CHAPTER

Multicast 16

What You Will Learn

In this chapter, you will learn how multicast routing protocols allow multicast traffic to make its way from a source to interested receivers through a router-based network. We'll look at both dense and parse multicast routing protocols, as well as some of the other protocols used with them (such as IGMP).

You will learn how the PIM rendezvous point (RP) has become the key component in a multicast network. We'll see how to configure an RP on the network and use it to deliver a simple multicast traffic stream to hosts.

If the Internet and TCP/IP are going to be used for everything from the usual data activities to voice and video, something must be done about the normal *unicast* packet addressing reflecting one specific source and one specific destination. Almost everything described in this book so far has featured unicast, although multicast addresses have been mentioned from time to time—especially when used by routing protocols.

The one-to-many operation of multicast is a technique between the one-to-one packet delivery operation of unicast and the one-to-all operation of broadcast. Broadcasts tend to disrupt hosts' normal processing because most broadcasts are not really intended for every host yet each receiving host must pay attention to the broadcast packet's content. Many protocols that routinely used broadcasts, such as RIPv1, were replaced by versions that used multicast groups instead (RIPv2, OSPF). Even the protocols in IPv4 that still routinely use broadcast, such as ARPing to find the MAC address that goes with an IP address, have been replaced in IPv6 with multicast-friendly versions of the same procedure.

Multicast protocols are still not universally supported on much of the Internet. Then how do large numbers of people all watch the same video feed from a Web server (for example) at the same time? Today, this is normally accomplished with numerous unicast links, each running from the server to every individual host. This works, but it does not scale. Can a server handle 100, 1000, or 1,000,000 simultaneous users? Many-to-many multicast applications, such as on-line gaming and gambling sites, use

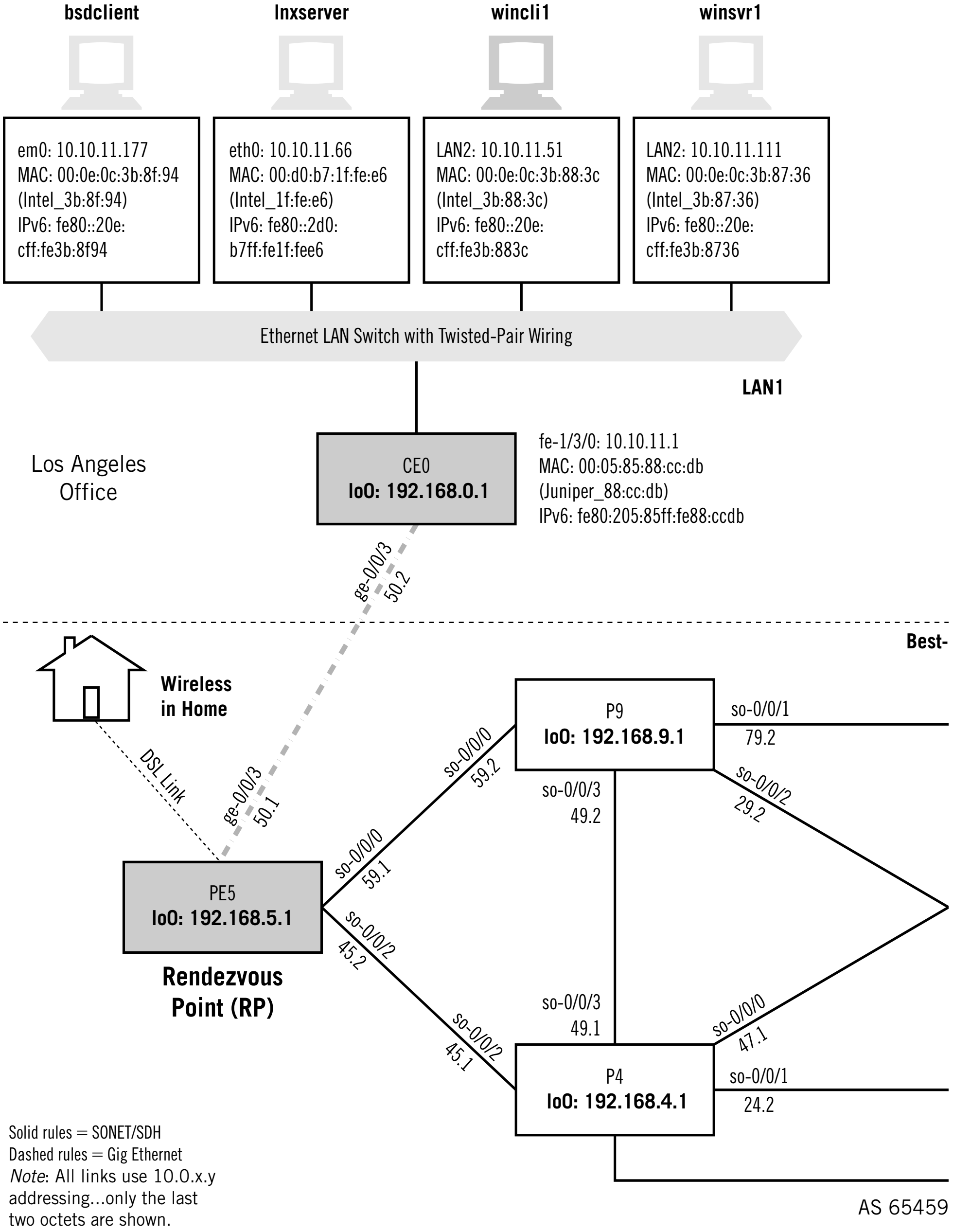
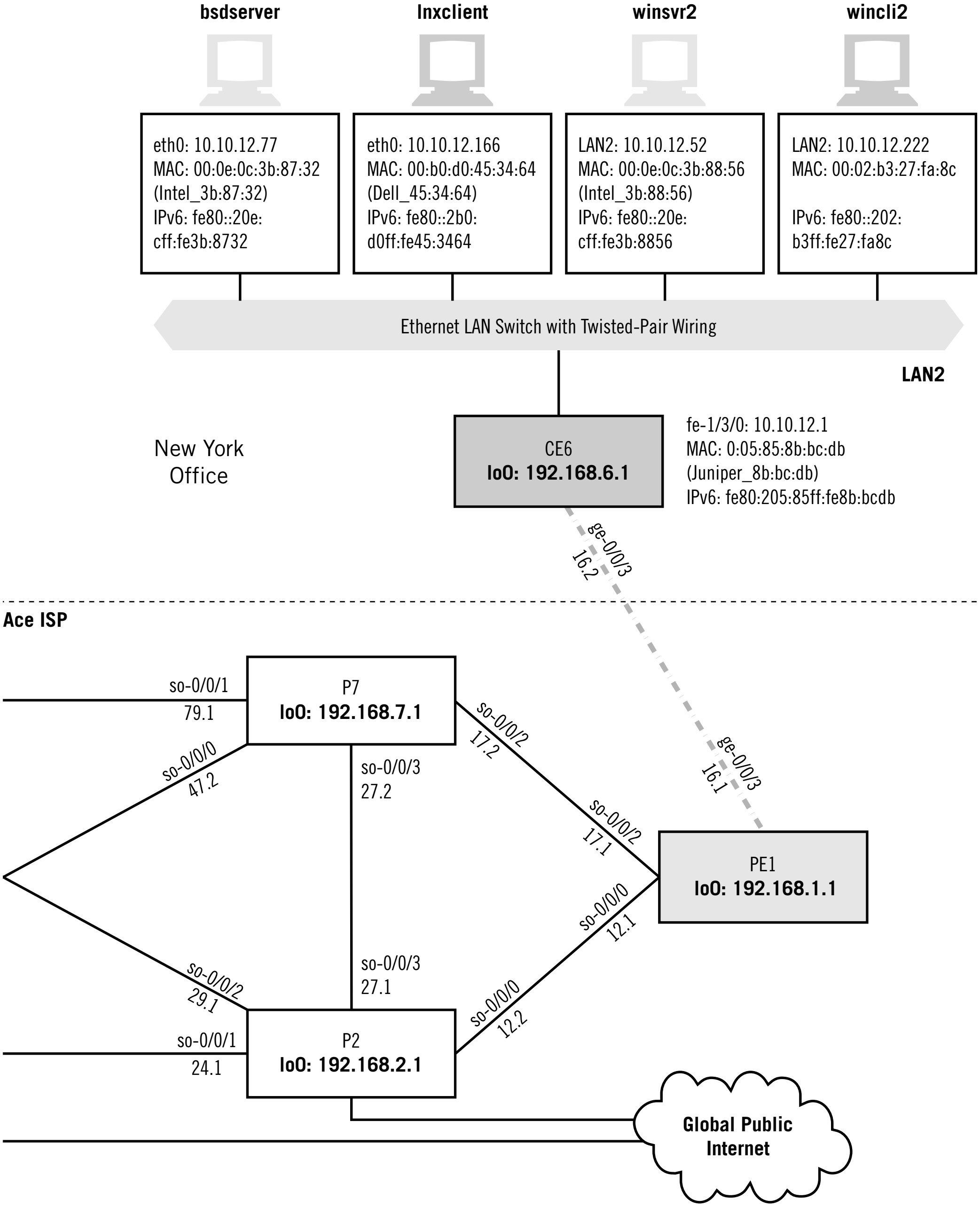


FIGURE 16.1

Portion of the Illustrated Network used for the multicast examples. The RP will be router PE5, and the ISPs have merged into a single AS for this chapter.


bsdserver
lnxclient
winsvr2
wincli2
eth0: 10.10.12.77
MAC: 00:0e:0c:3b:87:32
(Intel_3b:87:32)
IPv6: fe80::20e:
cff:fe3b:8732
eth0: 10.10.12.166
MAC: 00:b0:d0:45:34:64
(Dell_45:34:64)
IPv6: fe80::2b0:
d0ff:fe45:3464
LAN2: 10.10.12.52
MAC: 00:0e:0c:3b:88:56
(Intel_3b:88:56)
IPv6: fe80::20e:
cff:fe3b:8856
LAN2: 10.10.12.222
MAC: 00:02:b3:27:fa:8c
IPv6: fe80::202:
b3ff:fe27:fa8c
Ethernet LAN Switch with Twisted-Pair Wiring
LAN2
New York
Office
CE6
lo0: 192.168.6.1
fe-1/3/0: 10.10.12.1
MAC: 0:05:85:8b:bc:db
(Juniper_8b:bc:db)
IPv6: fe80:205:85ff:fe8b:bcdb
ge-0/0/3
16.2
Ace ISP
so-0/0/1
79.1
P7
lo0: 192.168.7.1
so-0/0/2
17.2
ge-0/0/3
16.1
so-0/0/0
47.2
so-0/0/3
27.2
so-0/0/2
17.1
PE1
lo0: 192.168.1.1
so-0/0/0
12.1
so-0/0/2
29.1
so-0/0/3
27.1
so-0/0/0
12.2
so-0/0/1
24.1
P2
lo0: 192.168.2.1
Global Public
Internet

multiple point-to-point meshes of links in most cases. Even if modern server clusters could do this, could all the routers and links handle this traffic? Multicast uses the *routers* to replicate packets, not the servers.

However, interdomain (or even intersubnet) multicasting is a problem. IP multicast is widely leveraged on localized subnets where it's solely a question of host support. Many-to-many applications have some fundamental scaling challenges and multicast does not address these very well. For example, how does each host in a shared tree of multicast traffic manage the receipt of perhaps 50 video streams from participants?

Today, multicast is a key component of local IPv6 and IPv4 resource discovery mechanisms and is not confined to enterprise applications. However, multicast *applications* are used mainly on enterprise networks not intended for the general public. In the future, multicast must move beyond a world where special routers (not all routers can handle multicast packets) use special parts of the Internet (most famously, the MBONE, or multicast backbone) to link interested hosts to their sources. Multicast must become an integral part of every piece of hardware and software on the Internet.

Let's look at a few simple multicast packets and frames on the Illustrated Network. We don't have any video cameras or music servers on the network to pump out content, but we do have the ability to use simple socket programs to generate a stream of packets to multicast group addresses as easily as to unicast destinations. We could look at multicast as used by OSPF or IPv6 router announcements, but we'll look at simple applications instead.

We'll look at IPv4 first, and then take a quick look at IPv6 multicasting. We'll use the devices shown in Figure 16.1 to illustrate multicast protocols, introducing the terms used in multicast protocols as we go. We'll explore all of the terms in detail later in the chapter.

This chapter uses `wincli2` and `lxnclient` on LAN2 and `wincli1` on LAN1. The router `PE5` will serve as our PIM sparse-mode RP. To simplify the number of multicast protocols used, we've merged the two ISPs into Best-Ace ISP for this chapter. This means we will not need to configure the Multicast Source Discovery Protocol (MSDP), which allows receivers in an AS to find RPs in another AS. A full investigation of MSDP is beyond the scope of this chapter, but we will go over the basics.

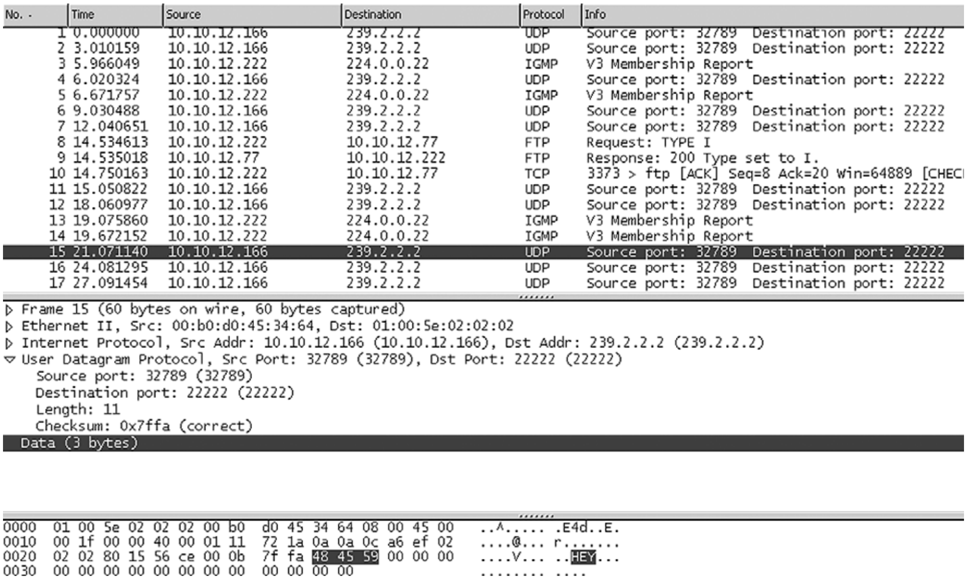
A FIRST LOOK AT IPv4 MULTICAST

This section uses two small socket programs from the source cited in Chapter 12: the excellent *TCP/IP Sockets in C* by Michael J. Donahoo and Kenneth L. Calvert. We'll use two programs run as MulticastReceiver and MulticastSender, and two free Windows multicast utilities, wsend and wlisten.

Let's start with two hosts on the same LAN. We'll use `lnxclient (10.10.12.166)` and `wincli2 (10.10.12.222)` for this exercise (both clients, but there's no heavy multicasting going on). We'll set the Linux client to multicast the text string `HEY` once every 3 seconds onto the LAN using multicast group address `239.2.2.2` (multicasts use special IP addresses for destinations) and UDP port `22222` (multicast applications often use UDP, and cannot use TCP). Naturally, we'll set the multicast receiver socket program on the Windows XP client to receive traffic sent to that group.

It should be noted that the multicast group addresses used here are *administratively scoped* addresses that should only reach a limited number of hosts and not be used on the global public Internet, much like private IP addresses. However, we won't discuss how the traffic to these groups is limited. This is mainly because there are some operational disagreements about how to apply administratively scoped boundaries. We are using scoped addresses primarily as an analogy for private IP addresses. We could also have used GLOP addresses (discussed in this chapter) or addresses from the dynamic multicast address block.

The receiver socket program does *not* generate any special messages to say, "Send me content addressed to group `239.2.2.2`." We know it's going to be there. Later, we'll see that a protocol called Internet Group Management Protocol (IGMP) sends join or leave messages and knows what content is carried at this time by group `239.2.2.2` because of the Session Announcement Protocol and Source Description Protocol (SAP/SDP) messages it receives. In reality, multicast is a *suite* of protocols—and much more is required to create a complete multicast *application*. However, this little send-and-receive exercise will still reveal a lot about multicast. Figure 16.2 shows a portion of the Ethereal capture of the packet stream, detailing the UDP content inside the IP packet.



```
No. -  Time         Source          Destination     Protocol  Info
     1 0.000000     10.10.12.166    239.2.2.2       UDP       Source port: 32789  Destination port: 22222
     2 3.010159     10.10.12.166    239.2.2.2       UDP       Source port: 32789  Destination port: 22222
     3 5.966049     10.10.12.222    224.0.0.22      IGMP      V3 Membership Report
     4 6.020324     10.10.12.166    239.2.2.2       UDP       Source port: 32789  Destination port: 22222
     5 6.671757     10.10.12.222    224.0.0.22      IGMP      V3 Membership Report
     6 9.030488     10.10.12.166    239.2.2.2       UDP       Source port: 32789  Destination port: 22222
     7 12.040651    10.10.12.166    239.2.2.2       UDP       Source port: 32789  Destination port: 22222
     8 14.534613    10.10.12.222    10.10.12.77     FTP       Request: TYPE I
     9 14.535018    10.10.12.77     10.10.12.222    FTP       Response: 200 Type set to I.
    10 14.750163    10.10.12.222    10.10.12.77     TCP       3373 > ftp [ACK] Seq=8 Ack=20 Win=64889 [CHEC
    11 15.050822    10.10.12.166    239.2.2.2       UDP       Source port: 32789  Destination port: 22222
    12 18.060977    10.10.12.166    239.2.2.2       UDP       Source port: 32789  Destination port: 22222
    13 19.075860    10.10.12.222    224.0.0.22      IGMP      V3 Membership Report
    14 19.672152    10.10.12.222    224.0.0.22      IGMP      V3 Membership Report
    15 21.071140    10.10.12.166    239.2.2.2       UDP       Source port: 32789  Destination port: 22222
    16 24.081295    10.10.12.166    239.2.2.2       UDP       Source port: 32789  Destination port: 22222
    17 27.091454    10.10.12.166    239.2.2.2       UDP       Source port: 32789  Destination port: 22222

▷ Frame 15 (60 bytes on wire, 60 bytes captured)
▷ Ethernet II, Src: 00:b0:d0:45:34:64, Dst: 01:00:5e:02:02:02
▷ Internet Protocol, Src Addr: 10.10.12.166 (10.10.12.166), Dst Addr: 239.2.2.2 (239.2.2.2)
▽ User Datagram Protocol, Src Port: 32789 (32789), Dst Port: 22222 (22222)
    Source port: 32789 (32789)
    Destination port: 22222 (22222)
    Length: 11
    Checksum: 0x7ffa (correct)
  Data (3 bytes)

0000  01 00 5e 02 02 02 00 b0  d0 45 34 64 08 00 45 00   ..^..... .E4d..E.
0010  00 1f 00 00 40 00 01 11  72 1a 0a 0a 0c a6 ef 02   ....@... r.......
0020  02 02 80 15 56 ce 00 0b  7f fa 48 45 59 00 00 00   ....V... ..HEY...
0030  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....
```

FIGURE 16.2

Multicast packet capture, showing the MAC address format used and the port in the UDP datagram. Some IGMPv3 messages appear also.

The Ethernet frame destination address is in a special form, starting with 01 and ending in 02:02:02—which corresponds to the 239.2.2.2 multicast group address. We'll explore the rules for determining this frame address in material following. Note that the packet is addressed to the entire group, not an individual host (as in unicast). How does the network know where to send replicated packets? Two strategies (discussed later in the chapter) are to send content everywhere and then stop if no one says they are listening (flood-and-prune, or dense mode), or to send content only to hosts that have indicated a desire to receive the content (sparse mode).

The figure also shows that the Windows XP receiver (10.10.12.222) is generating IGMPv3 membership reports sent to multicast group address 224.0.0.22 (the IGMP multicast group). XP does this to keep the multicast content coming, even though the socket sender program has no idea what it means. These messages from XP to the IGMP group sometimes cause consternation with Windows network administrators, who are not always familiar with multicast and wonder where the 224.0.0.22 "server" could be.

Now let's set our multicast group send program to span the router network from LAN1 to LAN2. We'll start the socket utility sending on wincli1 (10.10.11.51), using multicast group 239.1.1.1 and UDP port 11111. The listener will still be wincli2 (10.10.12.222).

This is easy enough, and Ethereal on wincli1 shows a steady stream of multicast traffic being dumped onto LAN1. However, the Ethereal capture on wincli2 (which had no problem receiving a multicast stream only moments ago) now receives absolutely nothing. What's wrong?

The problem is that the routers between LAN1 and LAN2 are not running a multicast routing protocol. The router on LAN1 at 10.10.11.1 adjacent to the source receives every multicast packet sent by wincli1. But the destination address of 239.1.1.1 is meaningless when considered as a unicast address. No entry exists in the unicast routing table, and there is yet no multicast "routing table" (more properly, table for multicast interface state) on the router network.

Before we configure multicast for use on our router network and allow multicast traffic to travel from LAN1 to LAN2, there are many new terms and protocols to explain—a few of which we've already mentioned (IGMP, SAP/SDP, how a multicast group maps to a frame destination address, and so on.) Let's start with the basics.

MULTICAST TERMINOLOGY

Multicast in TCP/IP has developed a reputation of being more difficult to understand than unicast. Part of the problem is the special terminology used with multicast, and the implication that if something is not universally supported, it must be complicated and difficult to understand. But there is nothing in multicast that is more complex than subnet masking, multicast sockets are nearly the same as unicast sockets (except that they don't use TCP sockets), and many things that routing protocols do with multicast packets are now employed in unicast as well (the reverse-path forwarding, or RFP

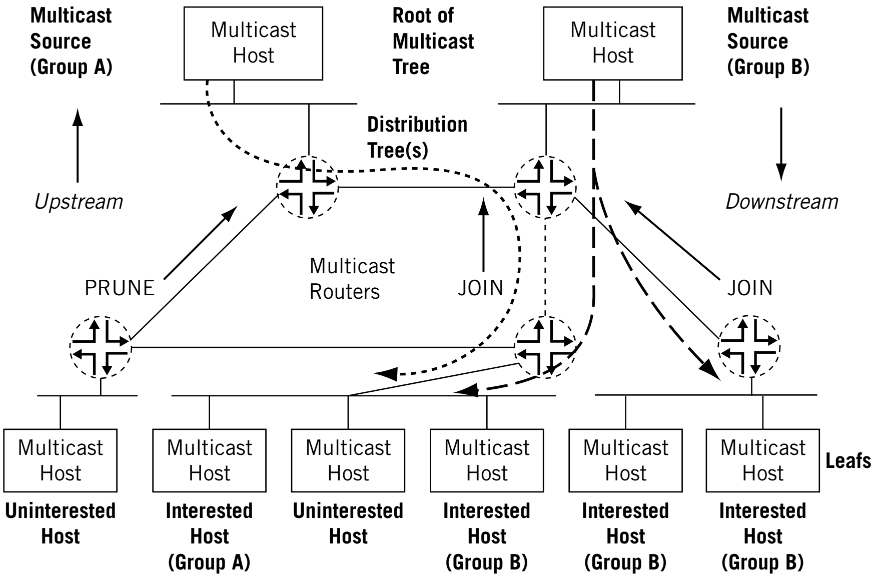


FIGURE 16.3

Examples of multicast terminology showing how multicast trees are "rooted" at the source. JOINs are also sent using IGMP from receivers to local routers.

check). Figure 16.3 shows a general view of some of the terms commonly used in an IP multicast network.

The key component of the multicast network is the multicast-capable router, which replicates the packets. The routers in the IP multicast network, which has exactly the same topology as the unicast network it is based on, use a *multicast routing protocol* to build a *distribution tree* to connect receivers (this term is preferred to the multimedia implications of *listeners*, but the listener term is also used) to *sources*. The distribution tree is rooted at the source. The interface on the router leading toward the source is the *upstream* interface, although the less precise terms *incoming* or *inbound* interface are also used. There should be only one upstream interface on the router receiving multicast packets. The interface on the router leading toward the receivers is the *downstream* interface, although the less precise terms *outgoing* or *outbound* interface are used as well. There can be 0 to *N* – 1 downstream interfaces on a router, where *N* is the number of logical interfaces on the router. To prevent looping, the upstream interface should never receive copies of downstream multicast packets.

Routing loops are disastrous in multicast networks because of the repeated replication of packets. Modern multicast routing protocols need to avoid routing loops, packet by packet, much more rigorously than in unicast routing protocols.

Each subnetwork with hosts on the router that has at least one interested receiver is a *leaf* on the distribution *tree*. Routers can have multiple *leafs* or *leaves* (both terms are used) on different interfaces and must send a copy of the IP multicast packet out on each interface with a leaf. When a new leaf subnetwork is added to the tree (i.e., the interface to the host subnetwork previously received no copies of the multicast packets), a new *branch* is built, the leaf is joined to the tree, and replicated packets are now sent out on the interface.

When a branch contains no leaves because there are no interested hosts on the router interface leading to that IP subnetwork, the branch is *pruned* from the distribution tree, and no multicast packets are sent out from that interface. Packets are replicated and sent out from multiple interfaces only where the distribution tree branches at a router, and no link ever carries a duplicate flow of packets.

Collections of hosts all receiving the same stream of IP packets, usually from the same multicast source, are called *groups*. In IP multicast networks, traffic is delivered to multicast groups based on an IP multicast address or group address. The groups determine the location of the leaves, and the leaves determine the branches on the multicast network. Some multicast routing protocols use a special RP router to allow receivers to find sources efficiently.

DENSE AND SPARSE MULTICAST

Multicast addresses represent groups of receivers, and two strategies can be employed to ensure that all receivers interested in a multicast group receive the traffic.

Dense-Mode Multicast

The assumption here is that almost all possible subnets have at least one receiver wanting to receive the multicast traffic from a source, so the network is *flooded* with traffic on all possible branches and then pruned back as branches do not express an interest in receiving the packets—explicitly (by message) or implicitly (timeout silence). This is the *dense mode* of multicast operation. LANs are appropriate environments for dense-mode operation. In practice, although PIM-DM is worth covering (and we'll even configure it!) there aren't a lot of scenarios in which people would seriously consider it. Periodic blasting of source content is neither a very scalable nor efficient use of resources.

Sparse-Mode Multicast

The assumption here is that very few of the possible receivers want packets from this source, so the network establishes and sends packets only on branches that have at least one leaf indicating (by message) a desire for the traffic. This is the *sparse mode* of multicast operation. WANs (like the Internet) are appropriate networks for sparse-mode operation. Sparse-mode multicast protocols use the special RP router to allow receivers to find sources efficiently.

Specific networks can run whichever mode makes sense. A low-volume multicast application can make effective use of dense mode, even on a WAN. A high-volume application on a LAN might still use sparse mode for efficiency.

Some multicast routing protocols, especially older ones, support only dense-mode operation—which makes them difficult to use efficiently on the public Internet. Others allow sparse mode as well. If sparse-dense mode is supported, the multicast routing protocol allows some special dense multicast groups to be used to the RPs—at which point the router operates in sparse mode.

MULTICAST NOTATION

To avoid multicast routing loops, every multicast router must always be aware of the interface that leads to the source of that multicast group content by the shortest path. This is the upstream (incoming) interface, and packets should never be forwarded back toward a multicast source. All other interfaces are potential downstream (outgoing) interfaces, depending on the number of branches on the distribution tree.

Routers closely monitor the status of the incoming and outgoing interfaces, a process that determines the multicast forwarding state. A router with a multicast forwarding state for a particular multicast group is essentially "turned on" for that group's content. Interfaces on the router's outgoing interface list (OIL) send copies of the group's packets received on the incoming interface list for that group. The incoming and outgoing interface lists might be different for different multicast groups.

The multicast forwarding state in a router is usually written in (S,G) or (*,G) notation. These are pronounced "S comma G" and "star comma G," respectively. In (S,G), the S refers to the unicast IP address of the source for the multicast traffic, and the G refers to the particular multicast group IP address for which S is the source. All multicast packets sent from this source have S as the source address and G as the destination address.

The asterisk (*) in the (*,G) notation is a wild card indicating that the source sending to group G is unknown. Routers try to track down these sources when they have to in order to operate more efficiently.

MULTICAST CONCEPTS

The basic terminology of multicast is complicated by the use of several related concepts. Many of these apply to how the routers on a multicast-capable network handle multicast packets and have little to do with hosts on LANs, but they are important concepts nonetheless.

Reverse-Path Forwarding

Unicast forwarding decisions are typically based on the destination address of the packet arriving at a router. The unicast routing table is organized by destination subnet and mainly set up to forward the packet toward the destination.

In multicast, the router forwards the packet away from the source to make progress along the distribution tree and prevent routing loops. The router's multicast forwarding state runs more logically by organizing tables based on the reverse path, from the receiver back to the root of the distribution tree. This process is known as reverse-path forwarding (RPF).

The router adds a branch to a distribution tree depending on whether the request for traffic from a multicast group passes the RPF check. Every multicast packet received must pass an RPF check before it is eligible to be replicated or forwarded on any interface.

The RPF check is essential for every router's multicast implementation. When a multicast packet is received on an interface, the router interprets the source address in the multicast IP packet as the destination address for a unicast IP packet. The source multicast address is found in the unicast routing table, and the outgoing interface is determined. If the outgoing interface found in the unicast routing table is the same as the interface that the multicast packet was received on, the packet passes the RPF check. Multicast packets that fail the RPF check are dropped because the incoming interface is not on the shortest path back to the source.

Routers can build and maintain separate tables for RPF purposes. The router must have some way to determine its RPF interface for the group, which is the interface topologically closest to the root. The distribution tree should follow the shortest-path tree topology for efficiency. The RPF check helps to construct this tree.

The RPF Table

The RPF table plays the key role in the multicast router. The RPF table is consulted for every RPF check, which is performed at intervals on multicast packets entering the multicast router. Distribution trees of all types rely on the RPF table to form properly, and the multicast forwarding state also depends on the RPF table.

The routing table used for RPF checks can be the same routing table used to forward unicast IP packets, or it can be a separate routing table used only for multicast RPF checks. In either case, the RPF table contains only unicast routes because the RPF check is performed on the source address of the multicast packet (not the multicast group destination address), and a multicast address is forbidden from appearing in the source address field of an IP packet header. The unicast address can be used for RPF checks because there is only one source host for a particular stream of IP multicast content for a multicast group address, although the same content could be available from multiple sources.

Populating the RPF Table

If the same routing table used to forward unicast packets is also used for the RPF checks, the routing table is populated and maintained by the traditional unicast routing protocols such as Border Gateway Protocol (BGP), Intermediate System-to-Intermediate System (IS–IS), OSPF, and Routing Information Protocol (RIP). If a dedicated multicast

RPF table is used, this table must be populated by some other method. Some multicast routing protocols, such as the Distance Vector Multicast Routing Protocol (DVMRP), essentially duplicate the operation of a unicast routing protocol and populate a dedicated RPF table. Others, such as Protocol Independent Multicast (PIM), do not duplicate routing protocol functions and must rely on some other routing protocol to set up this table—which is why PIM is protocol independent.

Some traditional routing protocols (such as BGP and IS–IS) now have extensions to differentiate between different sets of routing information sent between routers for unicast and multicast. For example, there is multiprotocol BGP (MBGP) and multitopology routing in IS–IS (M-ISIS). Multicast Open Shortest Path First (MOSPF) also extends OSPF for multicast use, but goes further than MBGP or M-ISIS and makes MOSPF into a complete multicast routing protocol on its own. When these routing protocols are used, routes can be tagged as multicast RPF routers and used by the receiving router differently than the unicast routing information.

Using the main unicast routing table for RPF checks provides simplicity. A dedicated routing table for RPF checks allows a network administrator to set up separate paths and routing policies for unicast and multicast traffic, allowing the multicast network to function more independently of the unicast network. The following section discusses in further detail how PIM operates, although the concepts could be applied to other multicast routing protocols.

Shortest-Path Tree

The distribution tree used for multicast is rooted at the source and is the shortest-path tree (SPT) as well. Consider a set of multicast routers without any active multicast traffic for a certain group (i.e., they have no multicast forwarding state for that group). When a router learns that an interested receiver for that group is on one of its directly connected subnets, the router attempts to join the tree for that group.

To join the distribution tree, the router determines the unicast IP address of the source for that group. This address can be a simple static configuration in the router, or use more complex methods.

To build the SPT for that group, the router executes an RPF check on the source address in its routing table. The RPF check produces the interface closest to the source, which is where multicast packets from this source for this group should flow into the router.

The router next sends a join message out on this interface using the proper multicast protocol to inform the upstream router that it wishes to join the distribution tree for that group. This message is an (S,G) join message because both S and G are known. The router receiving the (S,G) join message adds the interface on which the message was received to its OIL for the group and performs an RPF check on the source address. The upstream router then sends an (S,G) join message out the RPF interface toward the source, informing the upstream router that it also wants to join the group.

Each upstream router repeats this process, propagating joins out the RPF interface—building the SPT as it goes. The process stops when the join message does the following:

- Reaches the router directly connected to the host that is the source, or
- Reaches a router that already has multicast forwarding state for this source-group pair.

In either case, the branch is created, each of the routers has multicast forwarding state for the source-group pair, and packets can flow down the distribution tree from source to receiver. The RPF check at each router ensures that the tree is an SPT.

SPTs are always the shortest path, but they are not necessarily short. That is, sources and receivers tend to be on the periphery of a router network (not on the backbone) and multicast distribution trees have a tendency to sprawl across almost every router in the network. Because multicast traffic can overwhelm a slow interface, and one packet can easily become a hundred or a thousand on the opposite side of the backbone, it makes sense to provide a shared tree as a distribution tree so that the multicast source could be located more centrally in the network (on the backbone). This sharing of distribution trees with roots in the core network is accomplished by a multicast rendezvous point.

Rendezvous Point and Rendezvous-Point Shared Trees

In a shared tree, the root of the distribution tree is a router (not a host), and is located somewhere in the core of the network. In the primary sparse-mode multicast routing protocol, Protocol Independent Multicast sparse mode (PIM-SM), the core router at the root of the shared tree is the RP. Packets from the upstream source and join messages from the downstream routers "rendezvous" at this core router.

In the RP model, other routers do not need to know the addresses of the sources for every multicast group. All they need to know is the IP address of the RP router. The RP router knows the sources for all multicast groups.

The RP model shifts the burden of finding sources of multicast content from each router—the (S,G) notation—to the network—the (*,G) notation knows only the RP. Exactly how the RP finds the unicast IP address of the source varies, but there must be some method to determine the proper source for multicast content for a particular group.

Consider a set of multicast routers without any active multicast traffic for a certain group. When a router learns that an interested receiver for that group is on one of its directly connected subnets, the router attempts to join the distribution tree for that group back to RP (not to the actual source of the content). In some sparse-mode protocols, the shared tree is called the rendezvous-point tree (RPT).

When the branch is created, packets can flow from the source to the RP and from the RP to the receiver. Note that there is no guarantee that the shared tree (RPT) is the shortest path tree to the source. Most likely it is not. However, there are ways to "migrate" a shared tree to an SPT once the flow of packets begins. In other words, the forwarding state can transition from (*,G) to (S,G). The formation of both types of trees depends heavily on the operation of the RPF check and the RPF table.

PROTOCOLS FOR MULTICAST

Multicast is not a single protocol used for a specific function, like FTP. Nor is multicast a series of separate protocols that can be used as desired between adjacent hosts and routers to perform a function, like IS–IS and OSPF. Multicast is a series of related protocols that must be carefully coordinated across and between an AS and often among hosts.

The family of multicast protocols is due to the complexity of source discovery and the mechanisms used to perform this task. Most hosts can send and receive multicast frames and packets on a LAN as easily as they handle broadcast or unicast. Routers must be capable of sending copies of a single received packet out on more than one interface (replication), and many low-end routers cannot do this. In addition, routers must be able to use unicast routing tables for multicast purposes, or construct special tables for multicast information (again, many low-end routers cannot do this).

Multicast routers must be able to maintain state on each interface with regard to multicast traffic. That is, the router must know which multicast groups have active receivers on an outgoing interface (called downstream interfaces) and which interface is the "closest" to the source (called *upstream interface*). These interfaces vary from group to group, one group can have more than one potential source (for redundancy purposes), and special routers might be employed for many groups (the RPs).

Multicast Hosts and Routers

Multicast tasks are very different for hosts versus routers. At this juncture, we will extend the multicast discussion beyond IPv4 to IPv6 and hosts. General points follow.

- Hosts must be able to join and leave multicast groups. The major protocols here are various versions of the Internet Group Management Protocol (IGMP) in IPv4 and Multicast Listener Discovery (MLD) in IPv6.
- Hosts (users) must know the content of multicast groups. The related Session Announcement Protocol and Session Description Protocol (SAP/SDP, defined in RFC 2974 and RFC 2327) are the standard protocols used to describe the content and some other aspects of multicast groups. These should *not* be used as a method of multicast source discovery.
- Routers must be able to find the sources of multicast content, both in their own multicast (routing) domain and in others. For sparse modes, this means finding the RPs. These can be configured statically, or use protocols such as Auto-RP, anycast RP (RFC 3446), bootstrap router (BSR), or MSDP (RFC 3618). For IPv6, *embedded RP* is used instead of MSDP—which is not defined for IPv6 use. (This point actually applies to ASM, not SSM, discussed in material following.)
- Routers must be able to prevent loops that replicate the same packet over and over. The techniques here are not really protocols, and include the use of scoping (limiting multicast packet hops) and RPF checks.

- Routers must provide missing multicast information when feasible. Multicast networks can use Pragmatic General Multicast (PGM) to add some TCP features lacking in UDP to multicast networks. However, the only assurance is that you know you missed something. Application-specific mechanisms can do the same thing with simple sequence numbers.

Fortunately, only a few of these protocols are really used for multicast at present on the Internet. The only complication is that some of the special protocols used for IPv4 multicasting do not work with IPv6, and thus different protocols perform the same functions.

Multicast Group Membership Protocols

Multicast group membership protocols allow a router to know when a host on a directly attached subnet, typically a LAN, wants to receive traffic from a certain multicast group. Even if more than one host on the LAN wants to receive traffic for that multicast group, the router has to send only one copy of each packet for that multicast group out on that interface because of the inherent broadcast nature of LANs. Only when the router is informed by the multicast group membership protocol that there are no interested hosts on the subnet can the packets be withheld and that leaf pruned from the distribution tree.

Internet Group Management Protocol for IPv4

There is only one standard IPv4 multicast group membership protocol: the Internet Group Management Protocol (IGMP). However, IGMP has several versions that are supported by hosts and routers. There are currently three versions of IGMP.

IGMPv1—The original protocol defined in RFC 1112. An explicit join message is sent to the router, but a timeout is used to determine when hosts leave a group. This process wastes processing cycles on the router, especially on older or smaller routers.

IGMPv2—Among other features, IGMPv2 (RFC 2236) adds an explicit leave message to the join message so that routers can more easily determine when a group has no interested listeners on a LAN.

IGMPv3—Among other features, IGMPv3 (RFC 3376) optimizes support for a single source of content for a multicast group or source-specific multicast (SSM). (RFC 1112 supported both many-to-many and one-to-many multicast, but one-to-many is considered the more viable model for the Internet at large.)

Although the various versions of IGMP are backward compatible, it is common for a router to run multiple versions of IGMP on LAN interfaces because backward compatibility is achieved by dropping back to the most basic of all versions run on a LAN. For example, if one host is running IGMPv1, any router attached to the LAN running IGMPv2 drops back to IGMPv1 operation—effectively eliminating the IGMPv2 advantages. Running multiple IGMP versions ensures that both IGMPv1 and IGMPv2 hosts find peers for their versions on the router.

Multicast Listener Discovery for IPv6

IPv6 does not use IGMP to manage multicast groups. Multicast groups are an integral part of IPv6, and the Multicast Listener Discovery (MLD) protocol is an integral part of IPv6. Some IGMP functions are assumed by ICMPv6, but IPv6 hosts perform most multicast functions with MLD. MLD comes in two versions: MLD version 1 (RFC 2710) has basic functions, and MLDv2 (RFC 3590) supports SSM groups.

Multicast Routing Protocols

There are five multicast routing protocols.

Distance-Vector Multicast Routing Protocol

This is the first of the multicast routing protocols and hampered by a number of limitations that make this method unattractive for large-scale Internet use. DVMRP is a dense-mode-only protocol that uses the flood-and-prune, or implicit join method, to deliver traffic everywhere and then determines where uninterested receivers are. DVMRP uses source-based distribution trees in the form (S,G).

Multicast Open Shortest Path First

This protocol extends OSPF for multicast use, but only for dense mode. However, MOSPF has an explicit join message, and thus routers do not have to flood their entire domain with multicast traffic from every source. MOSPF uses source-based distribution trees in the form (S,G).

PIM Dense Mode

This is Protocol Independent Multicast operating in dense mode (PIM DM), but the differences from PIM sparse mode are profound enough to consider the two modes separately. PIM also supports sparse-dense mode, but there is no special notation for that operational mode. In contrast to DVMRP and MOSPF, PIM dense mode allows a router to use any unicast routing protocol and performs RPF checks using the unicast routing table. PIM dense mode has an implicit join message, so routers use the flood-and-prune method to deliver traffic everywhere and then determine where the uninterested receivers are. PIM dense mode uses source-based distribution trees in the form (S,G), as do all dense-mode protocols.

PIM Sparse Mode

PIM sparse mode allows a router to use any unicast routing protocol and performs RPF checks using the unicast routing table. However, PIM sparse mode has an explicit join message, so routers determine where the interested receivers are and send join messages upstream to their neighbors—building trees from receivers to RP. The Protocol

Table 16.1 Major Characteristics of Multicast Routing Protocols

Multicast Routing Protocol	Dense Mode	Sparse Mode	Implicit Join	Explicit Join	(S,G) SBT	(*,G) Shared Tree
DVMRP	Yes	No	Yes	No	Yes	No
MOSPF	Yes	No	No	Yes	Yes	No
PIM-DM	Yes	No	Yes	No	Yes	No
PIM-SM	No	Yes	No	Yes	Yes, maybe	Yes, initially
CBT	No	Yes	No	Yes	No	Yes

Independent Multicast sparse mode uses an RP router as the initial source of multicast group traffic and therefore builds distribution trees in the form (*,G), as do all sparse-mode protocols. However, PIM sparse mode migrates to an (S,G) source-based tree if that path is shorter than through the RP for a particular multicast group's traffic.

Core-Based Trees

Core-based trees (CBT) share all of the characteristics of PIM sparse mode (sparse mode, explicit join, and shared [*,G] trees), but are said to be more efficient at finding sources than PIM sparse mode. CBT is rarely encountered outside academic discussions and the experimental RFC 2201 from September 1997. There are no large-scale deployments of CBT, commercial or otherwise. The differences among the five multicast routing protocols are summarized in Table 16.1.

It is important to realize that retransmissions due to a high bit-error rate on a link or overloaded router can make multicast as inefficient as repeated unicast.

Any-Source Multicast and SSM

RFC 1112 originally described both one-to-many (for radio and television) and many-to-many (for videoconferences and application on-line gaming) multicasts. This model is now known as *Any-Source Multicast* (ASM). To support many-to-many multicasts, the *network* is responsible for source discovery. So, whenever a host expresses a desire to join a group the network must find all the sources for that group and deliver them to the receiver.

Source discovery is especially complex with interdomain scenarios (source in one AS, receiver/s in another). And most plans to commercialize Internet multicasts, such as bringing radio station and television channel multicasts directly onto the Internet, revolve around the one-to-many model exclusively. So, the one-to-many scenario has been essentially split off from the all-embracing RFC 1112 vision and become Source-Specific Multicast (SSM, defined in FC 3569).

As the name implies, SSM supports multicast content delivery from only one specific source. In SSM, source discovery is not the responsibility of the network but of the

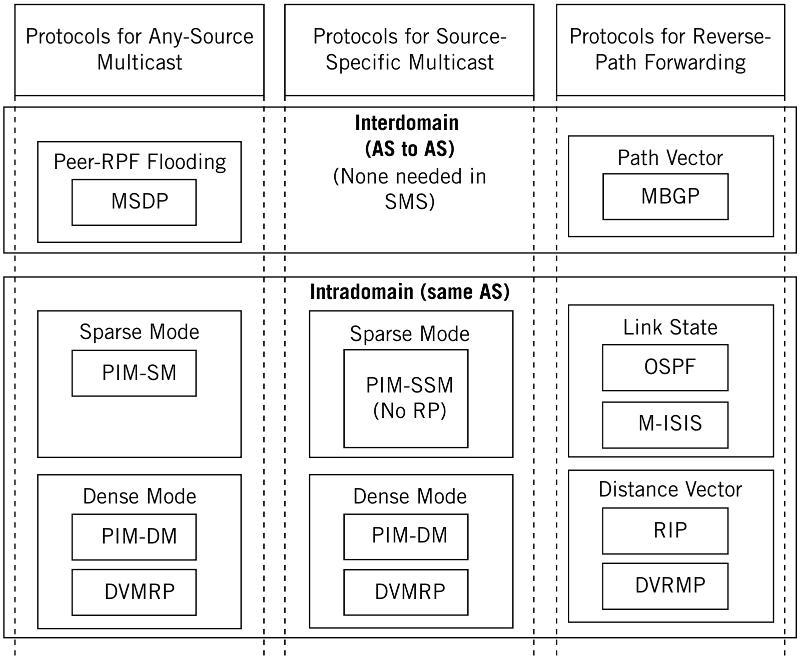


FIGURE 16.4

Suite of multicast protocols showing how those for ASM, SSM, and RFP checks fit together and are used.

receivers (hosts). This eliminates much of the complexity of multicast mechanisms required in ASM and the use of MSDP. It also eliminates some of the scaling considerations associated with traffic on (*,G) groups.

ASM and SSM are not protocols but service models. Most of what is described in this chapter applies to ASM (the more general model). But keep in mind that SSM does away with many of the procedures covered in detail here that apply to ASM, including RPs, RPTs, and MSDP. Figure 16.4 shows the current suite of multicast protocols and how they all fit together.

Multicast Source Discovery Protocol

MSDP, described in RFC 3618, is a mechanism to connect multiple PIM-SM domains (usually, each in an AS). Each PIM-SM domain can have its own independent RPs, and these do not interact in any way (so MSDP is not needed in SSM scenarios). The advantages of MSDP are that the RPs do not need any other resource to find each other and that domains can have receivers only and get content without globally advertising group membership. In addition, MSDP can be used with protocols other than PIM-SM.

MSDP routers in a PIM-SM domain peer with their MSDP router peers in other domains. The peering session uses a TCP connection to exchange control information. Each domain has one or more of these connections in its "virtual topology." This allows domains to discover multicast sources in other domains. If these sources are deemed of interest to receivers in another domain, the usual source-tree mechanism in PIM-SM is used to deliver multicast content—but now over an interdomain distribution tree. More details about MSDP are beyond the scope of this introductory chapter.

Frames and Multicast

Multicasting on a LAN is a good place to start an investigation of multicasting in general. Consider a single LAN, without routers, with a multicast source sending to a certain group. The rest of the hosts are receivers interested in the multicast group's content. So, the multicast source host generates packets with its unicast IP address as the source and the group address as the destination.

One issue comes up immediately. The packet source address obviously will be the unicast IP address of the host originating the multicast content. This translates to the MAC address for the source address in the frame in which the packet is encapsulated. The packet's destination address will be the multicast group. So far, so good. But what should be the frame's destination address that corresponds to the packet's multicast group address?

Using the LAN broadcast MAC address defeats the purpose of multicast, and hosts could have access to many multicast groups. Broadcasting at the LAN level makes no sense. Fortunately, there is an easy way out of this. The MAC address has a bit that is set to 0 for unicast (the LAN term is *individual address*) and to a 1 to indicate that this is a multicast address. Some of these addresses are reserved for multicast groups for specific vendors or MAC-level protocols. Internet multicast applications use the range 0x01-00-5E-00-00-00 to 0x01-00-5E-FF-FF-FF. TCP/IP multicast receivers listen for frames with one of these addresses when the application joins a multicast group and stops listening when the application terminates or the host leaves the group.

So, 24 bits are available to map IPv4 multicast addresses to MAC multicast addresses. But all IPv4 addresses, including multicast addresses, are 32 bits long. There are 8 bits left over. How should IPv4 multicast addresses be mapped to MAC multicast addresses to minimize the chance of "collisions" (two different multicast groups mapped to the same MAC multicast address)?

All IPv4 multicast addresses begin with the same four bits (1110), so we only have to really worry about 4 bits (not 8). We shouldn't drop the last bits of the IPv4 address, because these are almost guaranteed to be host bits—depending on subnet mask. But the high-order bits, the rightmost bits, are almost always network bits and we're only worried about one LAN for now.

One other bit of the remaining 24 MAC address bits is reserved (an initial 0 indicates an Internet multicast address), so let's just drop the 5 bits following the initial 1110 in the IPv4 address and map the 23 remaining bits (one for one) into the last 23 bits of the MAC address. This procedure is shown in Figure 16.5.

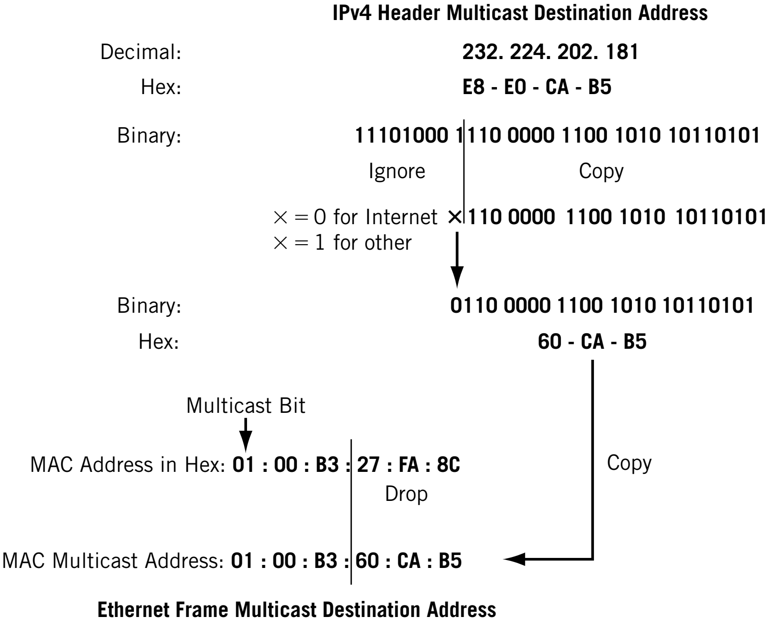

IPv4 Header Multicast Destination Address
Decimal: 232. 224. 202. 181
Hex: E8 - E0 - CA - B5
Binary: 11101000 1110 0000 1100 1010 10110101
Ignore
Copy
× = 0 for Internet
× = 1 for other
×110 0000 1100 1010 10110101
Binary: 0110 0000 1100 1010 10110101
Hex: 60 - CA - B5
Multicast Bit
MAC Address in Hex: 01 : 00 : B3 : 27 : FA : 8C
Drop
Copy
MAC Multicast Address: 01 : 00 : B3 : 60 : CA : B5
Ethernet Frame Multicast Destination Address

FIGURE 16.5

How to convert from IPv4 header multicast to Ethernet MAC multicast address formats.

Note that this process means that there are 32 (2^5) IPv4 multicast addresses that could map to the same MAC multicast addresses. For example, multicast IPv4 addresses `224.8.7.6` and `229.136.7.6` translate to the same MAC address (0x01-00-5E-08-07-06). This is a real concern, and because the host will accept frames sent to both multicast groups, the IP software must reject one or the other. This problem does not exist in IPv6, but is always a concern in IPv4.

Once the MAC address for the multicast group is determined, the operating system essentially orders the NIC card to join or leave the multicast group and accept frames sent to the address as well as the host's unicast address or ignore that multicast group's frames. It is possible for a host to receive multicast content from more than one group at the same time, of course. The procedure for IPv6 multicast packets inside frames is nearly identical, except for the MAC destination address 0x3333 prefix and other points outlined in the previous section.

IPv4 Multicast Addressing

The IPv4 addresses (Class D in the classful addressing scheme) used for multicast usage range from `224.0.0.0` to `239.255.255.255`. Assignment of addresses in this range is controlled by the Internet Assigned Numbers Authority (IANA). Multicast addresses can never be used as a source address in a packet (the source address is always the unicast

IP address of the content originator). Certain subranges within the range of addresses are reserved for specific uses.

- `224.0.0.0/24`—The link-local multicast range (these packets never pass through routers)
- `224.2.0.0/16`—The SAP/SDP range
- `232.0.0.0/8`—The Source-Specific Multicast (SSM) range
- `233.0.0.0/8`—The AS-encoded statically assigned GLOP range defined in RFC 3180
- `239.0.0.0/8`—The administratively scoped multicast range defined in RFC 2365 (these packets *may* pass through a certain number of routers)

For a complete list of currently assigned IANA multicast addresses, refer to the *www.iana.org/assignments/multicast-addresses* Web site. If multicast addresses had

Table 16.2 Multicast Addresses Used for Various Protocols

Address	Purpose	Comment
224.0.0.0	Reserved base address	RFC 1112
224.0.0.1	All systems of this subnet	RFC 1112
224.0.0.2	All routers on this subnet	
224.0.0.3	Unassigned	
224.0.0.4	DVMRP routers on this subnet	RFC 1075
224.0.0.5	All OSPF routers on this subnet	RFC 1583
224.0.0.6	All OSPF DRs on this subnet	RFC 1583
224.0.0.7	All ST (Streams protocol) routers on this subnet	RFC 1190
224.0.0.8	All ST hosts on this subnet	RFC 1190
224.0.0.9	All RIPv2 routers on this subnet	RFC 1723
224.0.0.10	All Cisco IGRP routers on this subnet	(Cisco)
224.0.0.11	All Mobile IP agents	
224.0.0.12	DHCP server/relay agents	RFC 1884
224.0.0.13	All PIM routers	(IANA)
224.0.014-224.0.0.21	Assigned to various routing protocols and router features	(IANA)
224.0.0.22	IGMP	(IANA)
224.0.0.23-244.0.0.255	See *www.iana.org/assignments/multicast-addresses*	(IANA)

been assigned in the same manner that unicast addresses were allocated, the Class D address space would have been exhausted long ago. However, IANA allocates static multicast addresses only for protocols. Routers cannot forward packets in these ranges. Some of these addresses are outlined in Table 16.2.

A simple dynamic address allocation mechanism is used in the SAP/SDP block to prevent multicast address exhaustion. Applications, such as the Session Directory Tool (SDR), use this mechanism to randomly select an unused address in this range. This dynamic allocation mechanism for global multicast addresses is similar to the DHCP function, which dynamically assigns unicast addresses on a LAN.

However, some applications require static multicast addresses. So, GLOP (described in RFC 3180) provides static multicast ranges for organizations that already have an AS number. (GLOP is not an acronym or abbreviation—it's just the name of the mechanism.) GLOP uses the 2-byte AS number to derive a /24 address block within the 233/8 range. It's worth noting that there are no GLOP addresses set aside for 4-byte AS numbers. The static multicast range is derived from the following form:

```
233.[first byte of AS].[second byte of AS].0/24
```

For example, AS 65001 is allocated 233.253.233.0/24—and only this AS can use it. The following is an easy way to compute this address.

1. Convert the AS number to hexadecimal (65001 = 0xFDE9).
2. Convert the first byte back to decimal (0xFD = 253).
3. Convert the second byte back to decimal (0xE9 = 233).

Addresses in the 239/8 range are defined as administratively scoped. Packets sent to these addresses should not be forwarded by a router outside an administratively defined boundary (usually a domain).

Addresses in the 232/8 range are reserved for SSM. A nice feature of SSM is that the multicast group address no longer needs to be globally unique. The source-group "channel," or tuple, provides uniqueness because the receiver is expressing interest in only one source for the group.

SSM has solved the multicast addressing allocation headache. With SSM, as well as GLOP, administrative scoping, and SAP/SDP, IPv4 multicast address allocation is sufficient until IPv6 becomes more common.

IPv6 Multicast Addressing

In IPv6, the number of multicast (and unicast) addresses available is not an issue. All IPv6 multicast addresses start with 1111 1111 (0xFF). As in IPv4, no IPv6 packet can have an IPv6 multicast address as a source address. There is really no such thing as a "broadcast" in IPv6. Instead, devices must belong to certain multicast groups and pay attention to packets sent to these groups. The structure of the IPv6 multicast address is shown in Figure 16.6.

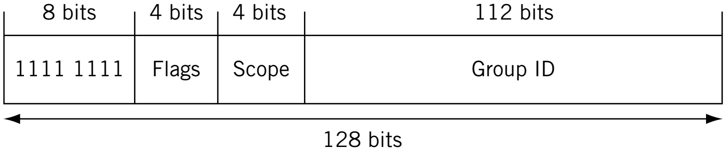


FIGURE 16.6

The IPv6 multicast address format. Note the presence of the scope field.

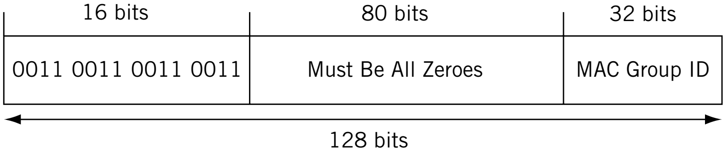


FIGURE 16.7

The IPv6 multicast group addresses showing how the MAC group ID is embedded.

Format Prefix

This 8-bit field is simply `1111 1111` (`0xFF`).

Flags

As of RFC 2373, the only flag defined for this 4-bit field is Transient (T). If 0, the multicast address is a permanently assigned well-known address allocated by IANA. If 1, the multicast address is not permanently assigned (transient).

Scope

This 4-bit field establishes the multicast packets' boundaries. RFC 2372 defines several well-known scopes, including node-local (1), link-local (2), site-local (3), organization-local (8), and global (E). Packets sent to `0xFF02:`*`X`* are confined to a single link and cannot pass through a router (this issue came up in the IGP chapter with RIPng).

Group ID

The IPv6 multicast group ID is 112 bits long. Permanently assigned group IDs are valid regardless of the scope value, whereas transient group IDs are valid *only* within a particular scope. The 122 bits of the Group ID field pose a challenge to the 48-bit MAC address (and only 23 of those bits were used in IPv4). But the solution is much simpler than in IPv4. RFC 2373 recommends using the low-order 32 bits of the Group ID and setting the high-order 16 bits to `0x3333`. This is shown in Figure 16.7.

Naturally, there are 80 more bits that could be used in the Group ID field. For now, RFC 2373 recommends setting the 80+ bits available for multicast group IDs to 0s. If there is a problem with 32 bits for multicast groups, which can be as many as 4 billion, probably in the future the RFC group will think about extending the bits.

PIM-SM

The most important multicast routing protocol for the Internet today is PIM sparse mode, defined in RFC 2362. PIM-SM is ideal for a number of reasons, such as its protocol-independent nature (PIM can use regular unicast routing tables for RPF checks and other things), and it's a nice fit with SSM (in fact, not much else fits at all with SSM). So, we'll look at PIM-SM in a little more detail (also in addition, because that's what we'll be using on the Illustrated Network's routers).

If a potential receiver is interested in the content of a particular multicast group, it sends an IGMP Join message to the local router—which must know the location of the network RPs servicing that group. If the local router is not currently on the distribution tree for that group, the router sends a PIM Join message (not an IGMP message) through the network until the router becomes a leaf on the shared tree (RPT) to the RP. Once multicast packets are flowing to the receiver, the routers all check to see if there is a shorter path from the source to the destination than through the RP. If there is, the routers will transition the tree from an RPT to an SPT using PIM Join and Prune messages (technically, they are PIM Join/Prune messages, but it is common to distinguish them). The SPT is rooted at the designated router of the source. All of this is done transparently to the receivers and usually works very smoothly.

There are other reasons to transition from an RPT to an SPT, even if the SPT is actually longer than the RPT. An RP might become quite busy, and the shortest path might not be optimal as determined by unicast routing protocols. A lot of multicast discussion at ISPs involves issues such as how many RPs there should be (how many groups should each service?) and where they should be located (near their sources? centrally?). A related issue is how routers know about RPs (statically? Auto-RP? BSR?), but these discussions have no clear or accepted answers.

There is only one PIM-SM feature that needs to be explained. How does traffic get from the sender's local router to the RP? The rendezvous point could create a tree directly to every source, but if there is a lot of sources, there is a lot of state information to maintain. It would be better if the senders' local routers could send the content directly to the RP.

But how? The destination address of all multicast packets is a group address and not a unicast address. So, the source's router (actually, the DR) encapsulates the multicast packets inside a unicast packet sent to the RP and tunnels the packet to the RP in this form. The RP decapsulates the multicast content and makes it available for distribution over the RPT tree.

There is much more to PIM-SM that has not been detailed here, such as PIM-SM for SSM (sometimes seen as PIM-SSM). But it is enough to explain the interplay among host receivers, IGMP (in IPv4), MLD (in IPv6), PIM itself, the RP, and the source.

The Resource Reservation Protocol and PGM

A lot of books and material on multicast include long discussions of the Resource Reservation Protocol (RSVP), and some multicast routers and hosts still use RSVP to signal the network that the multicast packet stream they will be receiving will consume a certain amount of resources on the network. However, the most common use of RSVP today is not with multicast but with Multiprotocol Label Switching (MPLS)—and that's where we'll put RSVP.

RVSP makes sense for multicast in a restricted bandwidth environment. But the need for RSVP was undermined (as was ATM) by the embarrassment of bandwidth available on LANs and router backbones (the video network YouTube today uses more bandwidth than the entire Internet had in 2000). On slow networks, the biggest shortcoming is that you can't reserve bandwidth you don't have. If you do anyway, you're really just performing admission control (limited to those who are allowed to connect over the network) and hosing the other applications. Everything works better with enough bandwidth.

However, this is not to say that multicast is fine using UDP in all cases—especially when multicast content must cross ISP boundaries, where bandwidth on these heavily used links is often consumed by traffic. Nothing is more annoying when receiving multicast content, voice, or video than dropped packets causing screen freezes and unpredictable silences. So, routers and hosts can use Pragmatic General Multicast (PGM), described in RFC 3208. PGM occupies the same place in the TCP/IP stack as TCP itself. PGM runs on sender and receiver hosts, and on routers (which perform the PGM router assist function).

As mentioned, the goal of PGM is not to make multicast UDP streams as reliable as TCP. The PGM goal is to allow senders or routers (performing router assist functions) to supply missing multicast packets if possible (such as for stock-ticker applications) or to assure receivers that the data is indeed missing and not just delayed (it does this by simply sequencing multicast packets). The issue is that you have to carry all of this state information in routers, which is not good for scaling.

Multicast Routing Protocols

Now we can go back to the network. We'll have to run a multicast routing protocol on our routers. We'll run PIM, which is the most popular multicast protocol. But PIM can be configured in dense "send-everywhere" mode or sparse "only if you ask" mode. Which should we use?

Let's consider our router configuration. Nothing is easier to configure than dense mode. We can just configure PIM dense mode (PIM-DM) to run on every router interface (even the LAN interfaces if we like—the PIM messages won't hurt anything), except for the network management interface on Juniper Networks routers (fxp0.0). Multicast traffic is periodically flooded everywhere and pruned back as IGMP membership reports come in on local area network interfaces. This is just an exercise for our lab network. You definitely should not try this at home. The following is the configuration on router CE6:

```
set protocols pim interface all mode dense;
set protocols pim interface fxp0.0 disable;
```

It is not necessary to configure IGMP on the LAN interface. As long as PIM is configured, IGMPv2 is run on all interfaces that support broadcasts (including frame relay and ATM). Of course, if a different version of IGMP—such as IGMPv1 or IGMPv3 (`wincli` was running IGMPv3, as shown in Figure 16.2)—is desired, this must be explicitly configured.

It is more interesting and meaningful to configure the PIM sparse mode, because that is what is used, with few exceptions, on the Internet. There are two distinct configurations: one for the RP router and the other on all the non-RP routers. We'll use simple static configuration to locate the RP router, but that's not what is typically done in the real world. The configuration on the RP router, which is router PE5 in this example, follows:

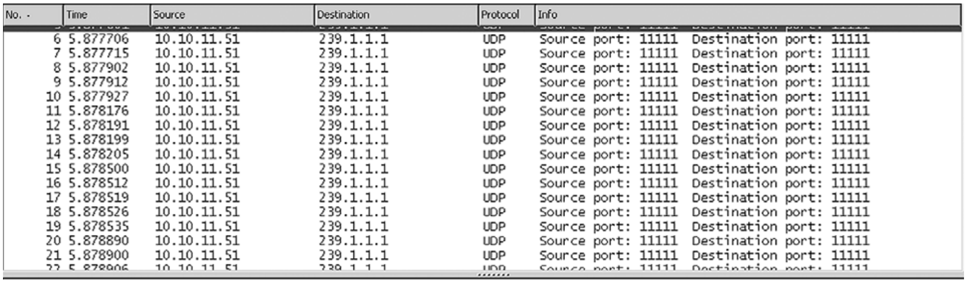
```
set protocols pim rp local address 192.168.5.1;
set protocols pim rp interface all mode sparse;
set protocols pim rp interface fxp0.0 disable;
```

The `local` keyword means that the local router is the RP. The address is the RP address that will be used in PIM messages between the routers. The configuration on the non-RP router, such as `P9`, follows:

```
set protocols pim rp static address 192.168.5.1;
set protocols pim rp interface all mode sparse;
set protocols pim rp interface fxp0.0 disable;
```

The static keyword means that another router is the RP, located at the IP address given. The RP address is used in PIM messages between the routers.

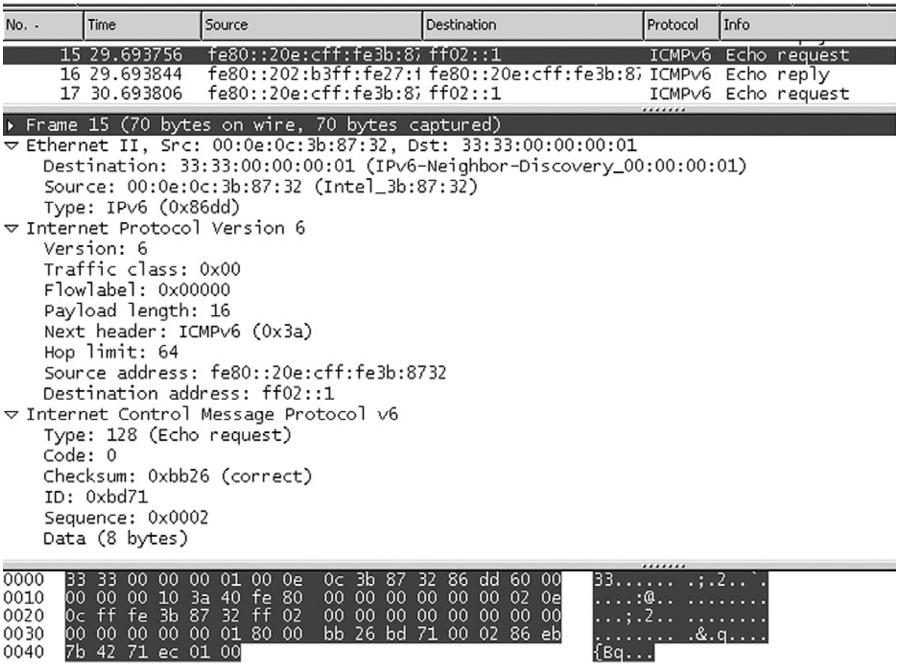
Once PIM is up and running on the rest of the router network (we don't need MSDP because the RP is known everywhere within the merged Best-Ace ISP routing domain and this precludes interdomain ASM use anyway), `wincli2` receives multicast traffic from `wincli1`, as shown in Figures 16.8 and 16.9.



```
No. -   Time        Source          Destination     Protocol  Info
      6 5.877706    10.10.11.51     239.1.1.1       UDP       Source port: 11111  Destination port: 11111
      7 5.877715    10.10.11.51     239.1.1.1       UDP       Source port: 11111  Destination port: 11111
      8 5.877902    10.10.11.51     239.1.1.1       UDP       Source port: 11111  Destination port: 11111
      9 5.877912    10.10.11.51     239.1.1.1       UDP       Source port: 11111  Destination port: 11111
     10 5.877927    10.10.11.51     239.1.1.1       UDP       Source port: 11111  Destination port: 11111
     11 5.878176    10.10.11.51     239.1.1.1       UDP       Source port: 11111  Destination port: 11111
     12 5.878191    10.10.11.51     239.1.1.1       UDP       Source port: 11111  Destination port: 11111
     13 5.878199    10.10.11.51     239.1.1.1       UDP       Source port: 11111  Destination port: 11111
     14 5.878205    10.10.11.51     239.1.1.1       UDP       Source port: 11111  Destination port: 11111
     15 5.878500    10.10.11.51     239.1.1.1       UDP       Source port: 11111  Destination port: 11111
     16 5.878512    10.10.11.51     239.1.1.1       UDP       Source port: 11111  Destination port: 11111
     17 5.878519    10.10.11.51     239.1.1.1       UDP       Source port: 11111  Destination port: 11111
     18 5.878526    10.10.11.51     239.1.1.1       UDP       Source port: 11111  Destination port: 11111
     19 5.878535    10.10.11.51     239.1.1.1       UDP       Source port: 11111  Destination port: 11111
     20 5.878890    10.10.11.51     239.1.1.1       UDP       Source port: 11111  Destination port: 11111
     21 5.878900    10.10.11.51     239.1.1.1       UDP       Source port: 11111  Destination port: 11111
     22 5.878906    10.10.11.51     239.1.1.1       UDP       Source port: 11111  Destination port: 11111
```

FIGURE 16.8

Receiving a stream of multicast traffic from wincli1 across the router network on wincli2.



```
No. -  Time        Source                  Destination             Protocol Info
    15 29.693756   fe80::20e:cff:fe3b:87   ff02::1                 ICMPv6   Echo request
    16 29.693844   fe80::202:b3ff:fe27:1   fe80::20e:cff:fe3b:87   ICMPv6   Echo reply
    17 30.693806   fe80::20e:cff:fe3b:87   ff02::1                 ICMPv6   Echo request

▷ Frame 15 (70 bytes on wire, 70 bytes captured)
▽ Ethernet II, Src: 00:0e:0c:3b:87:32, Dst: 33:33:00:00:00:01
    Destination: 33:33:00:00:00:01 (IPv6-Neighbor-Discovery_00:00:00:01)
    Source: 00:0e:0c:3b:87:32 (Intel_3b:87:32)
    Type: IPv6 (0x86dd)
▽ Internet Protocol Version 6
    Version: 6
    Traffic class: 0x00
    Flowlabel: 0x00000
    Payload length: 16
    Next header: ICMPv6 (0x3a)
    Hop limit: 64
    Source address: fe80::20e:cff:fe3b:8732
    Destination address: ff02::1
▽ Internet Control Message Protocol v6
    Type: 128 (Echo request)
    Code: 0
    Checksum: 0xbb26 (correct)
    ID: 0xbd71
    Sequence: 0x0002
    Data (8 bytes)

0000  33 33 00 00 00 01 00 0e  0c 3b 87 32 86 dd 60 00   33...... .;.2..`.
0010  00 00 00 10 3a 40 fe 80  00 00 00 00 00 00 02 0e   ....:@.. ........
0020  0c ff fe 3b 87 32 ff 02  00 00 00 00 00 00 00 00   ...;.2.. ........
0030  00 00 00 00 00 01 80 00  bb 26 bd 71 00 02 86 eb   ........ .&.q....
0040  7b 42 71 ec 01 00                                  {Bq...
```

FIGURE 16.9

ICMPv6 multicast packets for neighbor discovery, showing how the MAC address is embedded in the IPv6 source address field.

IPv6 Multicast

In contrast to IPv4, where multicast sometimes seems like an afterthought compared to the usual unicast business of the network, IPv6 is fairly teeming with multicast. You have to do a lot to add multicast to IPv4, but IPv6 simply will not work without multicasting. Of course, a lot of this multicast use is confined to single subnets. So, despite being more heavily used, IPv6 multicast is not necessarily easier to deploy (even though you don't have to worry about MSDP).

Figure 16.10 shows a multicast IPv6 neighbor discovery packet, which contains an ICMPv6 message (an echo request). As expected, the packet is sent to IPv6 multicast address `0xFF02::1`, and the frame is sent to the address beginning `0x33:33`.

QUESTIONS FOR READERS

Figure 16.10 shows some of the concepts discussed in this chapter and can be used to help you answer the following questions.

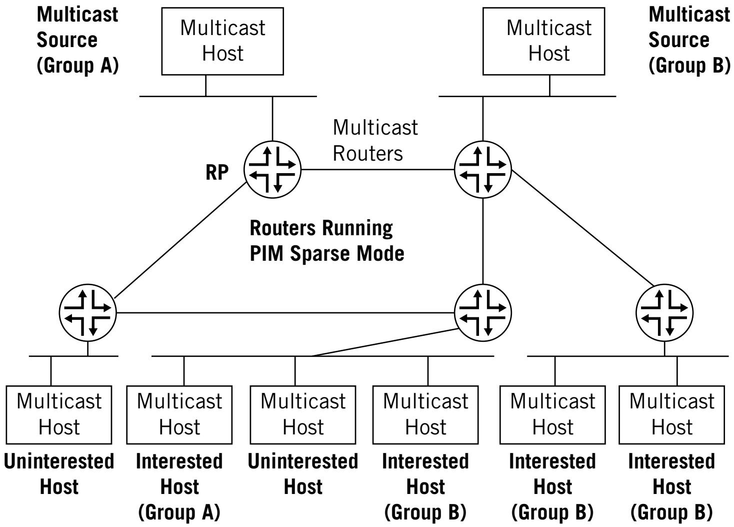

Multicast Source (Group A)
Multicast Host
Multicast Host
Multicast Source (Group B)
Multicast Routers
RP
Routers Running PIM Sparse Mode
Multicast Host
Uninterested Host
Multicast Host
Interested Host (Group A)
Multicast Host
Uninterested Host
Multicast Host
Interested Host (Group B)
Multicast Host
Interested Host (Group B)
Multicast Host
Interested Host (Group B)

FIGURE 16.10

A group of routers running PIM sparse mode with sources and receivers.

1. Generally, it is a good idea for RPs to be centrally located on the router network. Why does this make sense?
2. In Figure 16.10, does the rightmost host, which is interested in Group B content, have to get it initially from the RP when the source is closer?
3. Would the RP be required if the routers were running PIM dense mode?
4. Will the leftmost router with the uninterested host constantly stream multicast traffic onto the LAN anyway?
5. Is the uninterested host on the LAN in the middle able to listen in on Group A and Group B traffic without using IGMP to join the groups?