 00:29:29.331: IGMP(0): MRT Add/Update FastEthernet0/0 for
(*,229.7.7.7) by 0
*Nov 5 00:29:53.111: IP: s=192.168.1.1 (local), d=224.0.0.13
(FastEthernet0/0), len 54, sending broad/multicast
*Nov 5 00:30:22.819: IP: s=192.168.1.1 (local), d=224.0.0.13
(FastEthernet0/0), len 54, sending broad/multicast
*Nov 5 00:30:23.695: IGMP(0): Send v2 general Query on FastEthernet0/0
*Nov 5 00:30:23.695: IGMP(0): Set report delay time to 4.8 seconds for
224.0.1.40 on FastEthernet0/0
*Nov 5 00:30:23.695: IP: s=192.168.1.1 (local), d=224.0.0.1
(FastEthernet0/0), len 28, sending broad/multicast
*Nov 5 00:30:27.331: IP: s=192.168.1.2 (FastEthernet0/0), d=229.7.7.7, len
28, rcvd 0
*Nov 5 00:30:27.331: IGMP(0): Received v2 Report on FastEthernet0/0 from
192.168.1.2 for 229.7.7.7
*Nov 5 00:30:27.331: IGMP(0): Received Group record for group 229.7.7.7, mode
2 from 192.168.1.2 for 0 sources
*Nov 5 00:30:27.331: IGMP(0): Updating EXCLUDE group timer for 229.7.7.7
*Nov 5 00:30:27.331: IGMP(0): MRT Add/Update FastEthernet0/0 for
(*,229.7.7.7) by 0
*Nov 5 00:30:28.695: IGMP(0): Send v2 Report for 224.0.1.40 on
FastEthernet0/0
*Nov 5 00:30:28.695: IGMP(0): Received v2 Report on FastEthernet0/0 from
192.168.1.1 for 224.0.1.40
*Nov 5 00:30:28.695: IGMP(0): Received Group record for group 224.0.1.40,
mode 2 from 192.168.1.1 for 0 sources
*Nov 5 00:30:28.695: IGMP(0): Updating EXCLUDE group timer for 224.0.1.40
*Nov 5 00:30:28.695: IP: s=192.168.1.1 (local), d=224.0.1.40
(FastEthernet0/0), len 28, sending broad/multicast
*Nov 5 00:30:52.155: IP: s=192.168.1.1 (local), d=224.0.0.13
(FastEthernet0/0), len 54, sending broad/multicast
*Nov 5 00:31:22.079: IP: s=192.168.1.1 (local), d=224.0.0.13
(FastEthernet0/0), len 54, sending broad/multicast
*Nov 5 00:31:23.695: IGMP(0): Send v2 general Query on FastEthernet0/0
*Nov 5 00:31:23.695: IGMP(0): Set report delay time to 9.8 seconds for
224.0.1.40 on FastEthernet0/0
*Nov 5 00:31:23.695: IP: s=192.168.1.1 (local), d=224.0.0.1
(FastEthernet0/0), len 28, sending broad/multicast
*Nov 5 00:31:27.503: IP: s=192.168.1.3 (FastEthernet0/0), d=229.7.7.7, len
28, rcvd 0
*Nov 5 00:31:27.503: IGMP(0): Received v2 Report on FastEthernet0/0 from
192.168.1.3 for 229.7.7.7
*Nov 5 00:31:27.503: IGMP(0): Received Group record for group 229.7.7.7, mode
2 from 192.168.1.3 for 0 sources
*Nov 5 00:31:27.503: IGMP(0): Updating EXCLUDE group timer for 229.7.7.7

```

```
*Nov 5 00:31:27.503: IGMP(0): MRT Add/Update FastEthernet0/0 for
(*,229.7.7.7) by 0
```

The highlighted messages indicate the IGMP queries and responses. In each case, R1 sends a periodic IGMP query to all devices on the subnet. R2 and R3 periodically respond to those queries as necessary, requesting that multicast traffic to the 229.7.7.7 group be allowed onto this subnet. PIM-DM periodically floods traffic to a Layer 2 segment and then prunes that traffic from that segment if no listeners are configured.

At this point, disable debugging on all routers.

```
Router# undebug all
```

Step 5: Verify Multicast Operation at Layer 2

Which IP multicast groups do you expect to see in the IGMP records on R1?

Because R1's Fast Ethernet interface is the IGMP querier for the subnet, R1 will possess records for all of the IGMP groups configured on the subnet. At this point, the 229.7.7.7 group has been configured on R2 and R3. PIM uses 224.0.1.40 for communication in PIM sparse mode, R1's Fast Ethernet interface automatically subscribes to this group. This reserved control group will be explored in more detail in Lab 7.4.

On R1, issue the **show ip igmp groups** command to display the multicast groups for which IGMP has currently recorded information from connected interfaces.

```
R1# show ip igmp groups
IGMP Connected Group Membership
Group Address      Interface          Uptime    Expires    Last Reporter
Group Accounted
229.7.7.7          FastEthernet0/0   00:02:19  00:02:19  192.168.1.3
224.0.1.40         FastEthernet0/0   00:02:22  00:02:22  192.168.1.1
```

Display the IGMP membership for the IGMP groups reported by R1 using the **show ip igmp membership** command.

```
R1# show ip igmp membership
Flags: A - aggregate, T - tracked
       L - Local, S - static, V - virtual, R - Reported through v3
       I - v3lite, U - Urd, M - SSM (S,G) channel
       1,2,3 - The version of IGMP the group is in
Channel/Group-Flags:
/ - Filtering entry (Exclude mode (S,G), Include mode (*,G))
Reporter:
<mac-or-ip-address> - last reporter if group is not explicitly tracked
<n>/<m>              - <n> reporter in include mode, <m> reporter in exclude

Channel/Group      Reporter          Uptime    Exp.    Flags
Interface
```

```

*,229.7.7.7          192.168.1.3    00:02:20 02:20 2A    Fa0/0
*,224.0.1.40        192.168.1.1    00:02:23 02:23 2LA    Fa0/0

```

Display the IGMP status on R1's FastEthernet0/0 interface with the **show ip igmp interface** command.

```

R1# show ip igmp interface
FastEthernet0/0 is up, line protocol is up
  Internet address is 192.168.1.1/24
  IGMP is enabled on interface
  Current IGMP host version is 2
  Current IGMP router version is 2
  IGMP query interval is 60 seconds
  IGMP querier timeout is 120 seconds
  IGMP max query response time is 10 seconds
  Last member query count is 2
  Last member query response interval is 1000 ms
  Inbound IGMP access group is not set
  IGMP activity: 2 joins, 0 leaves
  Multicast routing is enabled on interface
  Multicast TTL threshold is 0
  Multicast designated router (DR) is 192.168.1.1 (this system)
  IGMP querying router is 192.168.1.1 (this system)
  Multicast groups joined by this system (number of users):
    224.0.1.40(1)

```

Step 6: Verify IGMP Snooping

On SW1, issue the **show mac address-table multicast**.

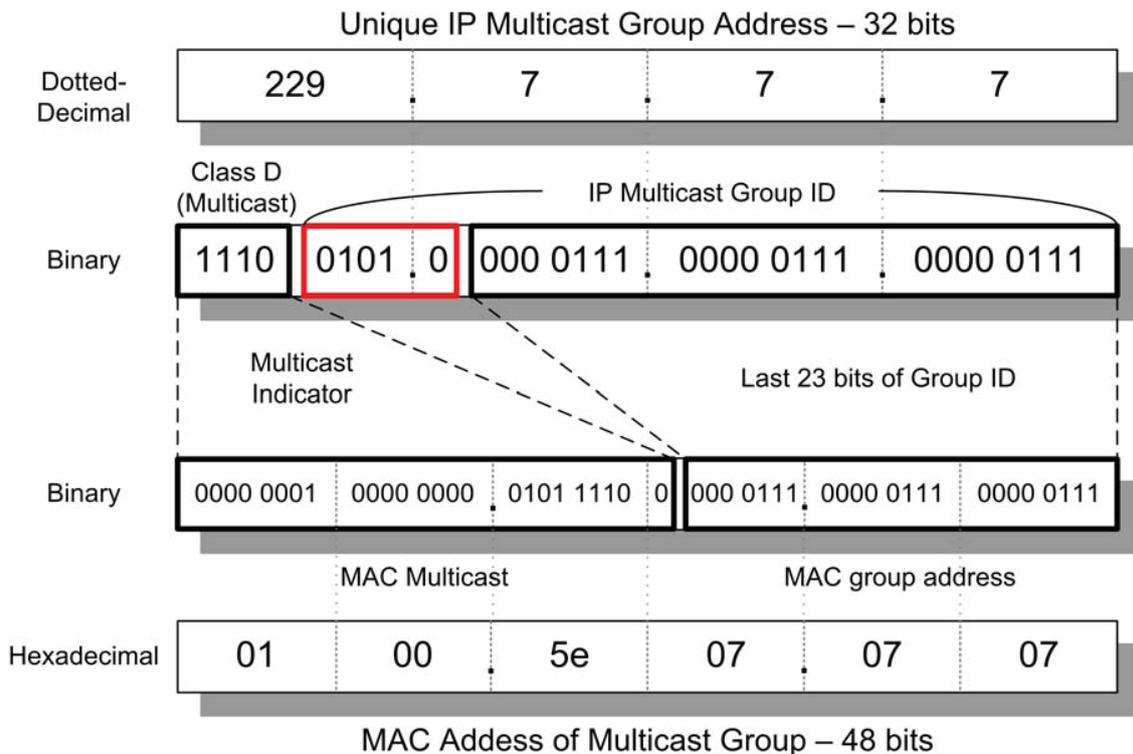
```

SW1# show mac address-table multicast
Vlan    Mac Address          Type    Ports
----    -
1       0100.5e00.0128      IGMP   Fa0/1
1       0100.5e07.0707      IGMP   Fa0/1, Fa0/3, Fa0/5

```

These MAC addresses are not unique to one IP multicast group, because there is a 1:32 correlation between each MAC address and the IP multicast groups it represents.

Consider the binary form of the group address for 229.7.7.7:



Because the center set of boxed bits in the binary representation of the IP address is not mapped to any bits in the MAC address, the MAC addresses of multicast groups are not unique to an IP multicast group. Class D addresses with any value in those 5 bits and a constant sequence of bits in the 23 least-significant digits all map to the same MAC address.

List at least three IP multicast groups with which the following MAC addresses correspond.

- 0100.5e00.0128

There are multiple answers. Each correct multicast IP address answer will be in the Class D range and have the binary form 1110xxxx.x0000000.00000001.00100100, where x indicates that the bits may have any value and still be correct.

Here are some valid answers in sequence: 224.0.1.40, 224.128.1.40, 225.0.1.40, 225.128.1.40, 226.0.1.40, 226.128.1.40, ..., 239.0.1.40, 239.128.1.40.

- 0100.5e07.0707

There are multiple answers. Each correct multicast IP address answer will be in the Class D range and have the binary form 1110xxxx.x0000111.00000111.00000111, where x indicates that the bits may have any value and still be correct.

Here are some valid answers in sequence: 224.7.7.7, 224.135.7.7, 225.7.7.7, 225.135.7.7, 226.7.7.7, 226.135.7.7, ..., 239.7.7.7, 239.135.7.7.

In this particular scenario, which IP multicast groups do the 0100.5e00.0128 and 0100.5e07.0707 MAC addresses represent?

In this scenario, 0100.5e00.0128 represents the 224.0.1.40 multicast group address. 0100.5e07.0707 represents the 224.7.7.7 multicast group address.

Although all 32 groups are forwarded at Layer 2 to the Layer 3 interfaces of the routers and hosts, IGMP only reads IP packets for groups to which the interface has subscribed or for which it is in a forwarding state. All other multicast traffic is dropped.

As discussed in Step 3, statically subscribe FastEthernet0/9 on SW1 to the multicast MAC group of 0100.5e07.07.07 via the **ip igmp snooping vlan static** global configuration command. This configuration does not affect the remainder of the lab.

```
SW1(config)# ip igmp snooping vlan 1 static 0100.5e07.0707 interface fastethernet 0/9
```

Reissue the **show mac address-table multicast** command on SW1.

```
SW1# show mac address-table multicast
Vlan      Mac Address      Type      Ports
----      -
1         0100.5e00.0128   IGMP      Fa0/1
1         0100.5e07.0707   USER     Fa0/1, Fa0/3, Fa0/5,
                                         Fa0/9
```

Notice that the Type field for the 0100.5e07.0707 entry now indicates that it is a statically configured group address. Statically configured addresses override mappings dynamically learned via IGMP snooping.

Step 7: Verify Multicast Operation at Layer 3

Verify that multicast route state has been recorded on R1 using the **show ip mroute** command.

```
R1# show ip mroute
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
```

```

L - Local, P - Pruned, R - RP-bit set, F - Register flag,
T - SPT-bit set, J - Join SPT, M - MSDP created entry,
X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
U - URD, I - Received Source Specific Host Report,
Z - Multicast Tunnel, z - MDT-data group sender,
Y - Joined MDT-data group, y - Sending to MDT-data group
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

```

```

(*, 229.7.7.7), 00:06:48/00:02:11, RP 0.0.0.0, flags: DC
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    FastEthernet0/0, Forward/Dense, 00:06:48/00:00:00

```

```

(*, 224.0.1.40), 00:07:33/00:02:15, RP 0.0.0.0, flags: DCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    FastEthernet0/0, Forward/Dense, 00:07:33/00:00:00

```

The **show ip mroute** command shows multicast state with reference to each multicast group. Multicast routes are more complex than unicast routes because the router needs to identify incoming interfaces which it considers the “upstream interface” to specific multicast sources using a reverse-path check. The multicast router must also store the outgoing interfaces that should forward the multicast traffic for each source and group.

Without PIM communicating between routers, much of the interface and RPF neighbor information the multicast routing table shown above is default information, such as RPF neighbors of 0.0.0.0. You will see more complex router state in the next lab.

Notice in the output shown that all traffic to the 229.7.7.7 group should be forwarded out FastEthernet0/0. Verify this by pinging from R1 to 229.7.7.7.

```

R1# ping 229.7.7.7
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 229.7.7.7, timeout is 2 seconds:

Reply to request 0 from 192.168.1.3, 1 ms
Reply to request 0 from 192.168.1.2, 1 ms

```

Final Configurations

```

R1# show run
!
hostname R1
!
ip multicast-routing
!
interface FastEthernet0/0
 ip address 192.168.1.1 255.255.255.0
 ip pim dense-mode
 no shutdown
!
end

```

```
R2# show run
!  
hostname R2  
!  
interface FastEthernet0/0  
 ip address 192.168.1.2 255.255.255.0  
 ip igmp join-group 229.7.7.7  
 no shutdown  
!  
end
```

```
R3# show run  
!  
hostname R3  
!  
interface FastEthernet0/0  
 ip address 192.168.1.3 255.255.255.0  
 ip igmp join-group 229.7.7.7  
 no shutdown  
!  
end
```

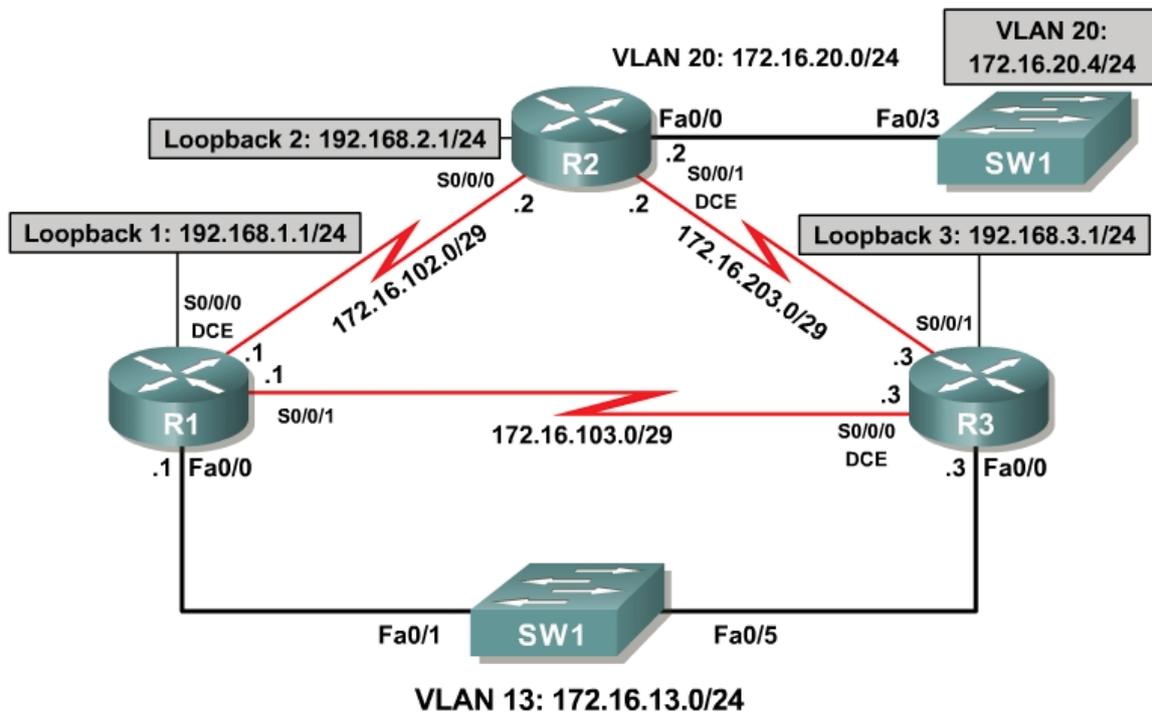
```
SW1# show run  
!  
hostname SW1  
!  
ip igmp snooping vlan 1 static 0100.5e07.0707 interface Fa0/9  
!  
interface FastEthernet0/1  
 switchport mode access  
!  
interface FastEthernet0/3  
 switchport mode access  
!  
interface FastEthernet0/5  
 switchport mode access  
!  
interface Vlan1  
 ip address 192.168.1.10 255.255.255.0  
 no shutdown  
!  
end
```

Lab 7-2 Routing IP Multicast with PIM Dense Mode

Learning Objectives

- Implement IGMP
- Review configuration of EIGRP
- Implement and verify PIM-DM operation and adjacencies
- Verify IGMP operation of PIM-DM flooding and pruning
- Explore the multicast routing table

Topology Diagram



Scenario

A community action organization, CrushASodaCan.com, contracted you as a consultant to set up IP multicast in their network. At the end of the week, the public relations department wants to multicast a live video feed of the CEO crushing a soda can to encourage citizens to compact their recyclable material before sending it to the recycling center. As a consultant working on this network, you are concerned about the use of the network resources.

The three remote sites are connected by a full mesh of serial leased lines. Sites 1 and 3 (R1 and R3) are also connected via Fast Ethernet.

The multicast source is a host on VLAN 20, which connects at Layer 3 to R2's Fast Ethernet interface. A remote network at each site represented by a loopback interface should receive the press feed of the CEO's message. While performing testing in this scenario, you will use a switched virtual interface (SVI) on SW1 to simulate the multicast source. At each multicast group member interface network, you should prefer the shortest cost path to the multicast source according to the underlying unicast routing table.

Step 1: Configure Addressing and Implement IGMP

Paste in the following initial configurations:

```
R1:
!
hostname R1
!
interface Loopback1
 ip address 192.168.1.1 255.255.255.0
!
interface FastEthernet0/0
 ip address 172.16.13.1 255.255.255.0
 no shutdown
!
interface Serial0/0/0
 bandwidth 64
 ip address 172.16.102.1 255.255.255.248
 clock rate 64000
 no shutdown
!
interface Serial0/0/1
 bandwidth 64
 ip address 172.16.103.1 255.255.255.248
 no shutdown
!
end

R2:
!
hostname R2
!
interface Loopback2
 ip address 192.168.2.1 255.255.255.0
!
interface FastEthernet0/0
 ip address 172.16.20.2 255.255.255.0
 no shutdown
!
interface Serial0/0/0
 bandwidth 64
 ip address 172.16.102.2 255.255.255.248
 no shutdown
!
interface Serial0/0/1
 bandwidth 128
 ip address 172.16.203.2 255.255.255.248
 clock rate 128000
```

```

    no shutdown
    !
end

R3:
!
hostname R3
!
interface Loopback3
 ip address 192.168.3.1 255.255.255.0
!
interface FastEthernet0/0
 ip address 172.16.13.3 255.255.255.0
 no shutdown
!
interface Serial0/0/0
 bandwidth 64
 ip address 172.16.103.3 255.255.255.248
 clock rate 64000
 no shutdown
!
interface Serial0/0/1
 bandwidth 128
 ip address 172.16.203.3 255.255.255.248
 no shutdown
!
end

SW1:
!
hostname SW1
!
interface FastEthernet0/1
 switchport access vlan 13
 switchport mode access
!
interface FastEthernet0/3
 switchport access vlan 20
 switchport mode access
!
interface FastEthernet0/5
 switchport access vlan 13
 switchport mode access
!
end

```

Use a switched virtual interface (SVI) on SW1 to simulate a multicast source on the VLAN 20 subnet. This will be used to generate a repeated multicast ping to simulate the multicast traffic while you set up the network. Assign the SVI the IP address 172.16.20.4/24 with a default gateway of 172.16.20.2.

```

SW1# conf t
SW1(config)# ip default-gateway 172.16.20.2
SW1(config)# interface vlan 20
SW1(config-if)# ip address 172.16.20.4 255.255.255.0
SW1(config-if)# no shutdown

```

Using IGMP, subscribe each of the loopback interfaces on the three routers to the multicast group 232.32.32.32.

```

R1# conf t

```

```
R1(config)# interface loopback 1
R1(config-if)# ip igmp join-group 232.32.32.32
```

```
R2# conf t
R2(config)# interface loopback 2
R2(config-if)# ip igmp join-group 232.32.32.32
```

```
R3# conf t
R3(config)# interface loopback 3
R3(config-if)# ip igmp join-group 232.32.32.32
```

Verify that each of the interfaces has subscribed to the multicast group using the **show ip igmp groups** command.

```
R1# show ip igmp groups
IGMP Connected Group Membership
Group Address   Interface          Uptime    Expires    Last Reporter
Group Accounted
232.32.32.32    Loopback1         00:02:13  stopped    192.168.1.1
```

```
R2# show ip igmp groups
IGMP Connected Group Membership
Group Address   Interface          Uptime    Expires    Last Reporter
Group Accounted
232.32.32.32    Loopback2         00:02:15  stopped    192.168.1.2
```

```
R3# show ip igmp groups
IGMP Connected Group Membership
Group Address   Interface          Uptime    Expires    Last Reporter
Group Accounted
232.32.32.32    Loopback3         00:02:17  stopped    192.168.1.3
```

Step 2: Configure EIGRP

Configure EIGRP on each router with the following configuration:

```
router eigrp 1
 network 192.168.0.0 0.0.255.255
 network 172.16.0.0
```

After the EIGRP adjacencies form, run the following TCL script to verify full unicast connectivity:

```
foreach address {
192.168.1.1
172.16.13.1
172.16.102.1
172.16.103.1
192.168.2.1
172.16.20.2
172.16.102.2
172.16.203.2
192.168.3.1
172.16.13.3
172.16.103.3
172.16.203.3
172.16.20.4
} { ping $address }
```

Compare the output you receive with the output in Appendix A (all pings successful). Make sure you have ICMP replies from the host. If your host has a reachable default gateway of R2's FastEthernet0/0 interface and if all routers have full connectivity between all subnets, you should be able to ping the host and receive all replies.

Step 3: Implement PIM-DM

Ping the 232.32.32.32 multicast address from the VLAN 20 SVI on SW1 discussed in Step 2.

```
SW1# ping 232.32.32.32
```

```
Type escape sequence to abort.
```

```
Sending 1, 100-byte ICMP Echos to 232.32.32.32, timeout is 2 seconds:
```

```
SW1#
```

Did you receive ICMP echo replies from the loopback interfaces subscribed to this group via IGMP?

No echo replies will be received. No multicast packets will be routed from one interface to another interface until you later configure the **ip multicast-routing** and **ip pim dense-mode** commands.

Why does the host not receive ICMP replies?

No packets are routed from one interface to another, so the multicast packets are not forwarded. Since no packets will reach the IGMP multicast group subscribers, the subscribers do not have the opportunity to respond to the pings.

List at least three responsibilities of an IP multicast routing protocol.

1. IP multicast routing protocols determine the "upstream interface" for every source so that each sourced packet will only be forwarded once by each router.
2. IP multicast routing protocols define which router interfaces are "downstream interfaces." It is out downstream interfaces that a router will forward multicast packets to receivers.
3. IP multicast routing protocols manage the shared and source trees on a node-by-node basis, grafting and pruning when necessary.

Routing protocols are primarily useful for “introducing” multicast sources to their receivers, a procedure known as source discovery. In unicast IP networks, a Layer 3 device must know the next downstream hop or interface to route a unicast packet to a destination network. Similarly, an IP multicast router must know the tree along which multicast data from a source should flow to the receivers.

Protocol Independent Multicast (PIM) has two modes: sparse mode (SM) and dense mode (DM). Cisco IOS supports both modes. PIM-DM is designed for networks in which devices on almost every subnet subscribe to the available groups. PIM-DM is useful in lab settings and in situations in which most of the Layer 2 segments involved have subscriptions to the multicast groups. PIM-SM is designed for multicast networks in which multicast groups do not have subscribers on many subnets.

Note: Cisco does not recommend implementing PIM-DM in production networks because of the inefficiency inherent in the push-based model. Though a multicast group does not have subscribers on many subnets in the network, multicast packets are still periodically flooded and pruned throughout the entire multicast network.

PIM-DM creates (S, G) multicast trees rooted at each source that form shortest-cost paths to the receivers throughout the network. Each multicast router uses the unicast IP routing table to track which interfaces are upstream with reference to each multicast source. PIM-DM also analyzes the downstream subscriptions to assign forwarding state to outgoing interfaces. When a PIM router receives a packet for a multicast group, it must determine which interfaces for that group are in a forwarding state before sending the packet.

To enable multicast routing, issue the **ip multicast-routing** command in global configuration mode.

```
R1(config)# ip multicast-routing
```

```
R2(config)# ip multicast-routing
```

```
R3(config)# ip multicast-routing
```

To enable PIM-DM on the Loopback2 and FastEthernet0/0 interfaces on R2, issue the **ip pim dense-mode** command in interface configuration mode.

```
R2(config)# interface loopback 2
R2(config-if)# ip pim dense-mode
R2(config-if)# interface fastethernet 0/0
R2(config-if)# ip pim dense-mode
```

Recall the behavior of IGMP when PIM-DM was configured on R1’s Fast Ethernet interface in Lab 7.1. Will IGMP queries be sent over VLAN 20?

Yes. When PIM Dense Mode is enabled on an interface, the Layer 2 segment will elect an IGMP querier. In this case, the IGMP querier for VLAN 20 will be R2's Fast Ethernet interface.

Now that multicast routing and PIM-DM are enabled on R2, will a ping from SW1 to the 232.32.32.32 group receive any replies? Explain.

Yes, the multicast routing protocol, PIM-DM, is now running on a node-by-node basis on R2. R2 will determine the Fast Ethernet 0/0 interface as the upstream interface and forward packets down the source tree to any downstream interfaces. R2's loopback interface is a downstream interface. The loopback interface receives the multicast packet and replies to SW1, sourcing the reply from its nearest interface according to the unicast routing table.

Verify your answer by pinging the multicast group from SW1:

```
SW1# ping 232.32.32.32
```

```
Type escape sequence to abort.
```

```
Sending 1, 100-byte ICMP Echos to 232.32.32.32, timeout is 2 seconds:
```

```
Reply to request 0 from 172.16.20.2, 4 ms
```

Why was the first packet dropped?

The first packet might have been dropped if R2 did not have an ARP entry for SW1's source IP/MAC address pair.

Why did the host not receive replies from the other loopback interfaces in your topology?

PIM is not running on R2 or R3, therefore R1 is not aware of any other multicast group members aside from those on directly connected networks. R2 must be directed to forward multicasts to remote devices via PIM/IGMP messages in order to forward traffic beyond the local router.

In Lab 7.1, you demonstrated that without IGMP snooping Layer 2 multicast traffic traveling over Ethernet is treated as broadcast traffic. The switch then forwards multicast traffic to all ports that would normally receive a broadcast from the same source. However, at OSI Layer 3, multicast traffic is discarded unless the router receives directives from IGMP to send traffic out certain

egress interfaces. If IGMP had registered such messages, the state would be shown in the IP multicast routing table.

Display the multicast routing table to check for egress interfaces to forward multicast traffic to group 232.32.32.32.

```
R2# show ip mroute
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 232.32.32.32), 00:25:11/stopped, RP 0.0.0.0, flags: DCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Loopback2, Forward/Dense, 00:25:11/00:00:00

(172.16.20.4, 232.32.32.32), 00:00:07/00:02:55, flags: LT
  Incoming interface: FastEthernet0/0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Loopback2, Forward/Dense, 00:00:07/00:00:00

(*, 224.0.1.40), 00:25:12/00:02:32, RP 0.0.0.0, flags: DCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Loopback2, Forward/Dense, 00:25:12/00:00:00
```

Because IGMP only queries interfaces that are running PIM-DM, the subscriptions for the remote loopback interfaces have not been reported to R2 and are, therefore, in a non-forwarding state. Only R2's Loopback2 interface will receive multicast packets destined for 232.32.32.32.

You can resolve this issue by applying the **ip pim dense-mode** command to each remaining interface in the topology.

```
R1(config)# interface loopback 1
R1(config-if)# ip pim dense-mode
R1(config-if)# interface fastethernet 0/0
R1(config-if)# ip pim dense-mode
R1(config-if)# interface serial 0/0/0
R1(config-if)# ip pim dense-mode
R1(config-if)# interface serial 0/0/1
R1(config-if)# ip pim dense-mode

R2(config)# interface serial 0/0/0
R2(config-if)# ip pim dense-mode
R2(config-if)# interface serial 0/0/1
R2(config-if)# ip pim dense-mode

R3(config)# interface loopback 1
R3(config-if)# ip pim dense-mode
R3(config-if)# interface fastethernet 0/0
```

```

R3(config-if)# ip pim dense-mode
R3(config-if)# interface serial 0/0/0
R3(config-if)# ip pim dense-mode
R3(config-if)# interface serial 0/0/1
R3(config-if)# ip pim dense-mode

```

Ping the multicast group from SW1 again. You should receive replies from each router. Note that you will not receive replies from the IP address of the interface on which the multicast packet was received but rather from whichever interface on the responding router encapsulated the return packet.

```
SW1# ping 232.32.32.32
```

```
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 232.32.32.32, timeout is 2 seconds:
```

```

Reply to request 0 from 172.16.20.2, 4 ms
Reply to request 0 from 172.16.102.1, 32 ms
Reply to request 0 from 172.16.203.3, 32 ms

```

Based on your understanding of IP unicast routing, why are the ICMP echo replies sent from the interfaces shown above?

When a router originates packets to a remote destination, it uses as a source address the IP address of the interface out of which it will send those packets. In this case, the packets will not be sent out of the loopback interface (though that is where they were received) but out the serial interface nearest the 172.16.20.0/24 network.

Step 4: Verify PIM Adjacencies

In this step, we explore PIM adjacencies and how PIM functions over various Layer 2 media.

Issue the **show ip pim neighbors** command to display all the routers on the connected Layer 3 subnets.

```

R1# show ip pim neighbor
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      S - State Refresh Capable
Neighbor      Interface      Uptime/Expires    Ver    DR
Address
172.16.13.3   FastEthernet0/0 00:02:29/00:01:42 v2     1 / DR S
172.16.102.2  Serial0/0/0     00:02:30/00:01:40 v2     1 / S
172.16.103.3  Serial0/0/1     00:02:29/00:01:43 v2     1 / S

```

```

R2# show ip pim neighbor
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      S - State Refresh Capable
Neighbor      Interface      Uptime/Expires    Ver    DR
Address
172.16.102.1  Serial0/0/0     00:03:16/00:01:25 v2     1 / S

```

```
172.16.203.3      Serial0/0/1      00:03:18/00:01:24 v2      1 / S
```

```
R3# show ip pim neighbor
```

```
PIM Neighbor Table
```

```
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,  
      S - State Refresh Capable
```

Neighbor Address	Interface	Uptime/Expires	Ver	DR Prio/Mode
172.16.13.1	FastEthernet0/0	00:03:00/00:01:42	v2	1 / S
172.16.103.1	Serial0/0/0	00:03:31/00:01:41	v2	1 / S
172.16.203.2	Serial0/0/1	00:03:03/00:01:38	v2	1 / S

Consider the adjacency between R1 and R3 on VLAN 13. If more than one multicast router exists on a multi-access VLAN, is it necessary for both devices to query for IGMP subscriptions on the VLAN?

No. Only one IGMP querier is necessary at a time. IGMP will elect a single querier for the subnet.

Recall the idea of a designated router (DR) in OSPF or a designated intermediate system (DIS) in IS-IS. These link-state protocols allow Layer 3 devices running those protocols to become adjacent only with the master device for that multi-access Layer 2 medium. This behavior decreases network control traffic and provides an authoritative source for routing information on that network segment.

A similar situation exists for IGMP control traffic on multi-access media, such as Ethernet. Rather than have each multicast router on a subnet running its own IGMP queries, PIM-DM elects one router to handle the IGMP querying for the entire network segment. PIM elects a DR for that subnet by selecting the PIM-DM router with the highest IP address.

From a practical perspective, do you need a DR on a point-to-point medium? Explain.

Point-to-point media do not require a DR because control traffic that needs to be sent from one router to another will be increased, not alleviated, by creating a DR.

Display detailed information about PIM-enabled interfaces with the **show ip pim interface detail** command on R1.

```
R1# show ip pim interface detail
Loopback1 is up, line protocol is up
  Internet address is 192.168.1.1/24
  Multicast switching: fast
  Multicast packets in/out: 919/0
  Multicast TTL threshold: 0
  PIM: enabled
```

```

PIM version: 2, mode: dense
PIM DR: 192.168.1.1 (this system)
PIM neighbor count: 0
  PIM Hello/Query interval: 30 seconds
  PIM Hello packets in/out: 77/78
  PIM State-Refresh processing: enabled
  PIM State-Refresh origination: disabled
  PIM NBMA mode: disabled
  PIM ATM multipoint signalling: disabled
  PIM domain border: disabled
Multicast Tagswitching: disabled
Serial0/0/0 is up, line protocol is up
  Internet address is 172.16.102.1/29
  Multicast switching: fast
  Multicast packets in/out: 917/0
  Multicast TTL threshold: 0
PIM: enabled
  PIM version: 2, mode: dense
  PIM DR: 0.0.0.0
  PIM neighbor count: 1
  PIM Hello/Query interval: 30 seconds
  PIM Hello packets in/out: 77/78
  PIM State-Refresh processing: enabled
  PIM State-Refresh origination: disabled
  PIM NBMA mode: disabled
  PIM ATM multipoint signalling: disabled
  PIM domain border: disabled
Multicast Tagswitching: disabled
Serial0/0/1 is up, line protocol is up
  Internet address is 172.16.103.1/29
  Multicast switching: fast
  Multicast packets in/out: 920/0
  Multicast TTL threshold: 0
PIM: enabled
  PIM version: 2, mode: dense
  PIM DR: 0.0.0.0
  PIM neighbor count: 1
  PIM Hello/Query interval: 30 seconds
  PIM Hello packets in/out: 77/78
  PIM State-Refresh processing: enabled
  PIM State-Refresh origination: disabled
  PIM NBMA mode: disabled
  PIM ATM multipoint signalling: disabled
  PIM domain border: disabled
Multicast Tagswitching: disabled
FastEthernet0/0 is up, line protocol is up
  Internet address is 172.16.13.1/24
  Multicast switching: fast
  Multicast packets in/out: 918/0
  Multicast TTL threshold: 0
PIM: enabled
  PIM version: 2, mode: dense
  PIM DR: 172.16.13.3
  PIM neighbor count: 1
  PIM Hello/Query interval: 30 seconds
  PIM Hello packets in/out: 76/77
  PIM State-Refresh processing: enabled
  PIM State-Refresh origination: disabled
  PIM NBMA mode: disabled
  PIM ATM multipoint signalling: disabled
  PIM domain border: disabled
Multicast Tagswitching: disabled

```

Notice that the two serial interfaces use the default DR address 0.0.0.0 as the DR for the interface. Because a multicast packet is received by either 0 or 1 remote routers on a serial segment, PIM does not need to set up a complex neighbor relationship.

Step 5: Verify Multicast Routing Operation

On each router, use the **mrinfo** command to view information about the connected multicast-enabled routers.

```
R1# mrinfo
172.16.13.1 [version 12.4] [flags: PMA]:
  192.168.1.1 -> 0.0.0.0 [1/0/pim/querier/leaf]
  172.16.13.1 -> 172.16.13.3 [1/0/pim]
  172.16.102.1 -> 172.16.102.2 [1/0/pim]
  172.16.103.1 -> 172.16.103.3 [1/0/pim]
```

```
R2# mrinfo
172.16.20.2 [version 12.4] [flags: PMA]:
  192.168.2.1 -> 0.0.0.0 [1/0/pim/querier/leaf]
  172.16.20.2 -> 0.0.0.0 [1/0/pim/querier/leaf]
  172.16.102.2 -> 172.16.102.1 [1/0/pim]
  172.16.203.2 -> 172.16.203.3 [1/0/pim]
```

```
R3# mrinfo
172.16.13.3 [version 12.4] [flags: PMA]:
  192.168.3.1 -> 0.0.0.0 [1/0/pim/querier/leaf]
  172.16.13.3 -> 172.16.13.1 [1/0/pim/querier]
  172.16.103.3 -> 172.16.103.1 [1/0/pim]
  172.16.203.3 -> 172.16.203.2 [1/0/pim]
```

Each router realizes that the loopback interfaces are topological leaves in which PIM will never establish an adjacency with any other routers. These routers also record the neighboring multicast router addresses and the multicast routing protocols they utilize.

Use the **show ip multicast interface** command to display statistics about the multicast traffic passing through the router. You should receive similar output on each router.

```
R1# show ip multicast interface
Loopback1 is up, line protocol is up
  Internet address is 192.168.1.1/24
  Multicast routing: enabled
  Multicast switching: fast
  Multicast packets in/out: 512/0
  Multicast TTL threshold: 0
  Multicast Tagswitching: disabled
FastEthernet0/0 is up, line protocol is up
  Internet address is 172.16.13.1/24
  Multicast routing: enabled
  Multicast switching: fast
  Multicast packets in/out: 524/6
  Multicast TTL threshold: 0
  Multicast Tagswitching: disabled
Serial0/0/0 is up, line protocol is up
  Internet address is 172.16.102.1/29
```

```

Multicast routing: enabled
Multicast switching: fast
Multicast packets in/out: 519/0
Multicast TTL threshold: 0
Multicast Tagswitching: disabled
Serial0/0/1 is up, line protocol is up
Internet address is 172.16.103.1/29
Multicast routing: enabled
Multicast switching: fast
Multicast packets in/out: 516/6
Multicast TTL threshold: 0
Multicast Tagswitching: disabled

```

Based on the above output and your knowledge of PIM-DM and this topology, which interfaces appear to be forwarding multicasts to 232.32.32.32 on R1?

Based on the above output, R1's FastEthernet 0/0 and Serial 0/0/1 interfaces are forwarding multicast traffic because they have non-zero outbound multicast packet counts.

Generate a stream of multicast data to the group by issuing an extended ping from SW1 with a repeat count of 100.

```

SW1# ping
Protocol [ip]:
Target IP address: 232.32.32.32
Repeat count [1]: 100
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 232.32.32.32, timeout is 2 seconds:

Reply to request 0 from 172.16.20.2, 8 ms
Reply to request 0 from 172.16.102.1, 36 ms
Reply to request 0 from 172.16.203.3, 36 ms
Reply to request 1 from 172.16.20.2, 4 ms
Reply to request 1 from 172.16.102.1, 28 ms
Reply to request 1 from 172.16.203.3, 28 ms
Reply to request 2 from 172.16.20.2, 8 ms
Reply to request 2 from 172.16.102.1, 32 ms
Reply to request 2 from 172.16.203.3, 32 ms
...

```

On each of the routers, you should see that PIM and IGMP have communicated to install the 232.32.32.32 multicast group in the multicast routing table. Verify this with the **show ip mroute** command on each router.

```

R1# show ip mroute
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,

```

Z - Multicast Tunnel, z - MDT-data group sender,
Y - Joined MDT-data group, y - Sending to MDT-data group
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

```
(* , 232.32.32.32), 02:34:00/stopped, RP 0.0.0.0, flags: DCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  Serial0/0/0, Forward/Dense, 00:31:23/00:00:00
  FastEthernet0/0, Forward/Dense, 02:33:31/00:00:00
  Serial0/0/1, Forward/Dense, 02:33:31/00:00:00
  Loopback1, Forward/Dense, 02:34:00/00:00:00
```

```
(172.16.20.4, 232.32.32.32), 00:00:09/00:03:00, flags: LT
Incoming interface: Serial0/0/0, RPF nbr 172.16.102.2
Outgoing interface list:
  Loopback1, Forward/Dense, 00:00:09/00:00:00
  Serial0/0/1, Prune/Dense, 00:00:09/00:02:52
  FastEthernet0/0, Prune/Dense, 00:00:09/00:02:49
```

```
(* , 224.0.1.40), 02:34:01/00:02:43, RP 0.0.0.0, flags: DCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  Serial0/0/0, Forward/Dense, 00:31:24/00:00:00
  FastEthernet0/0, Forward/Dense, 02:33:33/00:00:00
  Serial0/0/1, Forward/Dense, 02:33:33/00:00:00
  Loopback1, Forward/Dense, 02:34:02/00:00:00
```

R2# show ip mroute

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
L - Local, P - Pruned, R - RP-bit set, F - Register flag,
T - SPT-bit set, J - Join SPT, M - MSDP created entry,
X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
U - URD, I - Received Source Specific Host Report,
Z - Multicast Tunnel, z - MDT-data group sender,
Y - Joined MDT-data group, y - Sending to MDT-data group
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

```
(* , 232.32.32.32), 00:32:01/stopped, RP 0.0.0.0, flags: DCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  Serial0/0/0, Forward/Dense, 00:31:33/00:00:00
  Serial0/0/1, Forward/Dense, 00:31:33/00:00:00
  Loopback2, Forward/Dense, 00:32:01/00:00:00
```

```
(172.16.20.4, 232.32.32.32), 00:00:48/00:02:58, flags: LT
Incoming interface: FastEthernet0/0, RPF nbr 0.0.0.0
Outgoing interface list:
  Loopback2, Forward/Dense, 00:00:50/00:00:00
  Serial0/0/1, Forward/Dense, 00:00:50/00:00:00
  Serial0/0/0, Prune/Dense, 00:00:50/00:00:00
```

```
(* , 224.0.1.40), 00:32:03/00:02:47, RP 0.0.0.0, flags: DCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  Serial0/0/0, Forward/Dense, 00:31:34/00:00:00
  Serial0/0/1, Forward/Dense, 00:31:34/00:00:00
  Loopback2, Forward/Dense, 00:32:23/00:00:00
```

```

R3# show ip mroute
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
      L - Local, P - Pruned, R - RP-bit set, F - Register flag,
      T - SPT-bit set, J - Join SPT, M - MSDP created entry,
      X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
      U - URD, I - Received Source Specific Host Report,
      Z - Multicast Tunnel, z - MDT-data group sender,
      Y - Joined MDT-data group, y - Sending to MDT-data group
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 232.32.32.32), 02:34:45/stopped, RP 0.0.0.0, flags: DCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  Serial0/0/1, Forward/Dense, 00:32:05/00:00:00
  FastEthernet0/0, Forward/Dense, 02:34:13/00:00:00
  Serial0/0/0, Forward/Dense, 02:34:43/00:00:00
  Loopback3, Forward/Dense, 02:34:45/00:00:00

(172.16.20.4, 232.32.32.32), 00:00:52/00:02:59, flags: LT
Incoming interface: Serial0/0/1, RPF nbr 172.16.203.2
Outgoing interface list:
  Loopback3, Forward/Dense, 00:00:52/00:00:00
  Serial0/0/0, Prune/Dense, 00:00:51/00:02:11, A
  FastEthernet0/0, Forward/Dense, 00:00:48/00:02:11, A

(*, 224.0.1.40), 02:34:46/00:02:58, RP 0.0.0.0, flags: DCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  Serial0/0/1, Forward/Dense, 00:32:06/00:00:00
  FastEthernet0/0, Forward/Dense, 02:34:15/00:00:00
  Serial0/0/0, Forward/Dense, 02:34:46/00:00:00
  Loopback3, Forward/Dense, 02:34:47/00:00:00

```

The two timers shown on the first line of the (S, G) entry on each router indicate the time since the first multicast to that group and the time when the entry expires. All the expiration timers for (172.16.20.4, 232.32.32.32) state display as roughly three minutes while SW1 sends the multicast pings. Thus, each time the router receives a multicast packet matching the (S, G) entry (172.16.20.4, 232.32.32.32), IGMP resets the expiration timer to three minutes.

By default, IGMP sends a general query every 60 seconds on each PIM interface. A general query requests any devices running IGMP on that subnet to report any groups to which they subscribe to the querying router. If no membership reports are heard for a particular group on that interface over three times the query interval, the multicast router declares that there is no active member of the group on the interface. The router stops sending multicasts to that group on that egress interface. If a router does not receive multicasts from an (S, G) pair for three minutes, the multicast routing table deletes the (S, G) entry.

The IP multicast routing table also indicates whether the PIM interfaces negotiated to flood multicast data to that group out an interface or pruned (S, G) multicast traffic from being sent out that interface.

Based on the IP multicast routing tables shown above, which PIM interfaces on each router are forwarding traffic from 172.16.20.4 to 232.32.32.32?

On R1, the Loopback1 interface is forwarding traffic to 232.32.32.32.

On R2, the Loopback2 and Serial0/0/1 interfaces are forwarding traffic to 232.32.32.32.

On R3, the Loopback3 and FastEthernet0/0 interfaces are forwarding traffic to 232.32.32.32.

Step 6: Verify PIM-DM Flood-and-Prune Behavior

To view the flood-and-prune behavior of PIM-DM, wait for the multicast stream from 172.16.20.4 to complete and for the (S, G) state in the multicast routing tables to expire. Then issue the **debug ip igmp** and **debug ip pim** commands on all routers.

PIM relies on the unicast IP routing table to construct shortest-path trees from the sources to the multicast subscribers. PIM neighbors send control messages to determine which neighbor is closer to the source in terms of the unicast routing information on each neighbor. PIM neighbors on the subnet elect a particular router as the forwarder for that (S, G) pair using Assert messages. Each Assert message carries the best administrative distance and metric that the advertising router has to the source. The PIM router with the best administrative distance and metric is elected as the forwarder for that (S, G) or (*, G) entry. The forwarder then prunes that (S, G) pair from being forwarded by any other routers on the subnet.

Do not confuse the DR with the forwarder. Although both are elected on multi-access networks, the role of DR does not include the responsibility to forward multicast traffic. As defined by IGMPv1, a DR is elected based on highest IP address to control the IGMP querying. Thus, a DR exists to identify which receivers exist on a subnet by polling for listeners to any group. There can only be one DR on a subnet at a time.

In contrast, every multi-access subnet elects forwarders individually for each (S, G) and (*, G) pair. The forwarder is elected based on best administrative distance and metric to the source. The forwarder is the metrically closest router on the subnet to the source.

Display the unicast routing table entry for 172.16.20.4 on each router.

```
R1# show ip route 172.16.20.4
Routing entry for 172.16.20.0/24
  Known via "eigrp 1", distance 90, metric 20517120, type internal
  Redistributing via eigrp 1
```

```

Last update from 172.16.13.3 on FastEthernet0/0, 00:42:57 ago
Routing Descriptor Blocks:
* 172.16.13.3, from 172.16.13.3, 00:42:57 ago, via FastEthernet0/0
  Route metric is 20517120, traffic share count is 1
  Total delay is 20200 microseconds, minimum bandwidth is 128 Kbit
  Reliability 255/255, minimum MTU 1500 bytes
  Loading 1/255, Hops 2

R2# show ip route 172.16.20.4
Routing entry for 172.16.20.0/24
  Known via "connected", distance 0, metric 0 (connected, via interface)
  Redistributing via eigrp 1
  Routing Descriptor Blocks:
  * directly connected, via FastEthernet0/0
    Route metric is 0, traffic share count is 1

R3# show ip route 172.16.20.4
Routing entry for 172.16.20.0/24
  Known via "eigrp 1", distance 90, metric 20514560, type internal
  Redistributing via eigrp 1
  Last update from 172.16.203.2 on Serial0/0/1, 00:43:31 ago
  Routing Descriptor Blocks:
  * 172.16.203.2, from 172.16.203.2, 00:43:31 ago, via Serial0/0/1
    Route metric is 20514560, traffic share count is 1
    Total delay is 20100 microseconds, minimum bandwidth is 128 Kbit
    Reliability 255/255, minimum MTU 1500 bytes
    Loading 1/255, Hops 1

```

Begin the same extended ping on SW1 to the multicast group. Examine the debugging messages shown on each router.

```

R1# debug ip pim
*Nov 6 00:43:28.731: PIM(0): Send v2 Assert on Serial0/0/0 for 232.32.32.32,
source 172.16.20.4, metric [90/20517120]
*Nov 6 00:43:28.735: PIM(0): Assert metric to source 172.16.20.4 is
[90/20517120]
*Nov 6 00:43:28.735: PIM(0): We win, our metric [90/20517120]
*Nov 6 00:43:28.735: PIM(0): Prune Serial0/0/0/232.32.32.32 from
(172.16.20.4/32, 232.32.32.32)
*Nov 6 00:43:28.735: PIM(0): Pruning immediately Serial0/0/0 (p2p)
*Nov 6 00:43:28.743: PIM(0): Send v2 Assert on Serial0/0/1 for 232.32.32.32,
source 172.16.20.4, metric [90/20517120]
*Nov 6 00:43:28.743: PIM(0): Assert metric to source 172.16.20.4 is
[90/20517120]
*Nov 6 00:43:28.743: PIM(0): We win, our metric [90/20517120]
*Nov 6 00:43:28.743: PIM(0): Prune Serial0/0/1/232.32.32.32 from
(172.16.20.4/32, 232.32.32.32)
*Nov 6 00:43:28.743: PIM(0): Pruning immediately Serial0/0/1 (p2p)

... continues below ...

```

Because R1 has not yet received any PIMv2 assert messages from its neighbors on either Serial0/0/0 or Serial0/0/1, it has elected itself as the forwarder for (172.16.20.4/32, 232.32.32.32) on both interfaces and pruned other traffic flows.

On a point-to-point link, a PIM-DM router must assert itself as being the forwarder on the subnet for the group, unless another router sends an assert message with a lower metric to the source. This behavior allows PIM-DM to

succeed in the simple case of a single multicast router on a subnet. In this case, the router cannot wait for other multicast routers to respond to the assert message before flooding multicast data; it must simply begin sending data until another router with a lower metric to the source prunes it.

Notice that the metrics used for route calculation are the EIGRP and composite metrics for the unicast IP routes to those source networks from the unicast routing table.

... continued from above ...

```
*Nov 6 00:43:28.751: PIM(0): Received v2 Assert on Serial0/0/1 from
172.16.103.3
*Nov 6 00:43:28.751: PIM(0): Assert metric to source 172.16.20.4 is
[90/20514560]
*Nov 6 00:43:28.751: PIM(0): We lose, our metric [90/20517120]
*Nov 6 00:43:28.751: PIM(0): (172.16.20.4/32, 232.32.32.32) oif Serial0/0/1
in Prune state
*Nov 6 00:43:28.751: PIM(0): Received v2 Assert on Serial0/0/0 from
172.16.102.2
*Nov 6 00:43:28.751: PIM(0): Assert metric to source 172.16.20.4 is [0/0]
*Nov 6 00:43:28.751: PIM(0): We lose, our metric [90/20517120]
*Nov 6 00:43:28.751: PIM(0): (172.16.20.4/32, 232.32.32.32) oif Serial0/0/0
in Prune state
*Nov 6 00:43:29.663: PIM(0): Received v2 Assert on Serial0/0/0 from
172.16.102.2
*Nov 6 00:43:29.663: PIM(0): Assert metric to source 172.16.20.4 is [0/0]
*Nov 6 00:43:29.663: PIM(0): We lose, our metric [90/20517120]
*Nov 6 00:43:29.663: PIM(0): (172.16.20.4/32, 232.32.32.32) oif Serial0/0/0
in Prune state
*Nov 6 00:43:29.751: PIM(0): Received v2 Assert on Serial0/0/1 from
172.16.103.3
*Nov 6 00:43:29.751: PIM(0): Assert metric to source 172.16.20.4 is
[90/20514560]
*Nov 6 00:43:29.751: PIM(0): We lose, our metric [90/20517120]
*Nov 6 00:43:29.751: PIM(0): (172.16.20.4/32, 232.32.32.32) oif Serial0/0/1
in Prune state
```

PIM selects the router with the lowest administrative distance to be the designated forwarder for the subnet for the (S, G) pair. In the case of a tie, PIM prefers the router with the lowest metric.

Step 7: Explore the Multicast Routing Table

Verify the state you predicted at the end of Step 5 about forwarding and pruning interfaces against the following output of the **show ip mroute source_address group_address** command:

```
R1# show ip mroute 172.16.20.4 232.32.32.32
<output omitted>

(172.16.20.4, 232.32.32.32), 00:01:07/00:02:58, flags: LT
  Incoming interface: FastEthernet0/0, RPF nbr 172.16.13.3
  Outgoing interface list:
    Loopback1, Forward/Dense, 00:01:07/00:00:00
    Serial0/0/1, Prune/Dense, 00:01:07/00:01:55
    Serial0/0/0, Prune/Dense, 00:01:07/00:01:55
```

```
R2# show ip mroute 172.16.20.4 232.32.32.32
<output omitted>

(172.16.20.4, 232.32.32.32), 00:01:21/00:02:58, flags: LT
  Incoming interface: FastEthernet0/0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Loopback2, Forward/Dense, 00:01:21/00:00:00
    Serial0/0/0, Prune/Dense, 00:01:20/00:01:42, A
    Serial0/0/1, Forward/Dense, 00:01:21/00:00:00
```

```
R3# show ip mroute 172.16.20.4 232.32.32.32
<output omitted>

(172.16.20.4, 232.32.32.32), 00:01:22/00:02:59, flags: LT
  Incoming interface: Serial0/0/1, RPF nbr 172.16.203.2
  Outgoing interface list:
    Loopback3, Forward/Dense, 00:01:22/00:00:00
    Serial0/0/0, Prune/Dense, 00:01:22/00:01:40, A
    FastEthernet0/0, Forward/Dense, 00:01:22/00:00:00
```

How does PIM decide which incoming interface to use for each group?

PIM does an RPF (reverse-path forwarding) check in the unicast routing table for 172.16.20.4 to find the “upstream” incoming interface.

Because PIM does not use its own topological algorithm to locate multicast sources, it must have a way of determining which interface faces the upstream neighbor of the tree stemming from the multicast source. PIM uses the reverse-path forwarding (RPF) check to find the interface closest to the source in terms of destination-based unicast routing.

Cisco IOS allows you to run RPF checks for specific sources with the **show ip rpf source_address** command. Use this command on R1 to find the incoming interface for the (172.16.20.4, 232.32.32.32) pair.

```
R1# show ip rpf 172.16.20.4
RPF information for ? (172.16.20.4)
  RPF interface: FastEthernet0/0
  RPF neighbor: ? (172.16.13.3)
  RPF route/mask: 172.16.20.0/24
  RPF type: unicast (eigrp 1)
  RPF recursion count: 0
  Doing distance-preferred lookups across tables
```

Although the multicast routing table includes RPF information, the **show ip rpf** command can be useful when debugging hidden multicast issues.

Use this output from R1 to answer the following questions:

```
R1# show ip mroute 172.16.20.4 232.32.32.32
<output omitted>

(172.16.20.4, 232.32.32.32), 00:01:07/00:02:58, flags: LT
  Incoming interface: FastEthernet0/0, RPF nbr 172.16.13.3
```

```
Outgoing interface list:
  Loopback1, Forward/Dense, 00:01:07/00:00:00
  Serial0/0/1, Prune/Dense, 00:01:07/00:01:55
  Serial0/0/0, Prune/Dense, 00:01:07/00:01:55
```

What is the incoming interface for the (S, G) pair (172.16.20.4, 232.32.32.32)?

The incoming interface is Fast Ethernet 0/0.

How does Cisco IOS assign this interface as the incoming interface?

It does an RPF lookup for 172.16.20.4, and assigns the outgoing interface in the unicast routing table to be the incoming interface for the multicast routing table.

Which neighboring router appears to R1 as being the next hop upstream toward the multicast source 172.16.20.4?

R3, because the RPF neighbor is 172.16.13.3, the IP address of R3's Fast Ethernet interface.

Challenge

If your simulation of the (S, G) pair (172.16.20.4, 232.32.32.32) correctly models the CEO's presentation to be shown later this week, do you think the presentation will be available on the loopback interfaces at all three sites?

Yes. Because PIM is now running between these routers, multicasts to the 232.32.32.32 group from 172.16.20.4 will be available on each of those loopback interfaces.

Why does Cisco not recommend using PIM-DM in production networks?

PIM-DM uses the flood-and-prune behavior to send traffic down the source tree to every subnet in the network. Flood and prune behavior effectively acts like a broadcast through the network that will be sent even to subnets not interested in the traffic. PIM-DM also requires that every multicast router in the network have full knowledge of all multicast sources currently sending traffic. This is a significant amount of memory overhead that is not necessary when you compare to PIM-SM, which performs all of the requirements of multicast without the same bandwidth and memory overhead.

Appendix A: TCL Script Output – Unicast

```
R1# tclsh
R1(tcl)#foreach address {
+>(tcl)#192.168.1.1
+>(tcl)#172.16.13.1
+>(tcl)#172.16.102.1
+>(tcl)#172.16.103.1
+>(tcl)#192.168.2.1
+>(tcl)#172.16.20.2
+>(tcl)#172.16.102.2
+>(tcl)#172.16.203.2
+>(tcl)#192.168.3.1
+>(tcl)#172.16.13.3
+>(tcl)#172.16.103.3
+>(tcl)#172.16.203.3
+>(tcl)#172.16.20.4
+>(tcl)#} { ping $address }

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.102.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.103.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.20.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.102.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.203.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.103.3, timeout is 2 seconds:
```

```
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.203.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.20.4, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
R1(tcl)# tclquit
```

```
R2# tclsh
R2(tcl)#foreach address {
+>(tcl)#192.168.1.1
+>(tcl)#172.16.13.1
+>(tcl)#172.16.102.1
+>(tcl)#172.16.103.1
+>(tcl)#192.168.2.1
+>(tcl)#172.16.20.2
+>(tcl)#172.16.102.2
+>(tcl)#172.16.203.2
+>(tcl)#192.168.3.1
+>(tcl)#172.16.13.3
+>(tcl)#172.16.103.3
+>(tcl)#172.16.203.3
+>(tcl)#172.16.20.4
+>(tcl)#} { ping $address }
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.102.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.103.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.20.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.102.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/64 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.203.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/68 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:
```

```

!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.103.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/42/44 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.203.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.20.4, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
R2(tcl)# tclquit

```

```

R3# tclsh
R3(tcl)#foreach address {
+>(tcl)#192.168.1.1
+>(tcl)#172.16.13.1
+>(tcl)#172.16.102.1
+>(tcl)#172.16.103.1
+>(tcl)#192.168.2.1
+>(tcl)#172.16.20.2
+>(tcl)#172.16.102.2
+>(tcl)#172.16.203.2
+>(tcl)#192.168.3.1
+>(tcl)#172.16.13.3
+>(tcl)#172.16.103.3
+>(tcl)#172.16.203.3
+>(tcl)#172.16.20.4
+>(tcl)#} { ping $address }

```

```

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.102.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.103.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.20.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.102.2, timeout is 2 seconds:

```

```

!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.203.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.103.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.203.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/64 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.20.4, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
R3(tcl)# tclquit

```

Final Configurations

```

R1# show run
!
hostname R1
!
interface Loopback1
 ip address 192.168.1.1 255.255.255.0
 ip pim dense-mode
 ip igmp join-group 232.32.32.32
!
interface FastEthernet0/0
 ip address 172.16.13.1 255.255.255.0
 ip pim dense-mode
 no shutdown
!
interface Serial0/0/0
 bandwidth 64
 ip address 172.16.102.1 255.255.255.248
 ip pim dense-mode
 clock rate 64000
 no shutdown
!
interface Serial0/0/1
 bandwidth 64
 ip address 172.16.103.1 255.255.255.248
 ip pim dense-mode
 no shutdown
!
router eigrp 1
 network 172.16.0.0
 network 192.168.0.0 0.0.255.255
 auto-summary
!
end

```

```

R2# show run
!
hostname R2
!
interface Loopback2
 ip address 192.168.2.1 255.255.255.0
 ip pim dense-mode
 ip igmp join-group 232.32.32.32
!
interface FastEthernet0/0
 ip address 172.16.20.2 255.255.255.0
 ip pim dense-mode
 no shutdown
!
interface Serial0/0/0
 bandwidth 64
 ip address 172.16.102.2 255.255.255.248
 ip pim dense-mode
 no shutdown
!
interface Serial0/0/1
 bandwidth 128
 ip address 172.16.203.2 255.255.255.248
 ip pim dense-mode
 clock rate 128000
 no shutdown
!
router eigrp 1
 network 172.16.0.0
 network 192.168.0.0 0.0.255.255
 auto-summary
@
end

```

```

R3# show run
!
hostname R3
!
interface Loopback3
 ip address 192.168.3.1 255.255.255.0
 ip pim dense-mode
 ip igmp join-group 232.32.32.32
!
interface FastEthernet0/0
 ip address 172.16.13.3 255.255.255.0
 ip pim dense-mode
 no shutdown
!
interface Serial0/0/0
 bandwidth 64
 ip address 172.16.103.3 255.255.255.248
 ip pim dense-mode
 clock rate 64000
 no shutdown
!
interface Serial0/0/1
 bandwidth 128
 ip address 172.16.203.3 255.255.255.248
 ip pim dense-mode
 no shutdown
!
router eigrp 1

```

```
network 172.16.0.0
network 192.168.0.0 0.0.255.255
auto-summary
!
end
```

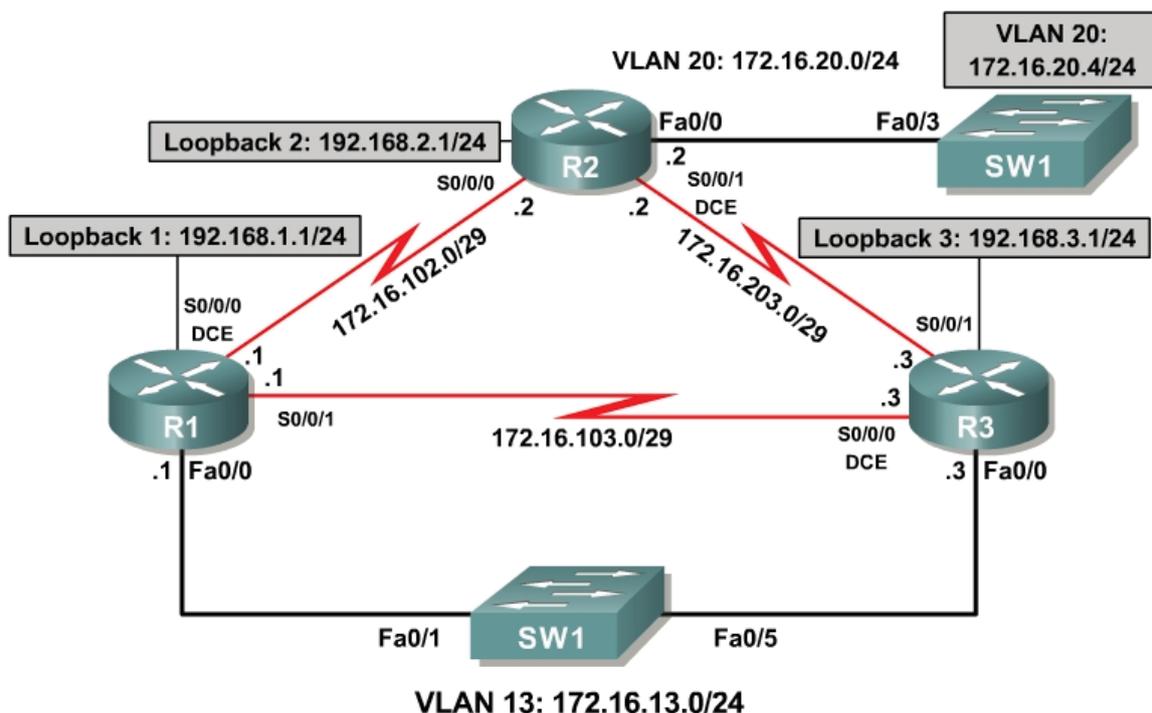
```
SW1# show run
!
hostname SW1
!
interface FastEthernet0/1
  switchport access vlan 13
  switchport mode access
!
interface FastEthernet0/3
  switchport access vlan 20
  switchport mode access
!
interface FastEthernet0/5
  switchport access vlan 13
  switchport mode access
!
interface Vlan20
  ip address 172.16.20.4 255.255.255.0
  no shutdown
!
ip default-gateway 172.16.20.2
!
end
```

Lab 7-3 Routing IP Multicast with PIM Sparse Mode

Learning Objectives

- Implement and verify PIM-SM operation and adjacencies
- Implement and verify the use of a static rendezvous point
- Observe the shared tree and source tree
- Debug the PIM-SM operation to discover SPT shared tree to shortest-path tree transition

Topology Diagram



Scenario

Your successful assistance in configuring IP multicast routing has allowed CrushASodaCan.com to become a community action organization with a wide base of interested citizens. Because of your help, the organization can now preview their TV commercials via the network before releasing them to the public. CrushASodaCan.com also projects high growth rates in terms of volunteer workers using their network. Unfortunately, the flooding and pruning with PIM-DM that you configured for them cannot handle the new demands being made on the network. So they have hired you again as a consultant to

solve this. You decide to implement PIM Sparse Mode (PIM-SM) to create a subscription-based multicast topology in CrushASodaCan.com's network.

Step 1: Load Initial Configurations

Begin with the final configurations from Lab 7.2.

```
R1:
!
hostname R1
!
interface Loopback1
 ip address 192.168.1.1 255.255.255.0
 ip pim dense-mode
 ip igmp join-group 232.32.32.32
!
interface FastEthernet0/0
 ip address 172.16.13.1 255.255.255.0
 ip pim dense-mode
 no shutdown
!
interface Serial0/0/0
 bandwidth 64
 ip address 172.16.102.1 255.255.255.248
 ip pim dense-mode
 clock rate 64000
 no shutdown
!
interface Serial0/0/1
 bandwidth 64
 ip address 172.16.103.1 255.255.255.248
 ip pim dense-mode
 no shutdown
!
router eigrp 1
 network 172.16.0.0
 network 192.168.0.0 0.0.255.255
 auto-summary
!
end

R2:
!
hostname R2
!
interface Loopback2
 ip address 192.168.2.1 255.255.255.0
 ip pim dense-mode
 ip igmp join-group 232.32.32.32
!
interface FastEthernet0/0
 ip address 172.16.20.2 255.255.255.0
 ip pim dense-mode
 no shutdown
!
interface Serial0/0/0
 bandwidth 64
 ip address 172.16.102.2 255.255.255.248
 ip pim dense-mode
 no shutdown
!
```

```

interface Serial0/0/1
  bandwidth 128
  ip address 172.16.203.2 255.255.255.248
  ip pim dense-mode
  clock rate 128000
  no shutdown
!
router eigrp 1
  network 172.16.0.0
  network 192.168.0.0 0.0.255.255
  auto-summary
!
end

```

R3:

```

!
hostname R3
!
interface Loopback3
  ip address 192.168.3.1 255.255.255.0
  ip pim dense-mode
  ip igmp join-group 232.32.32.32
!
interface FastEthernet0/0
  ip address 172.16.13.3 255.255.255.0
  ip pim dense-mode
  no shutdown
!
interface Serial0/0/0
  bandwidth 64
  ip address 172.16.103.3 255.255.255.248
  ip pim dense-mode
  clock rate 64000
  no shutdown
!
interface Serial0/0/1
  bandwidth 128
  ip address 172.16.203.3 255.255.255.248
  ip pim dense-mode
  no shutdown
!
router eigrp 1
  network 172.16.0.0
  network 192.168.0.0 0.0.255.255
  auto-summary
!
end

```

SW1# show run

```

!
hostname SW1
!
interface FastEthernet0/1
  switchport access vlan 13
  switchport mode access
!
interface FastEthernet0/3
  switchport access vlan 20
  switchport mode access
!
interface FastEthernet0/5
  switchport access vlan 13
  switchport mode access

```

```

!
interface Vlan20
 ip address 172.16.20.4 255.255.255.0
 no shutdown
!
ip default-gateway 172.16.20.2
!
end

```

Verify that each of the loopback interfaces subscribed to the multicast group using the **show ip igmp groups** command.

```

R1# show ip igmp groups
IGMP Connected Group Membership
Group Address      Interface          Uptime    Expires    Last Reporter
Group Accounted
232.32.32.32      Loopback1         00:05:45  00:01:59  192.168.1.1
224.0.1.40        Loopback1         00:05:45  00:01:59  192.168.1.1

```

```

R2# show ip igmp groups
IGMP Connected Group Membership
Group Address      Interface          Uptime    Expires    Last Reporter
Group Accounted
232.32.32.32      Loopback2         00:05:12  00:02:37  192.168.2.1
224.0.1.40        Loopback2         00:05:12  00:02:40  192.168.2.1

```

```

R3# show ip igmp groups
IGMP Connected Group Membership
Group Address      Interface          Uptime    Expires    Last Reporter
Group Accounted
232.32.32.32      Loopback3         00:05:27  00:02:23  192.168.3.1
224.0.1.40        Loopback3         00:05:27  00:02:21  192.168.3.1

```

Step 3: Implement PIM-SM

Planning is important when creating PIM-SM networks. Because traffic is not simply flooded out all PIM interfaces as it is in dense mode networks, you make sure to place the rendezvous point (RP) in a central location.

CrushASodaCan.com's management decided the RP should be placed on R1. This location allows you to observe the cutover from the shared tree to the source tree, because R2 and R3 have shorter paths to 172.16.20.4 than those through R1.

Before enabling PIM-SM on the interfaces, set the static RP address as R1's Loopback1 address using the **ip pim rp-address rp-address [access-list]** command. You can either assign a router to be the global RP for all multicast groups or have it be on a per-group basis using an access list. In this case, map R1 as the RP only for the 232.32.32.32 multicast group. This command must be used on all routers running PIM-SM.

```

R1# conf t
R1(config)# access-list 32 permit 232.32.32.32
R1(config)# ip pim rp-address 192.168.1.1 32

R2# conf t
R2(config)# access-list 32 permit 232.32.32.32

```

```
R2(config)# ip pim rp-address 192.168.1.1

R3# conf t
R3(config)# access-list 32 permit 232.32.32.32
R3(config)# ip pim rp-address 192.168.1.1
```

What is the purpose of the RP in PIM-SM?

RPs provide authoritative joining points between the multicast sources and group members in a multicast network. RPs create a common point for multicast routers to root shared trees.

Given the fact that an RP can be set for specific groups, can you have more than one static RP in a multicast network? Explain.

Yes. You may have numerous static RPs in a network. Each group could potentially have a unique RP.

PIM-SM has more redundant ways of configuring RPs and mapping agents. Notice in this example that each multicast router has a static RP address pointing to a specific router on the network and a static mapping agent of itself. A mapping agent is simply a device that provides a mapping of a group to an RP. You have statically the group-to-RP mapping on every router, so each router acts as its own mapping agent.

In PIM-DM, all (S, G) entries are present in *every* IP multicast routing table on the multicast network while the source is broadcasting. If you were using many groups or many sources for each group, the size of the multicast routing table would dramatically increase from the one in Lab 7.2.

PIM-SM rendezvous points introduce multicast sources by acting as the network receptionist that knows the location of all multicast sources. PIM-SM creates a shared distribution tree represented by (*, G) in each multicast routing table. This shared tree is calculated using the RPF upstream interface to the RP for that group. Thus, the shared tree is essentially a shortest-path tree to the RP address.

PIM-SM reduces the amount of multicast state in the network by using (*, G) shared tree entries on routers through which no sources are sending traffic for a given group G. When a node in the topology subscribes to a multicast group, that node's upstream multicast router registers with the RP and forwards traffic down the shared tree to the receiver until it discovers a shorter path to the source in its unicast routing table. At that point, the multicast router prunes the stream from the shared tree and begins receiving the stream via its shortest unicast path to the source represented by its (S, G) entry.

PIM-SM is designed for sparsely populated multicast networks in which it is not necessary to flood multicast traffic to every subnet. Sparse mode is a general-purpose multicast mode that should be used in most circumstances.

As you did in Lab 7.2, issue a repeated ping from SW1 to generate the (S, G) state in PIM-DM before you apply PIM-SM to the interfaces.

```
SW1# ping
Protocol [ip]:
Target IP address: 232.32.32.32
Repeat count [1]: 100
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 232.32.32.32, timeout is 2 seconds:

Reply to request 0 from 172.16.20.2, 8 ms
Reply to request 0 from 172.16.102.1, 36 ms
Reply to request 0 from 172.16.203.3, 36 ms
Reply to request 1 from 172.16.20.2, 4 ms
Reply to request 1 from 172.16.102.1, 28 ms
Reply to request 1 from 172.16.203.3, 28 ms
Reply to request 2 from 172.16.20.2, 8 ms
Reply to request 2 from 172.16.102.1, 32 ms
Reply to request 2 from 172.16.203.3, 32 ms
```

Display the multicast routing table on each router with the **show ip mroute** command.

```
R1# show ip mroute
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 232.32.32.32), 01:42:56/00:03:04, RP 192.168.1.1, flags: SJCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    FastEthernet0/0, Forward/Dense, 01:42:26/00:00:00
    Serial0/0/1, Forward/Dense, 01:42:27/00:00:00
    Serial0/0/0, Forward/Dense, 01:42:28/00:00:00
    Loopback1, Forward/Dense, 01:42:56/00:00:00

(172.16.20.4, 232.32.32.32), 00:01:20/00:02:59, flags: LT
  Incoming interface: FastEthernet0/0, RPF nbr 172.16.13.3
  Outgoing interface list:
    Loopback1, Forward/Dense, 00:01:20/00:00:00
    Serial0/0/0, Prune/Dense, 00:01:20/00:01:39
    Serial0/0/1, Prune/Dense, 00:01:20/00:01:39

(*, 224.0.1.40), 00:57:27/00:02:49, RP 0.0.0.0, flags: DCL
```

```
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  Serial0/0/1, Forward/Dense, 00:57:27/00:00:00
  Serial0/0/0, Forward/Dense, 00:57:35/00:00:00
  FastEthernet0/0, Forward/Dense, 00:57:35/00:00:00
  Loopback1, Forward/Dense, 00:57:35/00:00:00
```

R2# show ip mroute

IP Multicast Routing Table

```
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
```

```
(* , 232.32.32.32), 01:43:06/stopped, RP 192.168.1.1, flags: SJCLF
  Incoming interface: Serial0/0/1, RPF nbr 172.16.203.3
  Outgoing interface list:
    Serial0/0/0, Forward/Dense, 01:43:02/00:00:00
    Loopback2, Forward/Dense, 01:43:06/00:00:00
```

```
(172.16.20.4, 232.32.32.32), 00:01:24/00:02:59, flags: LFT
  Incoming interface: FastEthernet0/0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0/0/1, Forward/Dense, 00:01:26/00:00:00
    Loopback2, Forward/Dense, 00:01:26/00:00:00
    Serial0/0/0, Forward/Dense, 00:01:26/00:00:00, A
```

```
(* , 224.0.1.40), 00:55:42/00:02:41, RP 0.0.0.0, flags: DCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0/0/1, Forward/Dense, 00:55:42/00:00:00
    Serial0/0/0, Forward/Dense, 00:55:42/00:00:00
    Loopback2, Forward/Dense, 00:55:42/00:00:00
```

R3# show ip mroute

IP Multicast Routing Table

```
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
```

```
(* , 232.32.32.32), 01:43:06/00:03:26, RP 192.168.1.1, flags: SJCL
  Incoming interface: FastEthernet0/0, RPF nbr 172.16.13.1
  Outgoing interface list:
    Serial0/0/1, Forward/Dense, 01:42:37/00:00:00
    Serial0/0/0, Forward/Dense, 01:43:04/00:00:00
    Loopback3, Forward/Dense, 01:43:06/00:00:00
```

```
(172.16.20.4, 232.32.32.32), 00:01:26/00:02:58, flags: LT
  Incoming interface: Serial0/0/1, RPF nbr 172.16.203.2
  Outgoing interface list:
```

```

FastEthernet0/0, Forward/Dense, 00:01:28/00:00:00, A
Loopback3, Forward/Dense, 00:01:28/00:00:00
Serial0/0/0, Forward/Dense, 00:01:28/00:00:00, A

(*, 224.0.1.40), 01:43:07/00:02:30, RP 0.0.0.0, flags: DCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
FastEthernet0/0, Forward/Dense, 01:42:35/00:00:00
Serial0/0/1, Forward/Dense, 01:42:38/00:00:00
Serial0/0/0, Forward/Dense, 01:43:20/00:00:00
Loopback3, Forward/Dense, 01:43:21/00:00:00

```

Notice that R1 has 0.0.0.0 as the RPF neighbor of the (*, G) entry, indicating that it is the RP for the 232.32.32.32 multicast group. R2 and R3 listen to multicast traffic on the shared tree coming from their RPF neighbor for that (*, G) entry in their multicast routing tables.

Enable PIM-SM on all interfaces using the **ip pim sparse-mode** command in interface configuration mode.

```

R1(config)# interface loopback 1
R1(config-if)# ip pim sparse-mode
R1(config-if)# interface fastethernet 0/0
R1(config-if)# ip pim sparse-mode
R1(config-if)# interface serial 0/0/0
R1(config-if)# ip pim sparse-mode
R1(config-if)# interface serial 0/0/1
R1(config-if)# ip pim sparse-mode

R2(config)# interface loopback 2
R2(config-if)# ip pim sparse-mode
R2(config-if)# interface fastethernet 0/0
R2(config-if)# ip pim sparse-mode
R2(config-if)# interface serial 0/0/0
R2(config-if)# ip pim sparse-mode
R2(config-if)# interface serial 0/0/1
R2(config-if)# ip pim sparse-mode

R3(config)# interface loopback 1
R3(config-if)# ip pim sparse-mode
R3(config-if)# interface fastethernet 0/0
R3(config-if)# ip pim sparse-mode
R3(config-if)# interface serial 0/0/0
R3(config-if)# ip pim sparse-mode
R3(config-if)# interface serial 0/0/1
R3(config-if)# ip pim sparse-mode

```

Now that multicast routing and PIM-SM are enabled on R2, do you think a ping from SW1 to the 232.32.32.32 group will receive any replies? Explain.

Yes. ICMP echo replies will be sent from each router because PIM-SM is active throughout the network and there are subscribers to the 232.32.32.32 group.

Verify your answer by pinging the multicast group from SW1. You will not necessarily receive replies from the IP address of the interface on which the

multicast packet was received but rather from whichever interface on the responding router encapsulated the return packet.

```
SW1# ping 232.32.32.32
```

```
Type escape sequence to abort.
```

```
Sending 1, 100-byte ICMP Echos to 232.32.32.32, timeout is 2 seconds:
```

```
Reply to request 0 from 172.16.20.2, 4 ms
```

```
Reply to request 0 from 172.16.102.1, 32 ms
```

```
Reply to request 0 from 172.16.203.3, 32 ms
```

Based on your understanding of IP unicast routing, why are the ICMP echo replies sent from the interfaces shown above?

The interface where the ICMP replies originate is determined by the unicast routing table. The outgoing packets are labeled with the IP address of the interface out which packets will flow to 172.16.20.4.

Step 4: Verify PIM Adjacencies

In this step, we explore PIM adjacencies and how PIM functions over various Layer 2 media.

Issue the **show ip pim neighbors** command to display all adjacent PIM routers.

```
R1# show ip pim neighbor
```

```
PIM Neighbor Table
```

```
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,  
      S - State Refresh Capable
```

Neighbor Address	Interface	Uptime/Expires	Ver	DR Prio/Mode
172.16.13.3	FastEthernet0/0	00:07:22/00:01:19	v2	1 / DR S
172.16.102.2	Serial0/0/0	00:07:23/00:01:22	v2	1 / S
172.16.103.3	Serial0/0/1	00:07:23/00:01:29	v2	1 / S

```
R2# show ip pim neighbor
```

```
PIM Neighbor Table
```

```
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,  
      S - State Refresh Capable
```

Neighbor Address	Interface	Uptime/Expires	Ver	DR Prio/Mode
172.16.102.1	Serial0/0/0	00:08:27/00:01:20	v2	1 / S
172.16.203.3	Serial0/0/1	00:08:29/00:01:22	v2	1 / S

```
R3# show ip pim neighbor
```

```
PIM Neighbor Table
```

```
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,  
      S - State Refresh Capable
```

Neighbor Address	Interface	Uptime/Expires	Ver	DR Prio/Mode
172.16.13.1	FastEthernet0/0	00:08:28/00:01:23	v2	1 / S
172.16.103.1	Serial0/0/0	00:08:59/00:01:19	v2	1 / S
172.16.203.2	Serial0/0/1	00:08:32/00:01:39	v2	1 / S

PIM-DM uses a designated router (DR) as the source of IGMPv1 queries. PIM-SM also supports this type of behavior because it is required in the IGMPv1 protocol.

Which router is the IGMP querier for the 172.16.13.0/24 subnet?

R1 is the DR for the 172.16.13.0/24 subnet and therefore functions as the IGMP querier.

Display information about PIM-enabled interfaces with the **show ip pim interface detail** command.

R1# **show ip pim interface detail**

Address	Interface	Ver/ Mode	Nbr Count	Query Intvl	DR Prior	DR
192.168.1.1	Loopback1	v2/S	0	30	1	192.168.1.1
172.16.13.1	FastEthernet0/0	v2/S	1	30	1	172.16.13.3
172.16.102.1	Serial0/0/0	v2/S	1	30	1	0.0.0.0
172.16.103.1	Serial0/0/1	v2/S	1	30	1	0.0.0.0

R2# **show ip pim interface detail**

Address	Interface	Ver/ Mode	Nbr Count	Query Intvl	DR Prior	DR
192.168.2.1	Loopback2	v2/S	0	30	1	192.168.2.1
172.16.20.2	FastEthernet0/0	v2/S	0	30	1	172.16.20.2
172.16.102.2	Serial0/0/0	v2/S	1	30	1	0.0.0.0
172.16.203.2	Serial0/0/1	v2/S	1	30	1	0.0.0.0

R3# **show ip pim interface detail**

Address	Interface	Ver/ Mode	Nbr Count	Query Intvl	DR Prior	DR
192.168.3.1	Loopback3	v2/S	0	30	1	192.168.3.1
172.16.13.3	FastEthernet0/0	v2/S	1	30	1	172.16.13.3
172.16.103.3	Serial0/0/0	v2/S	1	30	1	0.0.0.0
172.16.203.3	Serial0/0/1	v2/S	1	30	1	0.0.0.0

The serial interfaces use the default DR address 0.0.0.0 as the DR for the interface. Because a multicast is received by either 0 or 1 remote routers on a serial segment, PIM does not need to set up a complex neighbor relationship.

The Ver/Mode column displays the PIM version and mode running on each interface. S refers to sparse mode. D is used for dense mode, and SD indicates the hybrid sparse-dense mode.

Step 5: Verify Multicast Routing Operation

On each router, use the **mrinfo** command to view information about the connected multicast-enabled routers.

R1# **mrinfo**

```
172.16.13.1 [version 12.4] [flags: PMA]:
 192.168.1.1 -> 0.0.0.0 [1/0/pim/querier/leaf]
172.16.13.1 -> 172.16.13.3 [1/0/pim]
172.16.102.1 -> 172.16.102.2 [1/0/pim]
172.16.103.1 -> 172.16.103.3 [1/0/pim]
```

```
R2# mrinfo
172.16.20.2 [version 12.4] [flags: PMA]:
 192.168.2.1 -> 0.0.0.0 [1/0/pim/querier/leaf]
172.16.20.2 -> 0.0.0.0 [1/0/pim/querier/leaf]
172.16.102.2 -> 172.16.102.1 [1/0/pim]
172.16.203.2 -> 172.16.203.3 [1/0/pim]
```

```
R3# mrinfo
172.16.13.3 [version 12.4] [flags: PMA]:
 192.168.3.1 -> 0.0.0.0 [1/0/pim/querier/leaf]
172.16.13.3 -> 172.16.13.1 [1/0/pim/querier]
172.16.103.3 -> 172.16.103.1 [1/0/pim]
172.16.203.3 -> 172.16.203.2 [1/0/pim]
```

Each router realizes that the loopback interfaces are topological leaves in which PIM will never establish an adjacency with any other routers. These routers also record the neighboring multicast router addresses and the multicast routing protocols they utilize.

As discussed in the previous step, IGMPv1 requires the multicast routing protocol to elect a querier on a multi-access network.

Based on the information in the output of the **mrinfo** command, which interface is the querier on VLAN 13?

R1's Fast Ethernet 0/0 interface with IP address 172.16.13.1 is the IGMP querier for the 172.16.13.0/24 subnet.

Generate a stream of multicast data to the group by issuing an extended ping from SW1 with a repeat count of 100.

```
SW1# ping
Protocol [ip]:
Target IP address: 232.32.32.32
Repeat count [1]: 100
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 232.32.32.32, timeout is 2 seconds:

Reply to request 0 from 172.16.20.2, 8 ms
Reply to request 0 from 172.16.102.1, 36 ms
Reply to request 0 from 172.16.203.3, 36 ms
Reply to request 1 from 172.16.20.2, 4 ms
Reply to request 1 from 172.16.102.1, 28 ms
Reply to request 1 from 172.16.203.3, 28 ms
Reply to request 2 from 172.16.20.2, 8 ms
```

```
Reply to request 2 from 172.16.102.1, 32 ms
Reply to request 2 from 172.16.203.3, 32 ms
...
```

On each of the routers, you should see that PIM and IGMP have communicated to install the 232.32.32.32 multicast group in the multicast routing table. Verify this with the **show ip mroute** command on each router.

```
R1# show ip mroute
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
```

```
(* , 232.32.32.32), 02:26:04/00:03:21, RP 192.168.1.1, flags: SJCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  FastEthernet0/0, Forward/Sparse, 02:25:34/00:03:21
  Serial0/0/1, Forward/Sparse, 02:25:34/00:00:00
  Serial0/0/0, Forward/Sparse, 02:25:35/00:00:00
  Loopback1, Forward/Sparse, 02:26:04/00:02:43
```

```
(172.16.20.4, 232.32.32.32), 00:00:14/00:02:59, flags: T
Incoming interface: FastEthernet0/0, RPF nbr 172.16.13.3
Outgoing interface list:
  Loopback1, Forward/Sparse, 00:00:14/00:02:45
  Serial0/0/1, Forward/Sparse, 00:00:14/00:02:45
```

```
(* , 224.0.1.40), 01:40:34/00:02:45, RP 0.0.0.0, flags: DCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  Serial0/0/1, Forward/Sparse, 01:40:34/00:00:00
  Serial0/0/0, Forward/Sparse, 01:40:34/00:00:00
  FastEthernet0/0, Forward/Sparse, 01:40:36/00:00:00
  Loopback1, Forward/Sparse, 01:40:36/00:02:44
```

```
R2# show ip mroute
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
```

```
(* , 232.32.32.32), 02:26:14/stopped, RP 192.168.1.1, flags: SJCLF
Incoming interface: Serial0/0/1, RPF nbr 172.16.203.3
Outgoing interface list:
  Serial0/0/0, Forward/Sparse, 02:26:10/00:00:00
  Loopback2, Forward/Sparse, 02:26:14/00:02:32
```

```
(172.16.20.4, 232.32.32.32), 00:00:19/00:03:28, flags: LFT
Incoming interface: FastEthernet0/0, RPF nbr 0.0.0.0
Outgoing interface list:
  Serial0/0/1, Forward/Sparse, 00:00:20/00:03:09, A
  Loopback2, Forward/Sparse, 00:00:20/00:02:39
  Serial0/0/0, Forward/Sparse, 00:00:20/00:02:39, A
```

```
(* , 224.0.1.40), 01:38:49/00:02:34, RP 0.0.0.0, flags: DCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  Serial0/0/1, Forward/Sparse, 01:38:49/00:00:00
  Serial0/0/0, Forward/Sparse, 01:38:49/00:00:00
  Loopback2, Forward/Sparse, 01:38:49/00:02:34
```

R3# **show ip mroute**

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
L - Local, P - Pruned, R - RP-bit set, F - Register flag,
T - SPT-bit set, J - Join SPT, M - MSDP created entry,
X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
U - URD, I - Received Source Specific Host Report,
Z - Multicast Tunnel, z - MDT-data group sender,
Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

```
(* , 232.32.32.32), 02:26:16/00:02:47, RP 192.168.1.1, flags: SJCL
Incoming interface: FastEthernet0/0, RPF nbr 172.16.13.1
Outgoing interface list:
  Serial0/0/1, Forward/Sparse, 00:30:25/00:02:32
  Loopback3, Forward/Sparse, 02:26:16/00:02:23
```

```
(172.16.20.4, 232.32.32.32), 00:00:24/00:03:29, flags: LT
Incoming interface: Serial0/0/1, RPF nbr 172.16.203.2
Outgoing interface list:
  FastEthernet0/0, Forward/Sparse, 00:00:25/00:03:06
  Loopback3, Forward/Sparse, 00:00:25/00:02:34
```

```
(* , 224.0.1.40), 02:26:17/00:02:22, RP 0.0.0.0, flags: DCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  Loopback3, Forward/Sparse, 02:26:17/00:02:22
```

Multicast forwarding state refers to the current set of (S, G) and (*, G) entries in the multicast routing table. In general, “state” is defined as a stored set of properties at a given point in time. In unicast tables, the state for a destination network is referred to as the route to that network. A unicast IP route is essentially stored in the unicast routing table as a destination network address and mask, along with the next-hop routing information.

Multicast routing tables contain significantly more information than unicast routing tables because a multicast routing table must track many time-based variables for sources and group members. For instance, a multicast router must track the time when the last packet from a source is received, the outgoing interface list, and the RPF neighbors for each source-based and shared tree it maintains. These variables are not stored in a PIM topology table but in the

multicast routing table natively. Multicast routing state is subject to quick changes based on source registrations and group subscriptions.

In PIM-DM, the IP multicast routing table indicates whether the PIM interfaces negotiated to flood multicast data to that group out an interface or pruned (S, G) multicast traffic from being sent out that interface.

In the multicast tables generated above, notice that PIM-SM does not refer to interfaces as pruned.

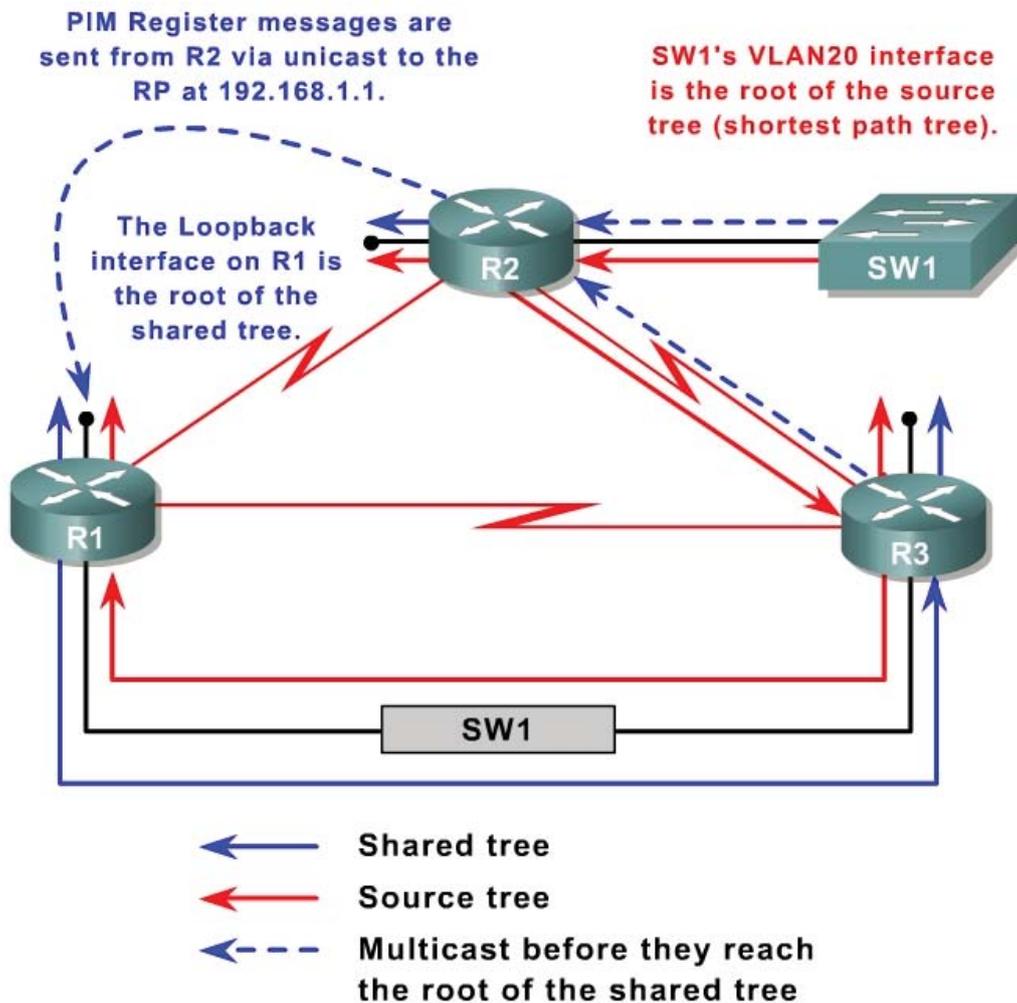
Why does PIM-SM not need to record the interfaces over which the router will prune (S, G) streams from flowing?

PIM-DM acts by flooding and pruning. Thus it maintains an outgoing interface list of all of the multicast interfaces facing away from the upstream source. PIM-DM blocks traffic to unnecessary subnets by pruning those interfaces from the source tree (shortest path tree). PIM-SM only sends multicast traffic out interfaces that have requested it. Therefore, PIM-SM's default state is to not send multicast traffic out of any of its interfaces. When requests are received to forward traffic, it *adds* those interfaces to the outgoing interface list. When requests are received to stop forwarding traffic, PIM-SM *removes* those interfaces from the outgoing interface list. PIM-SM does not need to record in the outgoing interface list any interfaces that do not request traffic because its default state is to not forward that traffic.

When a PIM-SM router receives an explicit join message from IGMP on an interface, it adds that interface to the outgoing interface list of the source-based tree. The source-based tree is also known as the shortest-path tree and is represented in the routing table by an (S, G) entry. Since PIM-SM is a registration-based multicast service, it simply drops from the list the interfaces to which it does not need to send data.

Based on the output of the **show ip mroute** command shown above:

- Trace the (*, G) shared tree on the diagram below. Be sure to reference the RPF interfaces for the (*, G) traffic.
- Trace the (172.16.20.4, 232.32.32.32) source-based tree.



From which PIM neighbor will each loopback interface receive data?

- R1 Loopback 1: R3's Fast Ethernet 0/0 interface, 172.16.13.3
- R2 Loopback 2: PIM will forward (172.16.20.4, 232.32.32.32) traffic directly to Loopback 2.
- R3 Loopback 3: R2's Serial 0/0/1 interface, 172.16.203.2

Step 6: Verify PIM-SM Registration and SPT Cutover

You are interested in viewing how sources register with the RP. You are also interested in exploring pruning data from the shared tree and the cutover to the source tree. Wait for the multicast stream from 172.16.20.4 to complete and for the (S, G) state in the multicast routing tables to expire. Then issue the **debug ip igmp** and **debug ip pim** commands on all routers.

PIM relies on the unicast IP routing table to construct shortest-path trees from the sources to the multicast subscribers. PIM interfaces send control messages to determine which neighbor is closer to the source in terms of the unicast routing information on each neighbor. PIM-SM neighbors elect a particular

router on the subnet as the forwarder for that (S, G) pair and then prune that (S, G) pair from being forwarded by any other routers on the subnet.

Display the unicast routing table entry for 172.16.20.4 on each router.

```
R1# show ip route 172.16.20.4
Routing entry for 172.16.20.0/24
  Known via "eigrp 1", distance 90, metric 20517120, type internal
  Redistributing via eigrp 1
  Last update from 172.16.13.3 on FastEthernet0/0, 03:05:58 ago
  Routing Descriptor Blocks:
  * 172.16.13.3, from 172.16.13.3, 03:05:58 ago, via FastEthernet0/0
    Route metric is 20517120, traffic share count is 1
    Total delay is 20200 microseconds, minimum bandwidth is 128 Kbit
    Reliability 255/255, minimum MTU 1500 bytes
    Loading 1/255, Hops 2

R2# show ip route 172.16.20.4
Routing entry for 172.16.20.0/24
  Known via "connected", distance 0, metric 0 (connected, via interface)
  Redistributing via eigrp 1
  Routing Descriptor Blocks:
  * directly connected, via FastEthernet0/0
    Route metric is 0, traffic share count is 1

R3# show ip route 172.16.20.4
Routing entry for 172.16.20.0/24
  Known via "eigrp 1", distance 90, metric 20514560, type internal
  Redistributing via eigrp 1
  Last update from 172.16.203.2 on Serial0/0/1, 03:06:09 ago
  Routing Descriptor Blocks:
  * 172.16.203.2, from 172.16.203.2, 03:06:09 ago, via Serial0/0/1
    Route metric is 20514560, traffic share count is 1
    Total delay is 20100 microseconds, minimum bandwidth is 128 Kbit
    Reliability 255/255, minimum MTU 1500 bytes
    Loading 1/255, Hops 1
```

Recall from the previous lab: What does PIM use to select the forwarder for a multicast routing entry on a point-to-point link?

PIM Assert messages are used to select the forwarder for each (S, G) pair on point-to-point links.

Begin the same extended ping on SW1 to the multicast group. Examine the debugging messages shown on R1.

```
R1# debug ip pim
*Nov  6 16:10:08.879: PIM(0): Received v2 Register on FastEthernet0/0 from
172.16.203.2
*Nov  6 16:10:08.879:           for 172.16.20.4, group 232.32.32.32
*Nov  6 16:10:08.883: PIM(0): Insert (172.16.20.4,232.32.32.32) join in nbr
172.16.13.3's queue
*Nov  6 16:10:08.883: PIM(0): Forward decapsulated data packet for
232.32.32.32 on FastEthernet0/0
*Nov  6 16:10:08.883: PIM(0): Forward decapsulated data packet for
232.32.32.32 on Serial0/0/1
```

```
*Nov 6 16:10:08.883: PIM(0): Forward decapsulated data packet for
232.32.32.32 on Serial0/0/0
*Nov 6 16:10:08.883: PIM(0): Forward decapsulated data packet for
232.32.32.32 on Loopback1
... continued below ...
```

Initially, R2 begins encapsulating multicast packets from 172.16.20.4 in unicast PIMv2 register packets and sends those unicast messages to the RP. This packet is sent via R2's shortest path to 192.168.1.1 and received on R1's FastEthernet0/0 interface.

The RP sends a PIM control message to the downstream neighbor in the shared tree informing R3 to record (172.16.20.4, 232.32.32.32) state in its multicast routing table. R1 then forwards the newly decapsulated *multicast* packets to 232.32.32.32 down the shared tree.

```
... continued from above ...
R1#
*Nov 6 16:10:08.883: PIM(0): Building Join/Prune packet for nbr 172.16.13.3
*Nov 6 16:10:08.883: PIM(0): Adding v2 (172.16.20.4/32, 232.32.32.32), S-bit
Join
*Nov 6 16:10:08.883: PIM(0): Send v2 join/prune to 172.16.13.3
(FastEthernet0/0)
*Nov 6 16:10:08.907: PIM(0): Received v2 Assert on Serial0/0/0 from
172.16.102.2
*Nov 6 16:10:08.907: PIM(0): Assert metric to source 172.16.20.4 is [0/0]
*Nov 6 16:10:08.907: PIM(0): We lose, our metric [90/20517120]
*Nov 6 16:10:08.907: PIM(0): Prune Serial0/0/0/224.0.0.2 from
(172.16.20.4/32, 232.32.32.32) - deleted
... continued below ...
```

Because Loopback1 has received the first multicast packet from SW1, R1 cuts over to the SPT. R1 builds and sends a join packet for R3 because R3 is its RPF neighbor for source 172.16.20.4. As soon as PIM routers accumulate state about the (S, G) pair, they can begin to cut over to the source-based tree.

By default, when a PIM router with a connected multicast leaf node receives the first packet from a source to a group via the shared tree, it cuts over to the shortest path tree (SPT). You can also manually configure this as a bandwidth threshold using the **ip pim spt-threshold** command. When the (S, G) stream reaches the threshold bandwidth (in kbps), the PIM router switches to the SPT.

The cutover process is initiated on R2 and R3 by the first (172.16.20.4, 232.32.32.32) multicast packet. R1 and R2 compare administrative distances and metrics regarding their shortest path to the source, and R2, the winner, sends a prune message to R1 indicating that it should not send traffic out Serial0/0/0 to 232.32.32.32. R2's SPT state process completes.

```
... continued from above ...
R1#
*Nov 6 16:10:10.879: PIM(0): Received v2 Register on FastEthernet0/0 from
172.16.203.2
*Nov 6 16:10:10.879: for 172.16.20.4, group 232.32.32.32
```

```

*Nov 6 16:10:10.879: PIM(0): Forward decapsulated data packet for
232.32.32.32 on FastEthernet0/0
*Nov 6 16:10:10.879: PIM(0): Forward decapsulated data packet for
232.32.32.32 on Serial0/0/1
*Nov 6 16:10:10.879: PIM(0): Forward decapsulated data packet for
232.32.32.32 on Serial0/0/0
*Nov 6 16:10:10.879: PIM(0): Forward decapsulated data packet for
232.32.32.32 on Loopback1
*Nov 6 16:10:10.891: PIM(0): Received v2 Join/Prune on FastEthernet0/0 from
172.16.13.3, to us
*Nov 6 16:10:10.895: PIM(0): Prune-list: (172.16.20.4/32, 232.32.32.32) RPT-
bit set
... continued below ...

```

R2 continues encapsulating multicast data in unicast packets and sending them to the RP via R3.

Since all the multicast receivers have now obtained (S, G) SPT state for (172.16.20.4, 232.32.32.32), R3 decides it does not need R1 to continue sending this traffic down the shared tree. R3 sends a prune message to R1 via the Ethernet link between them.

```

... continued from above ...
R1#
*Nov 6 16:10:10.903: PIM(0): Received v2 Assert on Serial0/0/0 from
172.16.102.2
*Nov 6 16:10:10.903: PIM(0): Assert metric to source 172.16.20.4 is [0/0]
*Nov 6 16:10:10.979: PIM(0): Insert (172.16.20.4,232.32.32.32) join in nbr
172.16.13.3's queue
*Nov 6 16:10:10.979: PIM(0): Building Join/Prune packet for nbr 172.16.13.3
*Nov 6 16:10:10.979: PIM(0): Adding v2 (172.16.20.4/32, 232.32.32.32), S-bit
Join
*Nov 6 16:10:10.979: PIM(0): Send v2 join/prune to 172.16.13.3
(FastEthernet0/0)
*Nov 6 16:10:12.879: PIM(0): Received v2 Register on FastEthernet0/0 from
172.16.203.2
*Nov 6 16:10:12.879: for 172.16.20.4, group 232.32.32.32
*Nov 6 16:10:12.879: PIM(0): Send v2 Register-Stop to 172.16.203.2 for
172.16.20.4, group 232.32.32.32

```

The first two messages indicate that R2 sent a periodic Assert message on the serial link between them. The Insert, Building, Adding, and Send messages represent R1 acting as a standard multicast router and sending a join message to its upstream SPT neighbor which requests (S, G) traffic to continue.

R2 sends another Register packet to the RP with an encapsulated multicast packet. R1, acting as the RP, checks its routing table for outgoing interfaces on the shared tree. Since all downstream branches of the shared tree have been pruned, indicating that R2 and R3 have switched over to the SPT, R1 sends a Register-Stop message to R2 telling it to stop forwarding multicast packets as unicasts to the RP.

At this point, all multicast routers are listening directly to the source via their shortest path to 172.16.20.4, and the current receivers do not need the RP or the shared tree because they are listening directly to the source via the SPT. It

would not be unusual if the router acting as the RP was still a part of the SPT, as it is in this case, because R1's Loopback1 interface subscribes to the 232.32.32.32 group. However, R1 is acting simply as a subscriber to the group and no longer as the RP.

Why does R2 initially encapsulate the multicast data in a unicast packet?

R2 cannot allow the traffic directly into the multicast network, or PIM-SM would degrade into PIM-DM. Traffic first needs to be sent to the rendezvous point at which other routers have rooted their shared trees. Traffic will then be sent down the shared tree until all sources have switched over to the source-based tree. In order to send the traffic to the rendezvous point without sending it through the multicast network (thereby creating multicast state), PIM-SM sends the data as a unicast to the RP.

According to the debugging output shown above, what happens to the multicast data once it reaches the RP?

At the RP, the unicast packets are de-encapsulated and the packets are sent as a multicast down the shared tree.

If R2 simply forwarded the data to all receivers via multicast packets, does that sound similar to another multicast routing protocol or procedure?

If R2 simply forwarded the data to all connected receivers using multicast packets, the PIM-SM network would degrade to PIM-DM.

Conclusion

Why is PIM-SM more efficient?

PIM-SM only stores multicast routing information about the shared trees and about source trees for which it has had some request. PIM-SM is also more efficient than PIM-DM because it does not utilize the amount of bandwidth on leaf networks that have no interest in the multicast traffic.

Why did the routers cut over to the SPT?

Routers cut over to the SPT to minimize the delay between the sources and the group members. This effectively prunes the shared tree out of the branch of the source tree between the source and that particular group member.

Issue the **show ip mroute** command on all routers. Compare the output to the output in Step 3 before you implemented PIM-SM.

```
R1# show ip mroute
<output omitted>
```

```
(*, 232.32.32.32), 04:27:40/stopped, RP 192.168.1.1, flags: SJCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  FastEthernet0/0, Forward/Sparse, 04:27:10/00:02:45
  Serial0/0/1, Forward/Sparse, 04:27:11/00:00:00
  Serial0/0/0, Forward/Sparse, 04:27:12/00:00:00
  Loopback1, Forward/Sparse, 04:27:40/00:02:04

(172.16.20.4, 232.32.32.32), 00:00:05/00:02:58, flags: T
Incoming interface: FastEthernet0/0, RPF nbr 172.16.13.3
Outgoing interface list:
  Loopback1, Forward/Sparse, 00:00:05/00:02:54
  Serial0/0/1, Forward/Sparse, 00:00:05/00:02:54

(*, 224.0.1.40), 03:42:11/00:02:01, RP 0.0.0.0, flags: DCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  Serial0/0/1, Forward/Sparse, 03:42:11/00:00:00
  Serial0/0/0, Forward/Sparse, 03:42:11/00:00:00
  FastEthernet0/0, Forward/Sparse, 03:42:12/00:00:00
  Loopback1, Forward/Sparse, 03:42:12/00:02:00
```

```
R2# show ip mroute
<output omitted>
```

```
(*, 232.32.32.32), 04:28:00/stopped, RP 192.168.1.1, flags: SJCLF
Incoming interface: Serial0/0/1, RPF nbr 172.16.203.3
Outgoing interface list:
  Serial0/0/0, Forward/Sparse, 04:27:56/00:00:00
  Loopback2, Forward/Sparse, 04:28:00/00:02:49

(172.16.20.4, 232.32.32.32), 00:00:19/00:03:18, flags: LFT
Incoming interface: FastEthernet0/0, RPF nbr 0.0.0.0
Outgoing interface list:
  Serial0/0/1, Forward/Sparse, 00:00:20/00:03:09, A
  Loopback2, Forward/Sparse, 00:00:20/00:02:48
  Serial0/0/0, Forward/Sparse, 00:00:20/00:02:39, A

(*, 224.0.1.40), 03:40:35/00:02:43, RP 0.0.0.0, flags: DCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  Serial0/0/1, Forward/Sparse, 03:40:35/00:00:00
  Serial0/0/0, Forward/Sparse, 03:40:35/00:00:00
  Loopback2, Forward/Sparse, 03:40:35/00:02:43
```

```
R3# show ip mroute
<output omitted>
```

```
(*, 232.32.32.32), 04:28:01/00:02:47, RP 192.168.1.1, flags: SJCL
```

```

Incoming interface: FastEthernet0/0, RPF nbr 172.16.13.1
Outgoing interface list:
  Serial0/0/1, Forward/Sparse, 02:32:10/00:02:46
  Loopback3, Forward/Sparse, 04:28:01/00:02:37

(172.16.20.4, 232.32.32.32), 00:00:23/00:03:28, flags: LT
Incoming interface: Serial0/0/1, RPF nbr 172.16.203.2
Outgoing interface list:
  FastEthernet0/0, Forward/Sparse, 00:00:24/00:03:07
  Loopback3, Forward/Sparse, 00:00:24/00:02:36

(*, 224.0.1.40), 04:28:02/00:02:31, RP 0.0.0.0, flags: DCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  Loopback3, Forward/Sparse, 04:28:02/00:02:31

```

Does the implementation of PIM-SM in CrushASodaCan.com's multicast network save bandwidth compared to the PIM-DM implementation? Explain.

Yes. Bandwidth has been saved because the flooding and pruning inherent in PIM-DM does not take place.

Although not part of this lab activity, could the implementation of PIM-SM also save memory by decreasing multicast routing state? Explain.

Normally, the implementation of PIM-SM indeed does decrease multicast state. Since only state about shared and source trees that pass through the local router need be stored, all other multicast state does not get created. The only (*, G) and (S, G) entries on a router running PIM-SM are for the trees that pass through that router.

Final Configurations

```

R1# show run
!
hostname R1
!
interface Loopback1
 ip address 192.168.1.1 255.255.255.0
 ip pim sparse-mode
 ip igmp join-group 232.32.32.32
!
interface FastEthernet0/0
 ip address 172.16.13.1 255.255.255.0
 ip pim sparse-mode
 no shutdown
!
interface Serial0/0/0
 bandwidth 64
 ip address 172.16.102.1 255.255.255.248
 ip pim sparse-mode
 clock rate 64000

```

```

no shutdown
!
interface Serial0/0/1
bandwidth 64
ip address 172.16.103.1 255.255.255.248
ip pim sparse-mode
no shutdown
!
router eigrp 1
network 172.16.0.0
network 192.168.0.0 0.0.255.255
auto-summary
!
ip pim rp-address 192.168.1.1 32
!
access-list 32 permit 232.32.32.32
!
end

```

R2# **show run**

```

!
hostname R2
!
interface Loopback2
ip address 192.168.2.1 255.255.255.0
ip pim sparse-mode
ip igmp join-group 232.32.32.32
!
interface FastEthernet0/0
ip address 172.16.20.2 255.255.255.0
ip pim sparse-mode
no shutdown
!
interface Serial0/0/0
bandwidth 64
ip address 172.16.102.2 255.255.255.248
ip pim sparse-mode
no shutdown
!
interface Serial0/0/1
bandwidth 128
ip address 172.16.203.2 255.255.255.248
ip pim sparse-mode
clock rate 128000
no shutdown
!
router eigrp 1
network 172.16.0.0
network 192.168.0.0 0.0.255.255
auto-summary
!
ip pim rp-address 192.168.1.1 32
!
access-list 32 permit 232.32.32.32
!
end

```

R3# **show run**

```

!
hostname R3
!
interface Loopback3
ip address 192.168.3.1 255.255.255.0

```

```

ip pim sparse-mode
ip igmp join-group 232.32.32.32
!
interface FastEthernet0/0
ip address 172.16.13.3 255.255.255.0
ip pim sparse-mode
no shutdown
!
interface Serial0/0/0
bandwidth 64
ip address 172.16.103.3 255.255.255.248
ip pim sparse-mode
clock rate 64000
no shutdown
!
interface Serial0/0/1
bandwidth 128
ip address 172.16.203.3 255.255.255.248
ip pim sparse-mode
no shutdown
!
router eigrp 1
network 172.16.0.0
network 192.168.0.0 0.0.255.255
auto-summary
!
ip pim rp-address 192.168.1.1 32
!
access-list 32 permit 232.32.32.32
!
end

```

SW1# **show run**

```

!
hostname SW1
!
interface FastEthernet0/1
switchport access vlan 13
switchport mode access
!
interface FastEthernet0/3
switchport access vlan 20
switchport mode access
!
interface FastEthernet0/5
switchport access vlan 13
switchport mode access
!
interface Vlan20
ip address 172.16.20.4 255.255.255.0
no shutdown
!
ip default-gateway 172.16.20.2
!
end

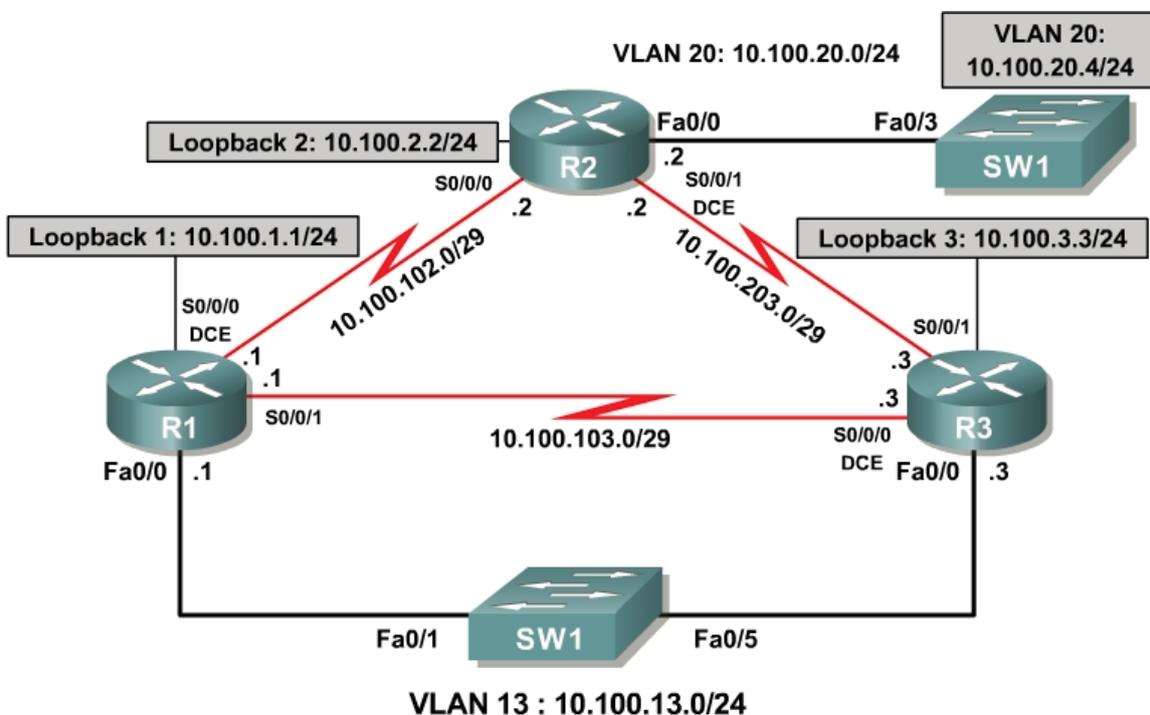
```

Lab 7-4 Routing IP Multicast with PIM Sparse-Dense Mode

Learning Objectives

- Configure multiple multicast sources and groups via IGMP
- Configure and verify PIM sparse-dense mode operation and adjacencies
- Configure and verify automatic rendezvous points and mapping agents
- Force PIM sparse-dense mode to fail over to dense mode operation

Topology Diagram



Scenario

After the incredible community response to CrushASodaCan.com TV spots, a skeptical citizen began questioning the usefulness of crushing aluminum cans before recycling them. After conducting a thorough case study, he founded the organization JustRecycleTheCan.org, and began producing TV announcements encouraging citizens to recycle but informing them that it is unnecessary to crush cans before committing them to the safe care of a recycle bin.

The founder of JustRecycleTheCan.org has contracted you to implement multicast on his network so that he can also preview TV spots before releasing

them to the public. However, he requests that you implement PIM in a more redundant, scalable manner than you did for CrushASodaCan.com.

Fully configure multicast routing on his network according to the requests of the founder of the organization. Verify that if the route to the rendezvous point (RP) is lost, multicast receivers on remote networks can still receive multicast streams.

Step 1: Configure Addressing and Implement IGMP

Paste in the following initial configurations:

```
R1:
!
hostname R1
!
interface Loopback1
 ip address 10.100.1.1 255.255.255.0
!
interface FastEthernet0/0
 ip address 10.100.13.1 255.255.255.0
 no shutdown
!
interface Serial0/0/0
 bandwidth 64
 ip address 10.100.102.1 255.255.255.248
 clock rate 64000
 no shutdown
!
interface Serial0/0/1
 bandwidth 64
 ip address 10.100.103.1 255.255.255.248
 no shutdown
!
end

R2:
!
hostname R2
!
interface Loopback2
 ip address 10.100.2.2 255.255.255.0
!
interface FastEthernet0/0
 ip address 10.100.20.2 255.255.255.0
 no shutdown
!
interface Serial0/0/0
 bandwidth 64
 ip address 10.100.102.2 255.255.255.248
 no shutdown
!
interface Serial0/0/1
 bandwidth 128
 ip address 10.100.203.2 255.255.255.248
 clock rate 128000
 no shutdown
!
end
```

```

R3:
!
hostname R3
!
interface Loopback3
 ip address 10.100.3.3 255.255.255.0
!
interface FastEthernet0/0
 ip address 10.100.13.3 255.255.255.0
 no shutdown
!
interface Serial10/0/0
 bandwidth 64
 ip address 10.100.103.3 255.255.255.248
 clock rate 64000
 no shutdown
!
interface Serial10/0/1
 bandwidth 128
 ip address 10.100.203.3 255.255.255.248
 no shutdown
!
end

```

```

SW1:
!
hostname SW1
!
interface FastEthernet0/1
 switchport access vlan 13
 switchport mode access
!
interface FastEthernet0/3
 switchport access vlan 20
 switchport mode access
!
interface FastEthernet0/5
 switchport access vlan 13
 switchport mode access
!
end

```

Use a switched virtual interface (SVI) on SW1 to simulate a multicast source on the VLAN 20 subnet. Use this SVI to send a repeated multicast ping that simulates multicast traffic while you set up the network. Assign the SVI the IP address 10.100.20.4/24 with a default gateway of 10.100.20.2.

```

SW1# conf t
SW1(config)# ip default-gateway 10.100.20.2
SW1(config)# interface vlan 20
SW1(config-if)# ip address 10.100.20.4 255.255.255.0
SW1(config-if)# no shutdown

```

Using IGMP, subscribe each of the loopback interfaces on the three routers to the multicast groups 225.25.25.25. Subscribe R1 and R3 loopbacks to 226.26.26.26 as well.

```

R1# conf t
R1(config)# interface loopback 1
R1(config-if)# ip igmp join-group 225.25.25.25

```

```

R1(config-if)# ip igmp join-group 226.26.26.26

R2# conf t
R2(config)# interface loopback 2
R2(config-if)# ip igmp join-group 225.25.25.25

R3# conf t
R3(config)# interface loopback 3
R3(config-if)# ip igmp join-group 225.25.25.25
R3(config-if)# ip igmp join-group 226.26.26.26

```

Verify that each of the interfaces has subscribed to the multicast group using the **show ip igmp groups** command.

```

R1# show ip igmp groups
IGMP Connected Group Membership
Group Address  Interface  Uptime    Expires   Last Reporter  Group Accounted
226.26.26.26  Loopback1  00:01:51  00:02:03  10.100.1.1     10.100.1.1
225.25.25.25  Loopback1  00:01:51  00:02:03  10.100.1.1     10.100.1.1
R2# show ip igmp groups
IGMP Connected Group Membership
Group Address  Interface  Uptime    Expires   Last Reporter  Group Accounted
225.25.25.25  Loopback2  00:01:51  00:02:03  10.100.2.2     10.100.2.2
R3# show ip igmp groups
IGMP Connected Group Membership
Group Address  Interface  Uptime    Expires   Last Reporter  Group Accounted
226.26.26.26  Loopback3  00:01:51  00:02:03  10.100.3.3     10.100.3.3
225.25.25.25  Loopback3  00:01:51  00:02:03  10.100.3.3     10.100.3.3

```

Step 2: Configure Single-Area OSPF

Configure single-area OSPF on each router with the following configuration:

```

router ospf 1
 network 10.0.0.0 0.255.255.255 area 0

```

After you apply this configuration and the OSPF adjacencies form, run the following TCL script to verify full unicast connectivity:

```

foreach address {
10.100.1.1
10.100.13.1
10.100.102.1
10.100.103.1
10.100.2.2
10.100.20.2
10.100.102.2
10.100.203.2
10.100.3.3
10.100.13.3
10.100.103.3
10.100.203.3
10.100.20.4
} { ping $address }

```

Compare the output you receive with the output found in Appendix A (all pings successful). Make sure that you have ICMP replies from the SVI interface on SW1. If SW1 has a reachable default gateway of R2's FastEthernet0/0 interface

and if all routers have full connectivity between all subnets, you should be able to ping the SVI address and receive all replies.

Step 3: Implement PIM Sparse-Dense Mode

First, enable your routers to accumulate and maintain multicast state using the **ip multicast-routing** command in global configuration mode.

```
R1(config)# ip multicast-routing
```

```
R2(config)# ip multicast-routing
```

```
R3(config)# ip multicast-routing
```

Next, enable PIM sparse-dense mode on all interfaces using the **ip pim sparse-dense-mode** command in interface configuration mode.

```
R1(config)# interface loopback 1
R1(config-if)# ip pim sparse-dense-mode
R1(config-if)# interface fastethernet 0/0
R1(config-if)# ip pim sparse-dense-mode
R1(config-if)# interface serial 0/0/0
R1(config-if)# ip pim sparse-dense-mode
R1(config-if)# interface serial 0/0/1
R1(config-if)# ip pim sparse-dense-mode
```

```
R2(config)# interface loopback 2
R2(config-if)# ip pim sparse-dense-mode
R2(config-if)# interface fastethernet 0/0
R2(config-if)# ip pim sparse-dense-mode
R2(config-if)# interface serial 0/0/0
R2(config-if)# ip pim sparse-dense-mode
R2(config-if)# interface serial 0/0/1
R2(config-if)# ip pim sparse-dense-mode
```

```
R3(config)# interface loopback 1
R3(config-if)# ip pim sparse-dense-mode
R3(config-if)# interface fastethernet 0/0
R3(config-if)# ip pim sparse-dense-mode
R3(config-if)# interface serial 0/0/0
R3(config-if)# ip pim sparse-dense-mode
R3(config-if)# interface serial 0/0/1
R3(config-if)# ip pim sparse-dense-mode
```

After completing the previous two labs, you should have an idea of how PIM-SM operates. The following questions test your understanding of the concepts.

What is the purpose of the PIM-SM rendezvous point?

The rendezvous point establishes a central point of attachment between sources and multicast group members. Multicast routers closest to the source of the multicast traffic will forward multicast packets to the RP for that group as unicasts. Multicast routers with attached listeners to that group will connect to the rendezvous point as the source of the shared tree for that group.

If no rendezvous points are reachable in an PIM-SM network, what functionality is lost? (Hint: Will PIMv2 Register messages be sent?)

No multicasts will be sent via PIM Register messages to the rendezvous point. Group subscribers will not be able to receive multicasts from sources via the shared tree.

If you implement PIM-SM with a static RP as in Lab 7.3, which routers identify the RP? Which routers identify the group-to-RP mappings?

Each router statically identifies the RP. Each router statically identifies what groups it will use for each RP.

Other than configuring a static RP globally for a network or on a per-group basis, which other ways can you configure an RP in sparse mode or sparse-dense mode? Why are other methods useful?

1. You can assign an RP dynamically via the Cisco-proprietary Auto-RP mechanism. This allows the RP to be dynamically assigned.
2. Advanced methods beyond the scope of this curriculum include using a Bootstrap Router (BSR) or Anycast RP. BSR allows mapping agents to be elected as well as RPs. Anycast RP allows an established IP address to be advertised from multiple points in a network. Each of these processes allows dynamic assignment of RPs or mapping agents.

As shown in the last lab, it is important to locate RPs in locations that are reachable via a unicast routing protocol when available. In this lab, you will configure two RP candidates for the 225.25.25.25 multicast group. Allow only R1 to be elected as the RP for the 226.26.26.26 group, but discover all RP candidates dynamically. Use PIM sparse-dense mode and Auto-RP to achieve this functionality.

PIM sparse-dense mode manages multicast groups dynamically between sparse mode and dense mode. Sparse-dense mode applies sparse mode processes and algorithms to any multicast groups for which it can discover an RP via static configuration, Auto-RP, or the more advanced Bootstrap Router (BSR) process. All multicast groups for which a sparse-dense mode router cannot discover an RP run in PIM-DM. This applies to PIM-SM networks as well. This process is referred to as dense mode fallback and can be disabled with the **no ip pim dm-fallback** command in global configuration mode.

Without additional configuration, will the multicast groups 225.25.25.25 and 226.26.26.26 effectively operate in sparse mode or dense mode? Explain.

These multicast groups will operate in dense mode until an RP can be located. Because the nearest router to the multicast sources cannot locate the RP, it cannot encapsulate the multicast packets in unicast packets. Therefore, it simply floods the multicasts to all connected multicast interfaces in dense mode.

Will a ping from SW1 to the 225.25.25.25 group receive any replies? Explain.

Yes, because the group is operating in dense mode until a rendezvous point can be found.

Issue a stream of multicast pings from the SVI on SW1 to generate the (S, G) state on your routers.

```
SW1# ping
Protocol [ip]:
Target IP address: 225.25.25.25
Repeat count [1]: 1
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 225.25.25.25, timeout is 2 seconds:

Reply to request 0 from 10.100.20.2, 8 ms
Reply to request 0 from 10.100.13.1, 32 ms
Reply to request 0 from 10.100.203.3, 20 ms
```

Display the multicast routing table on each router with the **show ip mroute** command.

```
R1# show ip mroute
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
L - Local, P - Pruned, R - RP-bit set, F - Register flag,
T - SPT-bit set, J - Join SPT, M - MSDP created entry,
X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
U - URD, I - Received Source Specific Host Report,
Z - Multicast Tunnel, z - MDT-data group sender,
Y - Joined MDT-data group, y - Sending to MDT-data group
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 226.26.26.26), 00:49:00/00:02:48, RP 0.0.0.0, flags: DCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
Serial0/0/0, Forward/Sparse-Dense, 00:47:56/00:00:00
```

```

Serial0/0/1, Forward/Sparse-Dense, 00:48:30/00:00:00
FastEthernet0/0, Forward/Sparse-Dense, 00:48:38/00:00:00
Loopback1, Forward/Sparse-Dense, 00:49:00/00:00:00

(*, 225.25.25.25), 00:50:35/stopped, RP 0.0.0.0, flags: DCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  Serial0/0/0, Forward/Sparse-Dense, 00:47:57/00:00:00
  Serial0/0/1, Forward/Sparse-Dense, 00:48:31/00:00:00
  FastEthernet0/0, Forward/Sparse-Dense, 00:48:39/00:00:00
  Loopback1, Forward/Sparse-Dense, 00:50:35/00:00:00

(10.100.20.4, 225.25.25.25), 00:00:43/00:02:58, flags: LT
Incoming interface: FastEthernet0/0, RPF nbr 10.100.13.3
Outgoing interface list:
  Loopback1, Forward/Sparse-Dense, 00:00:51/00:00:00
  Serial0/0/1, Prune/Sparse-Dense, 00:00:51/00:02:11
  Serial0/0/0, Prune/Sparse-Dense, 00:00:51/00:02:11

(*, 224.0.1.40), 00:50:43/00:02:47, RP 0.0.0.0, flags: DCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  Serial0/0/0, Forward/Sparse-Dense, 00:48:05/00:00:00
  Serial0/0/1, Forward/Sparse-Dense, 00:48:39/00:00:00
  FastEthernet0/0, Forward/Sparse-Dense, 00:48:47/00:00:00
  Loopback1, Forward/Sparse-Dense, 00:50:43/00:00:00

```

R2# show ip mroute

<output omitted>

```

(*, 225.25.25.25), 00:00:45/stopped, RP 0.0.0.0, flags: DL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  Serial0/0/1, Forward/Sparse-Dense, 00:00:45/00:00:00
  Serial0/0/0, Forward/Sparse-Dense, 00:00:45/00:00:00

(10.100.20.4, 225.25.25.25), 00:00:45/00:03:01, flags: LT
Incoming interface: FastEthernet0/0, RPF nbr 0.0.0.0
Outgoing interface list:
  Serial0/0/0, Prune/Sparse-Dense, 00:00:45/00:02:17, A
  Serial0/0/1, Forward/Sparse-Dense, 00:00:46/00:00:00

(*, 224.0.1.40), 00:48:03/00:02:48, RP 0.0.0.0, flags: DCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  Serial0/0/1, Forward/Sparse-Dense, 00:47:56/00:00:00
  Serial0/0/0, Forward/Sparse-Dense, 00:48:00/00:00:00
  FastEthernet0/0, Forward/Sparse-Dense, 00:48:03/00:00:00

```

R3# show ip mroute

<output omitted>

```

(*, 226.26.26.26), 00:00:03/00:02:58, RP 0.0.0.0, flags: DCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  Serial0/0/1, Forward/Sparse-Dense, 00:00:03/00:00:00
  Serial0/0/0, Forward/Sparse-Dense, 00:00:03/00:00:00
  FastEthernet0/0, Forward/Sparse-Dense, 00:00:03/00:00:00
  Loopback3, Forward/Sparse-Dense, 00:00:03/00:00:00

(*, 225.25.25.25), 00:00:05/00:02:55, RP 0.0.0.0, flags: DCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:

```

```
Serial0/0/1, Forward/Sparse-Dense, 00:00:05/00:00:00
Serial0/0/0, Forward/Sparse-Dense, 00:00:05/00:00:00
FastEthernet0/0, Forward/Sparse-Dense, 00:00:05/00:00:00
Loopback3, Forward/Sparse-Dense, 00:00:05/00:00:00
```

```
(10.100.20.4, 225.25.25.25), 00:03:26/00:02:56, flags: LT
Incoming interface: Serial0/0/1, RPF nbr 10.100.203.2
Outgoing interface list:
  FastEthernet0/0, Forward/Sparse-Dense, 00:03:27/00:00:00
  Serial0/0/0, Prune/Sparse-Dense, 00:00:22/00:02:40, A

(*, 224.0.1.40), 00:51:23/00:02:05, RP 0.0.0.0, flags: DCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  Serial0/0/1, Forward/Sparse-Dense, 00:50:37/00:00:00
  Serial0/0/0, Forward/Sparse-Dense, 00:51:15/00:00:00
  FastEthernet0/0, Forward/Sparse-Dense, 00:51:23/00:00:00
```

Referring to the output above, in which mode is PIM operating if it is not relying on the shared tree to connect it with receivers?

As the 'D' in the output suggests, PIM is operating in dense mode if no RP can be located.

How is this indicated in the multicast routing tables?

The D in the flags for each tree entry in the multicast routing table indicates that PIM is operating in dense mode.

If your output differs from that shown above, verify that you have configured PIM sparse-dense mode correctly on all interfaces by using the **show ip pim interface** command. Make sure that all PIM adjacencies are active by using the **show ip pim neighbor** command.

Step 4: Configure PIM Auto-RP

PIM now operates successfully, but you can configure the routers to use bandwidth more efficiently by forcing multicast groups to use PIM-SM.

PIM-SM rendezvous points introduce multicast sources by acting as a network receptionist that knows the location of all multicast sources for specific groups. PIM-SM creates shared distribution trees for multicast groups represented by (*, G) in each multicast routing table. This shared tree is calculated using the RPF upstream interface to the RP for that group. Thus, the shared tree is essentially a shortest-path tree to the RP address.

When you configured the interfaces in sparse-dense mode, every router implicitly subscribed to the RP Discovery group 224.0.1.40. Use the **show ip**

igmp groups command to verify this. Each router applies the information it receives about group-to-RP mappings via the RP Discovery group to its multicast routing table entries.

```
R1# show ip igmp groups
IGMP Connected Group Membership
Group Address      Interface          Uptime    Expires    Last Reporter
Group Accounted
226.26.26.26       Loopback1         00:42:38  00:02:14  10.100.1.1
225.25.25.25       Loopback1         00:42:38  00:02:09  10.100.1.1
224.0.1.40         FastEthernet0/0   00:41:44  00:02:10  10.100.13.3
224.0.1.40         Loopback1         00:42:38  00:02:13  10.100.1.1
```

When configuring interfaces running only PIM-SM, you also need to apply the **ip pim autorp listener** command to each sparse mode interface. This command allows RP mapping messages to be read by multicast routers and propagated throughout the network in dense mode fashion. Because sparse-dense mode acts in a PIM-DM fashion without an RP for the group, you achieve the same functionality without applying the command.

By default, sparse-dense mode also enables the installation of RP mapping information into the multicast routing table. Verify this with the **show ip pim autorp** command.

```
R1# show ip pim autorp
AutoRP Information:
  AutoRP is enabled.
```

```
PIM AutoRP Statistics: Sent/Received
  RP Announce: 0/0, RP Discovery: 0/0
```

Enable the advertisement of Auto-RP information for the specified groups on R1 and R3. Configure R1 to candidate as an RP for both groups. Allow R3 to only apply as a candidate for the 226.26.26.26 group. Configure access lists to control the groups for which Auto-RP information is sent. Finally, apply these access lists with the **ip pim send-rp-announce** *{interface-type interface-number | ip-address}* **scope ttl-value [group-list access-list]** command. The RP candidates should be the loopback interfaces on R1 and R3. Use a time-to-live (TTL) value of 3 so that RP announcements are not discarded anywhere in your existing network. The scope determines the number of times a multicast packet is routed at Layer 3 before being discarded because the TTL has reached 0.

```
R1(config)# access-list 1 permit 225.25.25.25
R1(config)# access-list 1 permit 226.26.26.26
R1(config)# ip pim send-rp-announce Loopback 1 scope 3 group-list 1

R3(config)# access-list 3 permit 226.26.26.26
R3(config)# ip pim send-rp-announce Loopback 3 scope 3 group-list 3
```

Enable PIM Auto-RP debugging with the **debug ip pim auto-rp** command on R1 and R3. The following excerpt shows that Auto-RP announcements are sent periodically, indicating that the announcements are being flooded from the local router.

```
R1# debug ip pim auto-rp
```

```
*Nov 7 16:08:05.803: Auto-RP(0): Build RP-Announce for 10.100.1.1, PIMv2/v1,
ttl 3, ht 181
*Nov 7 16:08:05.803: Auto-RP(0): Build announce entry for (225.25.25.25/32)
*Nov 7 16:08:05.803: Auto-RP(0): Build announce entry for (226.26.26.26/32)
*Nov 7 16:08:05.803: Auto-RP(0): Send RP-Announce packet on FastEthernet0/0
*Nov 7 16:08:05.803: Auto-RP(0): Send RP-Announce packet on Serial0/0/0
*Nov 7 16:08:05.803: Auto-RP(0): Send RP-Announce packet on Serial0/0/1
*Nov 7 16:08:05.803: Auto-RP: Send RP-Announce packet on Loopback1
```

Continue debugging Auto-RP messages while you configure the mapping agent.

PIM-SM networks using Auto-RP also need one or more mapping agents to inform routers that are listening to the Auto-RP group of the group-to-RP mappings. Configure R1 as the mapping agent for this network using the **ip pim send-rp-discovery** [*interface-type interface-number*] **scope** *ttl-value* command to generate group-to-RP mapping messages from R1.

```
R1(config)# ip pim send-rp-discovery Loopback 1 scope 3
```

You should see the following debug output on R1:

```
R1#
*Nov 7 16:26:08.803: Auto-RP(0): Build RP-Announce for 10.100.1.1, PIMv2/v1,
ttl 3, ht 181
*Nov 7 16:26:08.803: Auto-RP(0): Build announce entry for (225.25.25.25/32)
*Nov 7 16:26:08.803: Auto-RP(0): Build announce entry for (226.26.26.26/32)
*Nov 7 16:26:08.803: Auto-RP(0): Send RP-Announce packet on FastEthernet0/0
*Nov 7 16:26:08.803: Auto-RP(0): Send RP-Announce packet on Serial0/0/0
*Nov 7 16:26:08.803: Auto-RP(0): Send RP-Announce packet on Serial0/0/1
*Nov 7 16:26:08.803: Auto-RP: Send RP-Announce packet on Loopback1
... continued below ...
```

Because you previously configured R1 as an RP candidate, it advertises itself as an RP for the 225.25.25.25 and 226.26.26.26 groups. However, no router will select R1 as the confirmed RP for those groups until it hears from the mapping agent. R1 continues to multicast its announcements of RP candidacy out all interfaces.

```
... continued from above ...
R1#
*Nov 7 16:26:08.803: Auto-RP(0): Received RP-announce, from 10.100.1.1,
RP_cnt 1, ht 181
*Nov 7 16:26:08.803: Auto-RP(0): Added with (225.25.25.25/32, RP:10.100.1.1),
PIMv2 v1
*Nov 7 16:26:08.803: Auto-RP(0): Added with (226.26.26.26/32, RP:10.100.1.1),
PIMv2 v1
*Nov 7 16:26:08.807: Auto-RP(0): Build RP-Discovery packet
*Nov 7 16:26:08.807: Auto-RP: Build mapping (225.25.25.25/32,
RP:10.100.1.1), PIMv2 v1,
*Nov 7 16:26:08.807: Auto-RP: Build mapping (226.26.26.26/32,
RP:10.100.1.1), PIMv2 v1.
*Nov 7 16:26:08.811: Auto-RP(0): Send RP-discovery packet on FastEthernet0/0
(1 RP entries)
*Nov 7 16:26:08.811: Auto-RP(0): Send RP-discovery packet on Serial0/0/0 (1
RP entries)
```

```
*Nov 7 16:26:08.811: Auto-RP(0): Send RP-discovery packet on Serial0/0/1 (1
RP entries)
*Nov 7 16:26:08.811: Auto-RP: Send RP-discovery packet on Loopback1 (1 RP
entries)
```

The router acting as a mapping agent, which in this case is also R1, receives the candidacy announcement that the Auto-RP process on R1 just sent to the 224.0.1.39 Auto-RP announcement group.

R1, acting as an RP mapping agent, elects R1 as the RP for both groups because it has not received any other RP announcements with higher source IP addresses. The mapping agent sends the two group-to-RP mappings via multicast to all Auto-RP listeners in the network. The mappings are sent to the Auto-RP discovery group 224.0.1.40.

```
... continued from above ...
R1#
*Nov 7 16:26:08.819: Auto-RP(0): Received RP-announce, from 10.100.1.1,
RP_cnt 1, ht 181
*Nov 7 16:26:08.819: Auto-RP(0): Update (225.25.25.25/32, RP:10.100.1.1),
PIMv2 v1
*Nov 7 16:26:08.819: Auto-RP(0): Update (226.26.26.26/32, RP:10.100.1.1),
PIMv2 v1
*Nov 7 16:26:08.819: Auto-RP(0): Received RP-announce, from 10.100.1.1,
RP_cnt 1, ht 181
*Nov 7 16:26:08.819: Auto-RP(0): Update (225.25.25.25/32, RP:10.100.1.1),
PIMv2 v1
*Nov 7 16:26:08.823: Auto-RP(0): Update (226.26.26.26/32, RP:10.100.1.1),
PIMv2 v1
... continued below ...
```

The Auto-RP messages that R1 sent previously are received back again on different interfaces because of the dense-mode flooding throughout the network. R1 simply installs the group-to-RP mappings in its table again.

```
... continued from above ...
R1#
*Nov 7 16:26:36.911: Auto-RP(0): Received RP-announce, from 10.100.3.3,
RP_cnt 1, ht 181
*Nov 7 16:26:36.911: Auto-RP(0): Added with (226.26.26.26/32, RP:10.100.3.3),
PIMv2 v1
*Nov 7 16:26:36.915: Auto-RP(0): Build RP-Discovery packet
*Nov 7 16:26:36.915: Auto-RP: Build mapping (225.25.25.25/32,
RP:10.100.1.1), PIMv2 v1,
*Nov 7 16:26:36.915: Auto-RP: Build mapping (226.26.26.26/32,
RP:10.100.3.3), PIMv2 v1.
*Nov 7 16:26:36.915: Auto-RP(0): Send RP-discovery packet on FastEthernet0/0
(2 RP entries)
*Nov 7 16:26:36.915: Auto-RP(0): Send RP-discovery packet on Serial0/0/0 (2
RP entries)
R1#
*Nov 7 16:26:36.915: Auto-RP(0): Send RP-discovery packet on Serial0/0/1 (2
RP entries)
*Nov 7 16:26:36.915: Auto-RP: Send RP-discovery packet on Loopback1 (2 RP
entries)
```

Within a short amount of time, R3 sends its candidacy announcement for the 226.26.26.26 multicast group. R1 receives this packet and, acting as the RP

mapping agent, evaluates it against the existing RP. Since the IP address of 10.100.3.3 is higher than 10.100.1.1, the mapping agent elects R3 as the RP for 226.26.26.26. R1 remains the RP for the 225.25.25.25 group.

R1 sends notification of the winners of the group-to-RP elections to the 224.0.1.40 RP Discovery group. All multicast routers running PIM-SM or sparse-dense mode have been implicitly subscribed to this group and install this information in their multicast routing tables.

During this time, the PIM protocol running on R3 has logged the following output. Analyze the Auto-RP election from R3's perspective.

```
R3#
*Nov  7 16:26:48.763: Auto-RP(0): Received RP-discovery, from 10.100.1.1,
RP_cnt 1, ht 181
*Nov  7 16:26:48.763: Auto-RP(0): Added with (225.25.25.25/32, RP:10.100.1.1),
PIMv2 v1
*Nov  7 16:26:48.763: Auto-RP(0): Added with (226.26.26.26/32, RP:10.100.1.1),
PIMv2 v1
R3#
*Nov  7 16:27:16.863: Auto-RP(0): Build RP-Announce for 10.100.3.3, PIMv2/v1,
ttl 3, ht 181
*Nov  7 16:27:16.863: Auto-RP(0): Build announce entry for (226.26.26.26/32)
*Nov  7 16:27:16.863: Auto-RP(0): Send RP-Announce packet on FastEthernet0/0
*Nov  7 16:27:16.863: Auto-RP(0): Send RP-Announce packet on Serial0/0/0
*Nov  7 16:27:16.863: Auto-RP(0): Send RP-Announce packet on Serial0/0/1
*Nov  7 16:27:16.863: Auto-RP: Send RP-Announce packet on Loopback3
*Nov  7 16:27:16.871: Auto-RP(0): Received RP-discovery, from 10.100.1.1,
RP_cnt 2, ht 181
*Nov  7 16:27:16.871: Auto-RP(0): Update (225.25.25.25/32, RP:10.100.1.1),
PIMv2 v1
*Nov  7 16:27:16.871: Auto-RP(0): Added with (226.26.26.26/32, RP:10.100.3.3),
PIMv2 v1
```

Notice that when R1 floods the first RP discovery packet, the mapping agent has not yet received an Auto-RP announcement packet from R3. After R3 sends the announcement, the mapping agent compares the IP addresses of the two candidates for the 226.26.26.26 network and elects R3 as the RP. R1 then floods the new group-to-RP mappings to all routers within its scope via multicast.

Issue the **undebg all** command on R1 and R3 to stop debugging PIM Auto-RP events.

```
R1# undebg all
```

```
R3# undebg all
```

Step 5: Verify the RP Mappings

Use the **show ip pim rp** command on the Auto-RP routers to view group-to-RP mappings.

```
R1# show ip pim rp
Group: 226.26.26.26, RP: 10.100.3.3, v2, v1, uptime 00:53:51, expires 00:02:03
```

```
Group: 225.25.25.25, RP: 10.100.1.1, v2, v1, next RP-reachable in 00:01:10
```

```
R3# show ip pim rp
```

```
Group: 226.26.26.26, RP: 10.100.3.3, v2, v1, next RP-reachable in 00:01:17  
Group: 225.25.25.25, RP: 10.100.1.1, v2, v1, uptime 00:54:40, expires 00:02:23
```

For a full view of the group-to-RP mappings, issue the **show ip pim rp mapping** command on all routers. This command discloses how the RP was elected and which router performed the mapping. It is extremely useful in debugging Auto-RP elections.

```
R1# show ip pim rp mapping
```

```
PIM Group-to-RP Mappings  
This system is an RP (Auto-RP)  
This system is an RP-mapping agent (Loopback1)
```

```
Group(s) 225.25.25.25/32  
  RP 10.100.1.1 (?), v2v1  
    Info source: 10.100.1.1 (?), elected via Auto-RP  
    Uptime: 00:54:25, expires: 00:02:34  
Group(s) 226.26.26.26/32  
  RP 10.100.3.3 (?), v2v1  
    Info source: 10.100.3.3 (?), elected via Auto-RP  
    Uptime: 00:53:57, expires: 00:01:58  
  RP 10.100.1.1 (?), v2v1  
    Info source: 10.100.1.1 (?), via Auto-RP  
    Uptime: 00:54:25, expires: 00:02:32
```

```
R2# show ip pim rp mapping
```

```
PIM Group-to-RP Mappings  
  
Group(s) 225.25.25.25/32  
  RP 10.100.1.1 (?), v2v1  
    Info source: 10.100.1.1 (?), elected via Auto-RP  
    Uptime: 00:58:36, expires: 00:02:24  
Group(s) 226.26.26.26/32  
  RP 10.100.3.3 (?), v2v1  
    Info source: 10.100.1.1 (?), elected via Auto-RP  
    Uptime: 00:54:06, expires: 00:02:27
```

```
R3# show ip pim rp mapping
```

```
PIM Group-to-RP Mappings  
This system is an RP (Auto-RP)  
  
Group(s) 225.25.25.25/32  
  RP 10.100.1.1 (?), v2v1  
    Info source: 10.100.1.1 (?), elected via Auto-RP  
    Uptime: 00:54:44, expires: 00:02:20  
Group(s) 226.26.26.26/32  
  RP 10.100.3.3 (?), v2v1  
    Info source: 10.100.1.1 (?), elected via Auto-RP  
    Uptime: 00:54:16, expires: 00:02:15
```

How did each router learn these mappings?

Each router listens to the 224.0.1.40 group, to which Auto-RP mapping agents send the group-to-RP mappings. The mappings are then entered into each router's mapping tables.

Because you configured R1 as a mapping agent, PIM routers that are not elected as the Auto-RP track the number of RP discovery messages received. Verify this with the **show ip pim autorp** command on R2.

```
R2# show ip pim autorp
AutoRP Information:
  AutoRP is enabled.

PIM AutoRP Statistics: Sent/Received
  RP Announce: 0/0, RP Discovery: 0/96
```

Step 6: Verify Multicast Operation

Issue multicast pings to generate the (S, G) state in the multicast network. Use a repeat count of 100 to generate a stream of multicast packets flowing through the network.

```
SW1# ping
Protocol [ip]:
Target IP address: 225.25.25.25
Repeat count [1]: 100
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 225.25.25.25, timeout is 2 seconds:

Reply to request 0 from 172.16.20.2, 8 ms
Reply to request 0 from 172.16.13.1, 36 ms
Reply to request 0 from 172.16.203.3, 36 ms
Reply to request 1 from 172.16.20.2, 4 ms
Reply to request 1 from 172.16.13.1, 28 ms
Reply to request 1 from 172.16.203.3, 28 ms
Reply to request 2 from 172.16.20.2, 8 ms
Reply to request 2 from 172.16.13.1, 32 ms
Reply to request 2 from 172.16.203.3, 32 ms
...
```

Display the multicast routing entry for 225.25.25.25 on each router with the **show ip mroute group-address** command.

```
R1# show ip mroute 225.25.25.25
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 225.25.25.25), 02:39:52/00:03:20, RP 10.100.1.1, flags: SJCL
  Incoming interface: Null, RPF nbr 0.0.0.0
```

```

Outgoing interface list:
  FastEthernet0/0, Forward/Sparse-Dense, 00:18:52/00:03:20
  Loopback1, Forward/Sparse-Dense, 02:39:52/00:02:15

(10.100.20.4, 225.25.25.25), 00:03:14/00:02:59, flags: LT
  Incoming interface: FastEthernet0/0, RPF nbr 10.100.13.3
  Outgoing interface list:
    Loopback1, Forward/Sparse-Dense, 00:03:15/00:02:14

R2# show ip mroute 225.25.25.25
<output omitted>

(*, 225.25.25.25), 00:02:36/stopped, RP 10.100.1.1, flags: SJPLF
  Incoming interface: Serial0/0/1, RPF nbr 10.100.203.3
  Outgoing interface list: Null

(10.100.20.4, 225.25.25.25), 00:02:36/00:03:28, flags: LFT
  Incoming interface: FastEthernet0/0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0/0/1, Forward/Sparse-Dense, 00:02:36/00:02:52

R3# show ip mroute 225.25.25.25
<output omitted>

(*, 225.25.25.25), 00:19:07/00:02:59, RP 10.100.1.1, flags: SJCL
  Incoming interface: FastEthernet0/0, RPF nbr 10.100.13.1
  Outgoing interface list:
    Serial0/0/1, Forward/Sparse-Dense, 00:02:29/00:02:59
    Loopback3, Forward/Sparse-Dense, 00:19:05/00:02:02

(10.100.20.4, 225.25.25.25), 00:03:27/00:03:27, flags: LT
  Incoming interface: Serial0/0/1, RPF nbr 10.100.203.2
  Outgoing interface list:
    FastEthernet0/0, Forward/Sparse-Dense, 00:03:28/00:03:11
    Loopback3, Forward/Sparse-Dense, 00:03:28/00:02:01

```

Which PIM mode is the 225.25.25.25 group using on the routers based on the flags in the output shown above and why?

PIM is operating in sparse mode for the 225.25.25.25 group.

To which router will R2 send PIMv2 Register messages for the 225.25.25.25 group?

R2 will send PIM Register messages to 10.100.1.1, the RP for the 225.25.25.25 group.

To which router will R2 send PIMv2 Register messages for the 226.26.26.26 group?

R2 will send PIM Register messages for 226.26.26.26 to 10.100.3.3, the RP for the 226.26.26.26 group.

Step 7: Explore Auto-RP Operation with Sparse-Dense Mode

Enable Auto-RP debugging on R1 with the **debug ip pim auto-rp** command. Shut down the Loopback3 interface on R3.

```
R1# debug ip pim auto-rp

R3(config)# interface loopback 3
R3(config-if)# shutdown
```

How will this affect the Auto-RP election?

If the current RP for the 226.26.26.26 group becomes unreachable, then another RP must be elected via Auto-RP, or the group will fall back into dense mode. In this case, the group will elect R1's loopback interface as the new RP for the 226.26.26.26 group.. Until the RP discovery information on R2 times out, R2 will continue sending PIM Register messages to R1.

Issue the repeated multicast ping to generate (S, G) state on your routers before R1 takes over as the RP for the 226.26.26.26 group.

```
SW1# ping
Protocol [ip]:
Target IP address: 226.26.26.26
Repeat count [1]: 100
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 226.26.26.26, timeout is 2 seconds:
.....
Reply to request 60 from 10.100.13.1, 48 ms
Reply to request 61 from 10.100.13.1, 48 ms
Reply to request 62 from 10.100.13.1, 28 ms
Reply to request 63 from 10.100.13.1, 28 ms
Reply to request 64 from 10.100.13.1, 28 ms
Reply to request 65 from 10.100.13.1, 32 ms
Reply to request 66 from 10.100.13.1, 28 ms
Reply to request 67 from 10.100.13.1, 28 ms
Reply to request 68 from 10.100.13.1, 28 ms
Reply to request 69 from 10.100.13.1, 28 ms
... continues through request 99 ...
```

Why were so many pings dropped before reaching the IGMP-subscribed loopback interfaces on R1 and R3?

The RP information must timeout on the mapping agent, and the new RP must be elected. All routers that have been informed that R3's loopback interface is the RP for 226.26.26.26 must wait for their information to timeout before

sending the multicasts in dense mode. The number of pings indicates the length of time until the timeout period for the RP information is exceeded.

How does this affect multicast routing?

Multicast routing will be ineffective for the 226.26.26.26 group while the RP for that group is unreachable. PIM Register messages will not be sent, so no IGMP subscribers to that group will receive any data.

If Auto-RP debugging were enabled, you would see the following output on the mapping agent after three minutes:

```
*Nov 7 18:35:34.839: Auto-RP(0): Mapping (226.26.26.26/32, RP:10.100.3.3)
expired,
*Nov 7 18:35:34.843: Auto-RP(0): Build RP-Discovery packet
*Nov 7 18:35:34.843: Auto-RP: Build mapping (225.25.25.25/32,
RP:10.100.1.1), PIMv2 v1,
*Nov 7 18:35:34.843: Auto-RP: Build mapping (226.26.26.26/32,
RP:10.100.1.1), PIMv2 v1.
*Nov 7 18:35:34.843: Auto-RP(0): Send RP-discovery packet on FastEthernet0/0
(1 RP entries)
*Nov 7 18:35:34.843: Auto-RP(0): Send RP-discovery packet on Serial0/0/0 (1
RP entries)
*Nov 7 18:35:34.843: Auto-RP(0): Send RP-discovery packet on Serial0/0/1 (1
RP entries)
*Nov 7 18:35:34.843: Auto-RP: Send RP-discovery packet on Loopback1 (1 RP
entries)
```

Explain the preceding output.

The first message indicates that since no RP announcements have been received for the length of the timeout period, the mapping agent has released R3 as the RP for the 226.26.26.26 group. The mapping agent elects R1 as the new RP for the group and sends out the RP discovery message to all PIM routers.

You can check the resulting group-to-RP mappings by using the **show ip pim rp mapping** command.

```
R1# show ip pim rp mapping
PIM Group-to-RP Mappings
This system is an RP (Auto-RP)
This system is an RP-mapping agent (Loopback1)

Group(s) 225.25.25.25/32
  RP 10.100.1.1 (?), v2v1
  Info source: 10.100.1.1 (?), elected via Auto-RP
  Uptime: 02:53:45, expires: 00:02:12
Group(s) 226.26.26.26/32
```

```
RP 10.100.1.1 (?), v2v1
  Info source: 10.100.1.1 (?), elected via Auto-RP
  Uptime: 02:53:45, expires: 00:02:12
```

Auto-RP allows you to configure backup RPs in a sparse mode or sparse-dense mode network. If you configured a static RP and later it became unreachable, receivers would no longer be able to use the shared tree to receive data. Auto-RP provides a layer of redundancy in sparse mode networks by using mapping agents to delegate RP roles to RP candidates.

Step 8: Verify the Operation of Dense-Mode Fallback

How will the status of the RPs in the network change if you shut down the Loopback1 interface on R1?

The RPs for both groups will be unreachable. After three minutes, the PIM routers will be informed that there is no RP for either group and they will fall back to dense mode.

Will the PIM mappings in the network change their behavior immediately or after a short time? Which mode is PIM now operating in for the active groups?

After three minutes, the PIM routers will be informed that there is no RP for either group and they will fall back to dense mode.

Verify your answers with the **show ip pim rp mapping** and **show ip mroute summary** commands on R2 after the mappings expire.

```
R1# conf t
R1(config)# interface loopback 1
R1(config-if)# shutdown

R2# show ip pim rp mapping
PIM Group-to-RP Mappings

Group(s) 225.25.25.25/32
  RP 10.100.1.1 (?), v2v1
  Info source: 10.100.1.1 (?), elected via Auto-RP
  Uptime: 03:06:13, expires: 00:00:18
Group(s) 226.26.26.26/32
  RP 10.100.1.1 (?), v2v1
  Info source: 10.100.1.1 (?), elected via Auto-RP
  Uptime: 00:08:38, expires: 00:00:18

SW1# ping
Protocol [ip]:
Target IP address: 225.25.25.25
Repeat count [1]: 100
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]:
```

```

Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 225.25.25.25, timeout is 2 seconds:
.....
Reply to request 61 from 10.100.20.2, 8 ms
Reply to request 62 from 10.100.20.2, 4 ms
Reply to request 63 from 10.100.20.2, 4 ms
Reply to request 64 from 10.100.20.2, 4 ms
Reply to request 65 from 10.100.20.2, 4 ms
Reply to request 66 from 10.100.20.2, 4 ms
Reply to request 67 from 10.100.20.2, 4 ms
... continues through request 99 ...

```

```
R2# show ip pim rp mapping
```

```
PIM Group-to-RP Mappings
```

```
R2# show ip mroute
```

```
IP Multicast Routing Table
```

```

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group

```

```
Outgoing interface flags: H - Hardware switched, A - Assert winner
```

```
Timers: Uptime/Expires
```

```
Interface state: Interface, Next-Hop or VCD, State/Mode
```

```
(* , 225.25.25.25), 00:05:09/stopped, RP 0.0.0.0, flags: DL
```

```
Incoming interface: Null, RPF nbr 0.0.0.0
```

```
Outgoing interface list:
```

```
Serial0/0/1, Forward/Sparse-Dense, 00:05:09/00:00:00
```

```
Serial0/0/0, Forward/Sparse-Dense, 00:05:09/00:00:00
```

```
(10.100.20.4, 225.25.25.25), 00:00:06/00:02:57, flags: PLT
```

```
Incoming interface: FastEthernet0/0, RPF nbr 0.0.0.0
```

```
Outgoing interface list:
```

```
Serial0/0/0, Prune/Sparse-Dense, 00:00:07/00:02:55, A
```

```
Serial0/0/1, Prune/Sparse-Dense, 00:00:06/00:02:53
```

```
(* , 224.0.1.40), 00:16:56/00:02:38, RP 0.0.0.0, flags: DCL
```

```
Incoming interface: Null, RPF nbr 0.0.0.0
```

```
Outgoing interface list:
```

```
Serial0/0/1, Forward/Sparse-Dense, 00:16:56/00:00:00
```

```
Serial0/0/0, Forward/Sparse-Dense, 00:16:56/00:00:00
```

```
FastEthernet0/0, Forward/Sparse-Dense, 00:16:56/00:00:00
```

When the group-to-RP mappings expire on R2, PIM realizes that there is no longer an RP for the 225.25.25.25 and 226.26.26.26 groups and converts those groups to use dense mode operation. When this occurs, PIM floods multicast data out all interfaces and then prunes back to R2 because there are no other receivers of the 225.25.25.25 group.

Ping from SW1 again and then enable the loopback interfaces you shut down on R1 and R3.

```
SW1# ping
```

```
Protocol [ip]:
```

```
Target IP address: 225.25.25.25
```

```
Repeat count [1]: 100
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 225.25.25.25, timeout is 2 seconds:
```

```
Reply to request 0 from 10.100.20.2, 8 ms
Reply to request 1 from 10.100.20.2, 4 ms
Reply to request 2 from 10.100.20.2, 4 ms
Reply to request 3 from 10.100.20.2, 8 ms
Reply to request 4 from 10.100.20.2, 8 ms
Reply to request 5 from 10.100.20.2, 4 ms
Reply to request 6 from 10.100.20.2, 4 ms
Reply to request 7 from 10.100.20.2, 4 ms
Reply to request 8 from 10.100.20.2, 4 ms
Reply to request 9 from 10.100.20.2, 4 ms
Reply to request 10 from 10.100.20.2, 4 ms
Reply to request 11 from 10.100.20.2, 4 ms
Reply to request 12 from 10.100.20.2, 8 ms
Reply to request 12 from 10.100.13.1, 32 ms
Reply to request 13 from 10.100.20.2, 12 ms
Reply to request 13 from 10.100.13.1, 28 ms
Reply to request 14 from 10.100.20.2, 4 ms
Reply to request 14 from 10.100.13.1, 28 ms
Reply to request 15 from 10.100.20.2, 4 ms
Reply to request 15 from 10.100.13.1, 28 ms
Reply to request 16 from 10.100.20.2, 4 ms
Reply to request 16 from 10.100.13.1, 44 ms
Reply to request 16 from 10.100.203.3, 28 ms
... continued below ...
```

Over the period highlighted, both loopback interfaces were opened and PIM-DM flooded traffic to them. No RPs for these groups were sent to R2 by the mapping agent, so it continues to employ PIM-DM flood-and-prune behavior.

```
... continued from above ...
Reply to request 17 from 10.100.20.2, 4 ms
Reply to request 17 from 10.100.13.1, 44 ms
Reply to request 17 from 10.100.203.3, 28 ms
Reply to request 18 from 10.100.20.2, 8 ms
Reply to request 18 from 10.100.13.1, 44 ms
Reply to request 18 from 10.100.203.3, 28 ms
Reply to request 19 from 10.100.20.2, 8 ms
Reply to request 19 from 10.100.13.1, 44 ms
Reply to request 19 from 10.100.203.3, 32 ms
Reply to request 20 from 10.100.20.2, 8 ms
Reply to request 20 from 10.100.13.1, 44 ms
Reply to request 20 from 10.100.203.3, 32 ms
Reply to request 21 from 10.100.20.2, 8 ms
Reply to request 21 from 10.100.13.1, 44 ms
Reply to request 21 from 10.100.203.3, 32 ms
Reply to request 22 from 10.100.20.2, 8 ms
Reply to request 22 from 10.100.13.1, 44 ms
Reply to request 22 from 10.100.203.3, 28 ms
Reply to request 23 from 10.100.20.2, 8 ms
Reply to request 23 from 10.100.13.1, 48 ms
Reply to request 23 from 10.100.203.3, 32 ms
Reply to request 24 from 10.100.20.2, 4 ms
Reply to request 24 from 10.100.13.1, 44 ms
Reply to request 24 from 10.100.203.3, 28 ms
```

```
Reply to request 25 from 10.100.20.2, 4 ms
Reply to request 25 from 10.100.13.1, 44 ms
Reply to request 25 from 10.100.203.3, 28 ms
Reply to request 26 from 10.100.20.2, 8 ms
Reply to request 26 from 10.100.13.1, 44 ms
Reply to request 26 from 10.100.203.3, 28 ms
Reply to request 27 from 10.100.20.2, 4 ms
Reply to request 27 from 10.100.13.1, 44 ms
Reply to request 27 from 10.100.203.3, 28 ms
Reply to request 28 from 10.100.20.2, 4 ms
Reply to request 28 from 10.100.13.1, 44 ms
Reply to request 28 from 10.100.203.3, 28 ms
Reply to request 29 from 10.100.20.2, 4 ms
Reply to request 29 from 10.100.13.1, 44 ms
Reply to request 29 from 10.100.203.3, 32 ms
Reply to request 30 from 10.100.20.2, 4 ms
Reply to request 31 from 10.100.20.2, 8 ms
Reply to request 32 from 10.100.20.2, 4 ms
Reply to request 33 from 10.100.20.2, 4 ms
Reply to request 34 from 10.100.20.2, 4 ms
... continued below ...
```

R2 hears R1's Auto-RP announcement to the 224.0.1.240 group. Because there are now RPs in the network, R2 converts the 225.25.25.25 and 226.26.26.26 state to use PIM-SM. R2 does not begin forwarding data to the RP because the mapping agent has not yet elected an RP for the 225.25.25.25 group. R2 waits to receive the RP address from the mapping agent before it begins encapsulating multicast data as unicasts to R1, the RP.

```
... continued from above ...
Reply to request 35 from 10.100.20.2, 4 ms
Reply to request 36 from 10.100.20.2, 8 ms
Reply to request 37 from 10.100.20.2, 4 ms
Reply to request 38 from 10.100.20.2, 4 ms
Reply to request 39 from 10.100.20.2, 4 ms
Reply to request 40 from 10.100.20.2, 4 ms
Reply to request 41 from 10.100.20.2, 8 ms
Reply to request 42 from 10.100.20.2, 4 ms
Reply to request 43 from 10.100.20.2, 4 ms
Reply to request 44 from 10.100.20.2, 8 ms
Reply to request 45 from 10.100.20.2, 8 ms
Reply to request 46 from 10.100.20.2, 8 ms
Reply to request 47 from 10.100.20.2, 8 ms
Reply to request 48 from 10.100.20.2, 4 ms
Reply to request 49 from 10.100.20.2, 4 ms
Reply to request 50 from 10.100.20.2, 4 ms
Reply to request 51 from 10.100.20.2, 8 ms
Reply to request 52 from 10.100.20.2, 4 ms
Reply to request 53 from 10.100.20.2, 4 ms
Reply to request 54 from 10.100.20.2, 4 ms
Reply to request 55 from 10.100.20.2, 4 ms
Reply to request 56 from 10.100.20.2, 8 ms
Reply to request 57 from 10.100.20.2, 4 ms
Reply to request 58 from 10.100.20.2, 4 ms
Reply to request 59 from 10.100.20.2, 8 ms
Reply to request 59 from 10.100.13.1, 44 ms
Reply to request 59 from 10.100.203.3, 32 ms
Reply to request 60 from 10.100.20.2, 8 ms
Reply to request 60 from 10.100.13.1, 44 ms
Reply to request 60 from 10.100.203.3, 32 ms
```

Request 59 is the first packet that R2 encapsulates in a unicast packet and sends to the RP. R1, the RP, forwards the multicast traffic down the shared tree to any subscribers of the 225.25.25.25 group.

When R2 converts a group to a different mode, multicast packets are inevitably dropped before reaching receivers. Although sparse-dense mode brings a high level of resiliency compared to sparse mode, packets are still lost during the transition to and recovery from dense-mode fallback.

This lab demonstrates two basic features of resiliency available on Cisco routers; however, there are more advanced ways to deploy high-availability solutions in a multicast network.

The first three labs on multicast provided the foundation to understand sparse-dense mode. Explain how the concepts in those first three labs lead to an in-depth understanding of PIM sparse-dense mode.

Sparse-dense mode is a hybrid mode in which the mode for groups defaults to dense mode if an RP cannot be found.

IGMP allows hosts and routers to subscribe to multicast groups. IGMP snooping prevents multicasts at Layer 2 from acting like broadcasts.

PIM-DM provides a simplistic basis for running multicast, but with high bandwidth and memory overhead.

PIM-SM provides mechanisms to create a multicast network with low bandwidth and memory overhead.

PIM-SDM creates a resilient network that implements all of the resource-saving features of PIM-SM with the resiliency of a PIM-DM backup.

Appendix A: TCL Script Output

```
R1# tclsh
R1(tcl)#foreach address {
+>(tcl)#10.100.1.1
+>(tcl)#10.100.13.1
+>(tcl)#10.100.102.1
+>(tcl)#10.100.103.1
+>(tcl)#10.100.2.2
+>(tcl)#10.100.20.2
+>(tcl)#10.100.102.2
+>(tcl)#10.100.203.2
+>(tcl)#10.100.3.3
+>(tcl)#10.100.13.3
+>(tcl)#10.100.103.3
+>(tcl)#10.100.203.3
+>(tcl)#10.100.20.4
+>(tcl)#} { ping $address }
```

```

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.100.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.100.13.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.100.102.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.100.103.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.100.2.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/15/16 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.100.20.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/15/16 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.100.102.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.100.203.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/14/16 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.100.3.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.100.13.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.100.103.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.100.203.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.100.20.4, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/16/20 ms
R1(tcl)# tclquit

```

```

R2# tclsh
R2(tcl)#foreach address {
+>(tcl)#10.100.1.1
+>(tcl)#10.100.13.1
+>(tcl)#10.100.102.1
+>(tcl)#10.100.103.1
+>(tcl)#10.100.2.2
+>(tcl)#10.100.20.2
+>(tcl)#10.100.102.2

```

```
+>(tcl)#10.100.203.2
+>(tcl)#10.100.3.3
+>(tcl)#10.100.13.3
+>(tcl)#10.100.103.3
+>(tcl)#10.100.203.3
+>(tcl)#10.100.20.4
+>(tcl)#} { ping $address }
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.100.1.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 12/15/16 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.100.13.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 12/15/16 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.100.102.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.100.103.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.100.2.2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.100.20.2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.100.102.2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/64 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.100.203.2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/36 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.100.3.3, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 12/14/16 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.100.13.3, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 12/14/16 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.100.103.3, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 12/15/16 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.100.203.3, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 12/15/16 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.100.20.4, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

R2(tcl)# **tclquit**

R3# **tclsh**

```
R3(tcl)#foreach address {
+>(tcl)#10.100.1.1
+>(tcl)#10.100.13.1
+>(tcl)#10.100.102.1
+>(tcl)#10.100.103.1
+>(tcl)#10.100.2.2
+>(tcl)#10.100.20.2
+>(tcl)#10.100.102.2
+>(tcl)#10.100.203.2
+>(tcl)#10.100.3.3
+>(tcl)#10.100.13.3
+>(tcl)#10.100.103.3
+>(tcl)#10.100.203.3
+>(tcl)#10.100.20.4
+>(tcl)#} { ping $address }
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.100.1.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.100.13.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.100.102.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.100.103.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.100.2.2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 12/15/16 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.100.20.2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 12/15/16 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.100.102.2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 20/20/24 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.100.203.2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 12/13/16 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.100.3.3, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.100.13.3, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.100.103.3, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/64 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.100.203.3, timeout is 2 seconds:

!!!!

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/36 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.100.20.4, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/16/16 ms
R3(tcl)# tclquit
```

Final Configurations

```
R1# show run
!
hostname R1
!
ip multicast-routing
!
interface Loopback1
 ip address 10.100.1.1 255.255.255.0
 ip pim sparse-dense-mode
 ip igmp join-group 225.25.25.25
 ip igmp join-group 226.26.26.26
!
interface FastEthernet0/0
 ip address 10.100.13.1 255.255.255.0
 ip pim sparse-dense-mode
 no shutdown
!
interface Serial0/0/0
 bandwidth 64
 ip address 10.100.102.1 255.255.255.248
 ip pim sparse-dense-mode
 clock rate 64000
 no shutdown
!
interface Serial0/0/1
 bandwidth 64
 ip address 10.100.103.1 255.255.255.248
 ip pim sparse-dense-mode
 no shutdown
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 0
!
ip pim send-rp-announce Loopback1 scope 3 group-list 1
ip pim send-rp-discovery Loopback1 scope 3
!
access-list 1 permit 225.25.25.25
access-list 1 permit 226.26.26.26
!
end

R2# show run
!
hostname R2
!
ip multicast-routing
!
interface Loopback2
 ip address 10.100.2.2 255.255.255.0
 ip igmp join-group 225.25.25.25
!
interface FastEthernet0/0
 ip address 10.100.20.2 255.255.255.0
 ip pim sparse-dense-mode
```

```

    no shutdown
    !
interface Serial0/0/0
    bandwidth 64
    ip address 10.100.102.2 255.255.255.248
    ip pim sparse-dense-mode
    no shutdown
    !
interface Serial0/0/1
    bandwidth 128
    ip address 10.100.203.2 255.255.255.248
    ip pim sparse-dense-mode
    clock rate 64000
    no shutdown
    !
router ospf 1
    network 10.0.0.0 0.255.255.255 area 0
    !
end

R3# show run
!
hostname R3
!
ip multicast-routing
!
interface Loopback3
    ip address 10.100.3.3 255.255.255.0
    ip pim sparse-dense-mode
    ip igmp join-group 225.25.25.25
    ip igmp join-group 226.26.26.26
    !
interface FastEthernet0/0
    ip address 10.100.13.3 255.255.255.0
    ip pim sparse-dense-mode
    no shutdown
    !
interface Serial0/0/0
    bandwidth 64
    ip address 10.100.103.3 255.255.255.248
    ip pim sparse-dense-mode
    clock rate 64000
    no shutdown
    !
interface Serial0/0/1
    bandwidth 128
    ip address 10.100.203.3 255.255.255.248
    ip pim sparse-dense-mode
    no shutdown
    !
router ospf 1
    network 10.0.0.0 0.255.255.255 area 0
    !
ip pim send-rp-announce Loopback3 scope 3 group-list 3
!
access-list 3 permit 226.26.26.26
!
end

SW1# show run
!
hostname SW1
!

```

```
interface FastEthernet0/1
  switchport access vlan 13
  switchport mode access
!
interface FastEthernet0/3
  switchport access vlan 20
  switchport mode access
!
interface FastEthernet0/5
  switchport access vlan 13
  switchport mode access
!
interface Vlan20
  ip address 10.100.20.4 255.255.255.0
  no shutdown
!
ip default-gateway 10.100.20.2
!
end
```