CHAPTER 10

IP MULTICAST

This chapter is about *IP multicast*, the network layer mechanisms in the Internet to support applications where data is sent from a sender to multiple receivers. The first section provides an introduction to multicasting in a packet-switched network, and gives an overview of the multicast service in the Internet. The second section highlights the protocol issues for IP multicast in a local area network. The third and fourth sections discuss multicast routing. The third section provides an overview over multicast routing protocols, and the fourth section covers the Protocol Independent Multicast (PIM) routing protocol. The fifth section reviews the issues of interdomain multicast routing. The sixth section is about the configuration of IP multicast in IOS on Cisco routers. The last section of this chapter presents the software tools needed for Lab 10.

TABLE OF CONTENTS

1. Multicast in the Internet

Some applications require data to be delivered from a sender to multiple receivers. Examples of such applications include audio and video broadcasts, real-time delivery of stock quotes, and teleconferencing applications. A service where data is delivered from a sender to multiple receivers is called multipoint communication or *multicast*, and applications that involve a multicast delivery service are called *multicast applications*.

Figure 10.1 compares multicast to other communication paradigms. In unicast or point-to-point communication, data is sent to a single host. In broadcast or one-to-all communications, data is transmitted to all hosts with respect to a given scope, for example, all hosts in a LAN. Multicast can be thought of as a generalization of unicast and broadcast. In multicast, data is transmitted to a set of hosts that have indicated interest in receiving the data, referred to as a *multicast group* or *host group.*

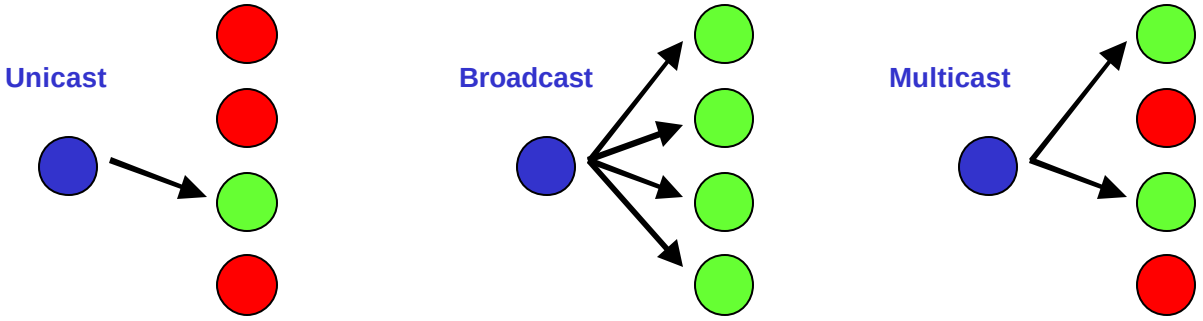


Figure 10.1. Unicast, Broadcast and Multicast.

In principle, it is feasible to implement multicast in a network using either unicast or broadcast. However, both solutions have shortcomings. In a unicast solution to multicast, the sender transmits one copy of the data separately to each host in the multicast group. This is viable for small multicast groups, but when the number of hosts is large, transmitting the same data multiple times wastes a lot of resources. In a broadcast solution to multicast, data is delivered to all hosts in a network; for example, and hosts drop the data if they are not members of the multicast group. This solutions works when the hosts of a multicast group are all located on the same LAN and the LAN supports broadcast transmissions. Otherwise, sending data to a large number of hosts just to have it dropped by most hosts is not an economical use of network capacity.

Making multicast delivery efficient in a packet switching network requires a whole set of new protocols and mechanisms at the network layer. First, multicast addresses must be available that can designate a multicast group as the destination of a datagram. Second, there must be mechanisms that allow a host to join and leave a multicast group. Third, there is a need for multicast routing protocols that set up paths, called *distribution tree*, from the sender to the

members of a multicast group. The issues related to setting up multicast distribution trees are referred to as *multicast routing*.

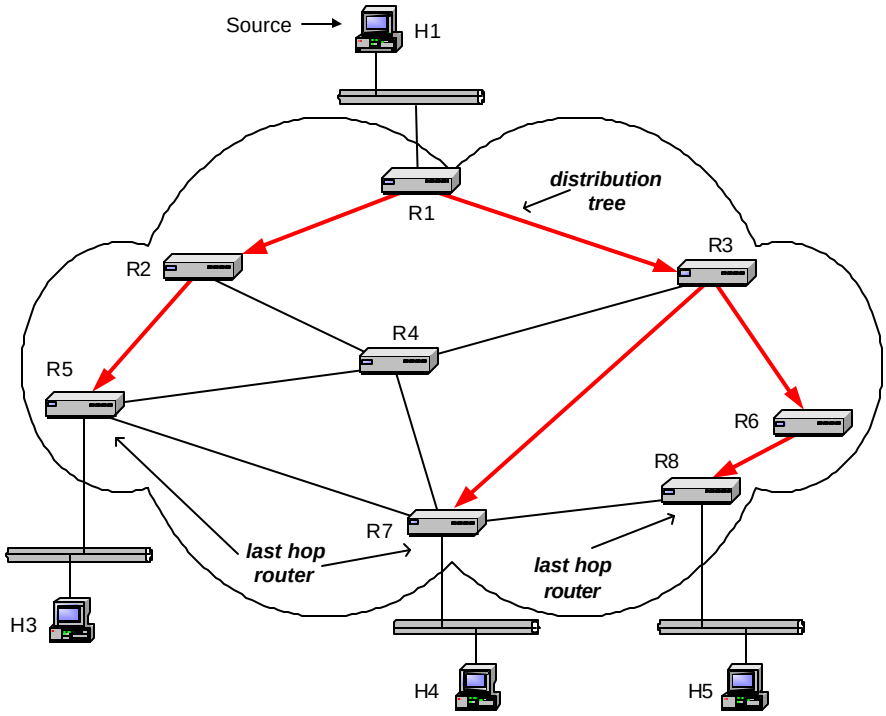


Figure 10.2. Multicast delivery in an IP network.

The network-layer mechanisms in the Internet that support multicast are referred to as *IP multicast*. Figure 10.2 depicts an example of multicast in an IP network. The figure shows four hosts and eight routers. Routers are connected by point-to-point links and hosts connect to routers via Ethernet LANs. Host H1 is a source of multicast data, and hosts H2, H3 and H4 are multicast receivers. The distribution tree, indicated with arrows, is established by the routers using a multicast routing protocol. Data is delivered downstream in the distribution tree from the source to the receivers.

IP multicast involves both hosts and routers. In IPv4, support of IP multicast is optional, but almost all hosts and most routers support multicast. Hosts that are members of a multicast group exchange Internet Group Management Protocol (IGMP) messages with routers. Routers perform two main processes in IP multicast: multicast routing and multicast forwarding. Multicast routing sets up the distribution tree for a multicast group by setting the content of

multicast routing tables. In multicast, a routing table may list multiple next hop addresses for a routing table entry. As in unicast, forwarding refers to the processing of an incoming datagram, the routing table lookup, and the transmission on an outgoing interface. When a multicast packet arrives at a router, the router performs a lookup in the multicast routing table for a matching entry. The router forwards one copy of the packet to each next hop address in the matching routing table entry.

1.1. IP Multicast service

A central notion of IP multicast is the IP multicast address. A multicast address is an IP address in the range from 224.0.0.0 to 239.255.255.255. Each multicast address specifies a multicast group. A host joins a multicast group by starting to listen to packets sent to the IP address associated with the multicast group. There is no separate mechanism for creating an IP multicast group.

IP multicast groups are open and dynamic, in the sense that every host can join and leave any multicast group. In many ways, IP multicast is similar to broadcast radio. A multicast address can be thought of as a radio frequency. Joining a multicast group is similar to tuning to a radio frequency, and there is no access control for joining a multicast group. The size of a multicast group is not limited. A host can join multiple multicast groups at a time, and on each host there may be multiple applications that receive packets sent to an IP multicast address.

A multicast packet is an IP datagram that has an IP multicast address as destination address. A multicast packet is delivered to all hosts that are members the multicast group designated by the IP multicast address. The semantics of IP multicast is similar to that of IP unicast, that is, IP multicast provides a connectionless best-effort datagram delivery service. A host can transmit multicast packets to any multicast group. A host does not need to be a member of a multicast group to transmit multicast packets to that group. Also, multiple hosts can send to a multicast group at the same time.

A notable difference between multicast and unicast on the Internet is that only UDP is available as transport protocol for multicast applications. There is no multicast version of TCP. Figure 10.3 shows that that unicast applications can use TCP and UDP, whereas multicast applications must use UDP as transport protocol. This means that, at the transport layer, the Internet does not offer a reliable or in-sequence delivery service for multicast traffic. For applications that require a reliable data transfer, for example, one-to-many file transfer applications, the mechanisms that ensure a reliable data transfer, such as sequence numbers, timers and retransmissions, must be provided by the application layer.

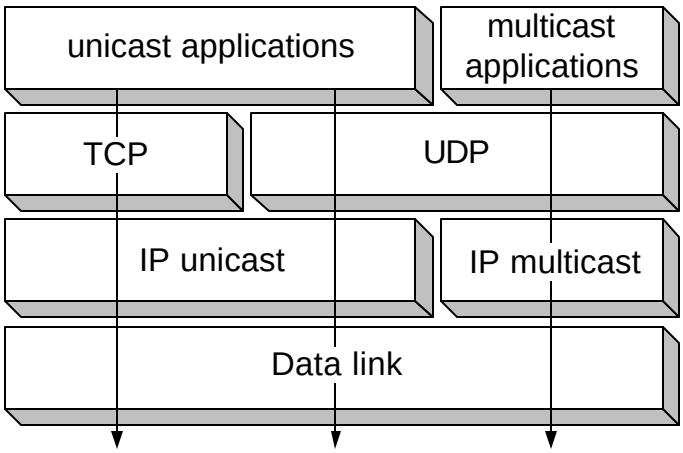


Figure 10.3. Protocol layers for unicast and multicast.

1.2. IP Multicast Addresses

The range of IP multicast addresses, 224.0.0.0 to 239.255.255.255, corresponds to the Class D addresses in the class-based IP address scheme. These are the addresses that start with *"1110"* (Figure 10.4). This leaves 28 bits to identify a multicast group.

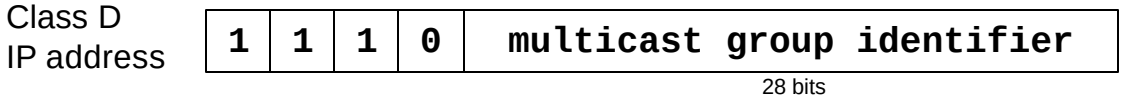


Figure 10.4. Multicast IP address.

Multicast addresses designate either permanent or transient multicast groups. A permanent multicast group is associated with an IP multicast address that is registered with IANA.[2] Important permanent multicast groups are the *all hosts* group with address 224.0.0.1, which designates all hosts and all routers on a network, and the *all routers* group with address 224.0.0.2, which designates all routers on a network. Several protocols reserve permanent multicast groups. For example, the IP address 224.0.0.9 is used by RIP2 protocol to designate all RIP2 routers on a network, and 224.0.0.4 is the group of all routers that use the DVMRP multicast routing protocol.

Addresses with the prefix 233.0.0.0/8 are reserved for owners of autonomous systems[3] and are used as follows. The AS number of an autonomous system is converted into a 16 bit number,

[2] RFC 1700

[3] RFC 3180

and these 16 bits are assigned to the second and third byte of the IP address. Then each autonomous system can use the fourth byte to designate a set of 256 reserved multicast addresses. For example, if the AS number of an autonomous system is 0x8080, the autonomous system effectively owns the multicast addresses in the range from 233.128.128.0 to 233.128.128.255.

All addresses that do not designate permanent multicast groups designate transient groups. Addresses for these groups are assigned dynamically. An application implicitly allocates a multicast address by starting to transmit to a transient group. The address is released implicitly when there is no sender and no receiver for this group. If two applications use the same multicast address, IP multicast treats the applications as a single multicast group, and packets sent by any of the applications are delivered to both applications.

Since the TTL field in the IP multicast datagrams limits the scope of an IP datagram, it is feasible that different applications use the same IP multicast group, without interfering with each other. For example, if a host transmits to multicast group 230.1.1.1 and sets the TTL field to 2 in each IP datagram, the transmission will not be seen by any host or router that is more than two hops away. Problems occur only if the scope of the IP datagrams transmitted by different applications that use to the same multicast group overlap. The addresses in the range 224.0.0.0- 224.0.0.255 are always sent with the TTL field set to one, and, therefore, the scope of the corresponding groups are limited to the local network.

2. IP Multicast in a Local Area Network

We first explore IP multicast in a local area network. The main issues here are the interactions of hosts and routers when hosts join and leave a multicast group, and the address translation between multicast IP addresses and multicast MAC addresses. We also point out to problems when IP multicast traffic is sent over a switched Ethernet network.

2.1. IGMP

The *Internet Group Management Protocol (IGMP)* helps routers to keep track of the multicast membership of hosts on the directly connected networks of the router. When a host joins or leaves a multicast group it informs a router that it wishes to receive multicast packets for a given multicast address. When a router learns via IGMP that a host on a connected network has joined a multicast group, the router uses a multicast routing protocol to get connected to the distribution tree for this multicast group. When multicast packets for a group arrive at the router, the router forwards the multicast packets to the connected networks where hosts have joined the group. A router does not keep track of the identity or even the number of hosts that have joined a multicast group. A router simply keeps a record if there is at least one multicast group network on a directly connected network.

The operations of IGMP are as follows. When a host joins a multicast group, it transmits a membership report message, which is received by the router. The router, on the other hand, periodically sends out queries to determine if there are hosts that are still members in the multicast group. Hosts, that are members of a multicast group, respond to a query with a membership report. If a network has multiple multicast group members, only one host responds. If hosts on a network do not respond to a query, the router assumes that no host is listening, and, eventually, stops forwarding multicast packets on this network. In certain situations a host sends a message to the router indicating that it is leaving a multicast group.

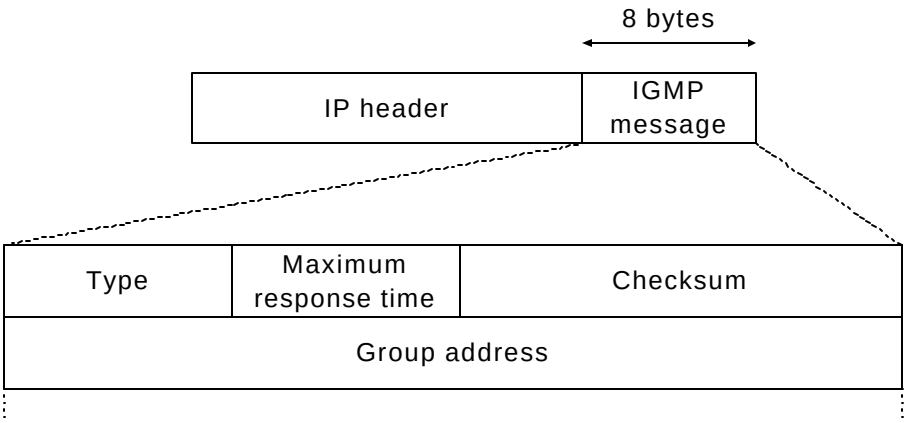


Figure 10.5. IGMPv2 message format.

Here, we discuss the IGMP Version 2 (IGMPv2)[4], which is currently the most widely used version of IGMP. IGMP messages are 8 bytes long and have a format as shown in Figure 10.5. IGMP messages are sent in the payload of IP datagrams with a protocol number set to 2. The first field of the IGMP message identifies the type of the message as a *membership query* (0x11), a *membership report* (0x16), or a *leave group message* (0x17). There are two subtypes of membership queries: general queries and group-specific queries. A router sends a general query to find out which multicast groups have members. A group-specific query is used to find out if a specific multicast group has a member. The subtype is determined by looking at the content of the group address field. In a general query, the group address is set to zero, and in a group specific query it contains a multicast IP address. The maximum response time field in an IGMP message is used in a membership query message. It specifies the maximum time until a report must be sent in response to the query. The time is specifies in multiples of $1/10^{th}$ of a second. The checksum field in an IGMP message is set in the same way as in an ICMP message. The group address field contains the IP multicast address or, in the case of a general query, is set to zero. IGMP messages are sent with the TTL field in the IP header set to one. Therefore, IGMP messages are never forwarded by routers.

[4] RFC 2336

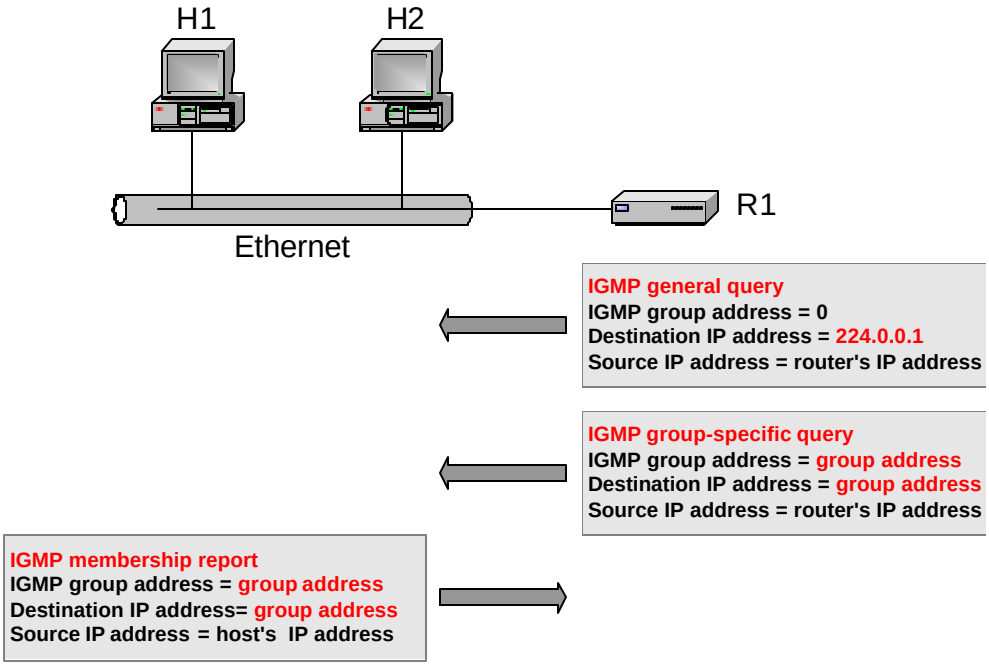


Figure 10.6. IGMP messages between routers and hosts.

Periodically, by default every 125 seconds, a router sends an *IGMP general query* to all hosts on each of the networks that the router is connected to. The query is sent by setting the destination IP address to the *all hosts* group with IP address 224.0.0.1. If a host is a member of multicast groups it responds to an *IGMP query* with an *IGMP membership report*. To prevent duplication of reports, each host waits a random amount of time before sending a membership report. If, during the waiting time, another host sends a report for the same group, the host suppresses the transmission of its report. In this fashion, generally only one host responds to a general query.

A host always sends an *IGMP membership report* when it joins a multicast group. If multiple applications on the same host join the same multicast group, an IGMP report is sent only for the first application that joins. A host with multiple network interfaces can join a multicast group on any of its interfaces.

When the host that has sent the last membership report for a multicast group leaves the group, it sends an *IGMP leave* message to the router. When the router receives an IGMP leave message it sends an *IGMP group-specific query* on the network where the leave message was received. If the router does not receive a response to the query, it assumes that that there are no group members for this group, and stops forwarding multicast packets for that group. A host does not send a *leave* message if it was not the last host to send a membership report.

When a network has multiple multicast routers only one router responds to IGMP queries. This router is called the *querier*. The rule is that that the multicast router with the smallest IP address becomes the querier on a network. If a router observes IGMP query reports on a network that have a source IP address smaller than its own IP address, the router stops sending IGMP queries. The querier is not necessarily the same router that forwards multicast packets to the network, called the *forwarder*. In fact, the roles of the querier and the forwarder are often assigned to different routers. The division of labor between the querier and the forwarder is illustrated in Figure 10.7.

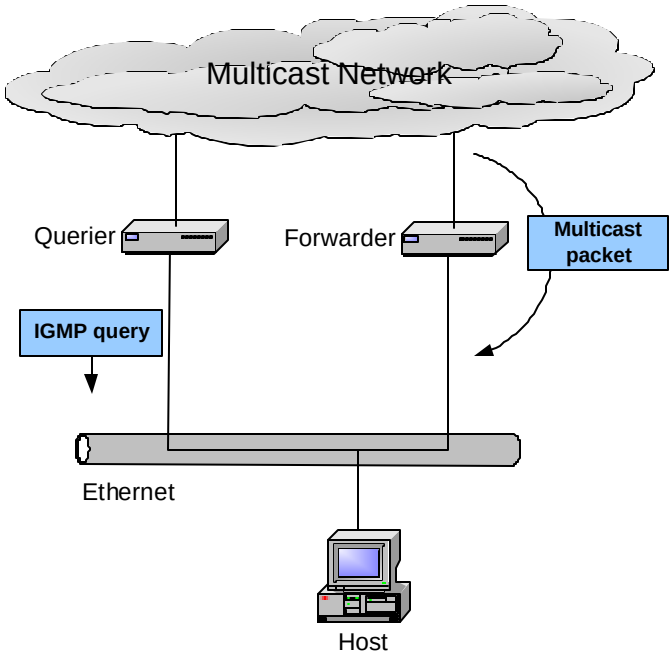


Figure 10.7. Querier and forwarder in a network with multiple routers.

2.2. Multicast Address Resolution

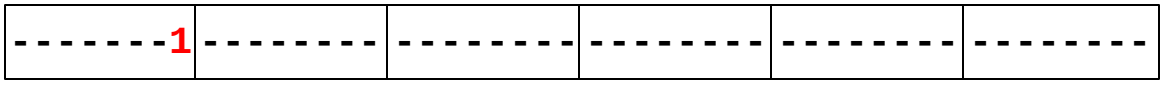


Figure 10.8. Multicast MAC address.

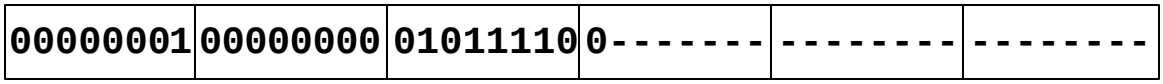


Figure 10.9. Multicast MAC address allocated for IP multicast.

On Ethernet networks or other networks that use MAC addresses, an IP multicast address must be translated into a MAC address. Just as in IP, a certain range of MAC addresses are reserved for multicast transmissions. Specifically, all MAC addresses that have the low order bit of the first byte of the address set to one are multicast MAC addresses (see Figure 10.8). Just like unicast MAC addresses, organizations can reserve blocks of multicast addresses from the IEEE. IANA has reserved the MAC address block from 01-00-5e-80-00-00 through 01:00:5e:ff:ff:ff, and has reserved the first half of this block, namely, addresses in the range 01-00-5e-80-00-00 through 01-00-5e-7f-ff-ff to encapsulate IP multicast datagrams. Thus, Ethernet frames where the destination MAC address have the first 25 bits set as shown in Figure 10.9 encapsulate IP multicast datagrams.

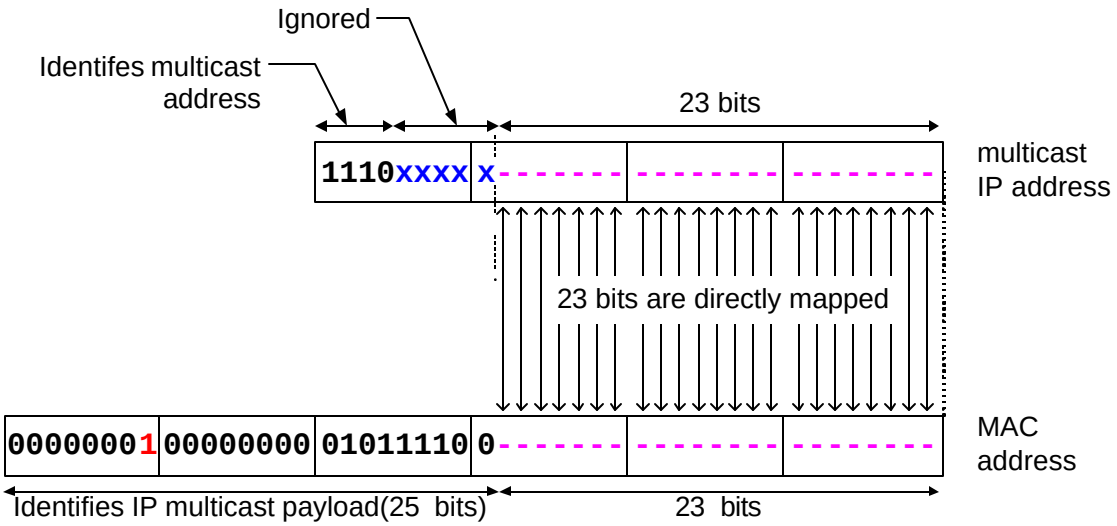


Figure 10.10. Mapping multicast IP address to multicast MAC address.

Different from unicast, where the translation of an IP address to a MAC address is done dynamically by ARP, the mapping from a multicast IP address to a multicast MAC address is static. The mapping is done by assigning the low-order 23 bits of the multicast IP address to fill the low-order 23 bits of the multicast MAC address. The mapping is shown in Figure 10.10. The first 4 bits of the IP address identify an IP multicast address, the following five bits, indicated by *x*'s, are ignored. And the last 23 bits are directly mapped into the MAC address.

When a host joins an IP multicast group, it instructs the data link layer to receive frames that match the MAC address that corresponds to the IP address of the multicast group. The data link

layer filters the frames and passes frames with matching destination addresses to the IP module. The filtering can be done in hardware on the network interface card, or in software by the device driver.

Since the mapping from multicast IP address to a MAC address ignores 5 bits of the IP address, groups of 32 multicast IP addresses are mapped to the same MAC address. As a result a multicast MAC address cannot be uniquely mapped to a multicast IP address. On the other hand, a host that receives a frame can determine the multicast IP address by inspecting the IP destination address in the IP header.

2.3. IP Multicast in Switched Ethernet Networks

An interesting problem arises when multicast traffic is transmitted over an Ethernet switch. Recall from Chapter 5, that an Ethernet switch forwards a unicast frame on a port where the destination MAC address is located. If the port of the destination address is not known, the frame is transmitted to all ports. An Ethernet switch learns about the port location of MAC addresses by snooping the source MAC address of frames. However, this does not work for multicast frames, since multicast addresses do not appear as source addresses. Therefore, an Ethernet switch always forwards multicast frames to all ports, thereby flooding a multicast frame to all Ethernet segments. As a result, in a network of Ethernet switches, multicast traffic is effectively handled like a broadcast frame.

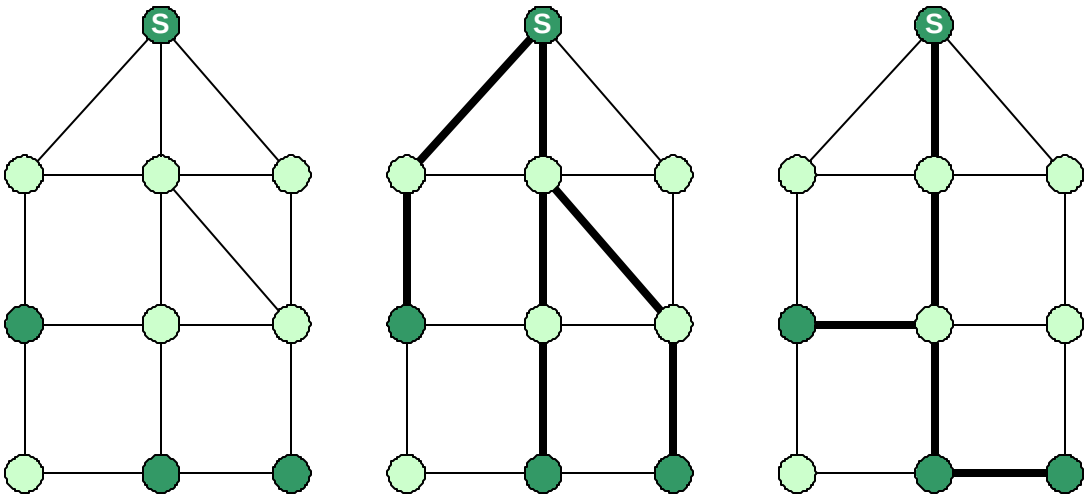
There are several solutions to prevent flooding of multicast frames at Ethernet switches. The simplest solution is to make an Ethernet switch aware of IGMP. Here, an Ethernet switch keeps track of the IGMP messages that are received on a port, and forwards a frame with a multicast MAC address only to those ports where an IGMP message has been received for the IP multicast group that corresponds to the destination MAC address in the frame. This technique, known as *IGMP snooping*, requires that a switch examines each multicast frame.

Cisco has developed a proprietary protocol, called *Cisco Group Management Protocol (CGMP)*, which is implemented at routers and Ethernet switches. In CGMP, a router sends Join messages to an Ethernet switch, when it receives an *IGMP membership report*. The router sends a *CGMP Leave* message to an Ethernet switch, when the router has received an *IGMP leave* message. The Join and Leave messages are sent the multicast MAC addresses that correspond to an IP multicast address. Hence, a switch does not need to parse the IP and IGMP information in an Ethernet frame.

3. Multicast Routing

Multicast routing is concerned with setting up distribution trees that provide a path from the multicast sources to multicast group members. This section discusses the objectives of multicast routing, and the protocol mechanisms used in routing protocols.

3.1. Multicast Routing Algorithms in a Graph



(a) Network graph (b) Shortest-path tree (c) Minimum-cost tree

Figure 10.11. Objectives of multicast routing.

Before discussing the routing protocols of the Internet in detail, let us look at multicast routing as an algorithmic problem. We view an IP network as a graph as shown in Figure 10.11(a), where each router is represented as a node and each link between two routers is represented by an edge. Each edge is assigned two cost metrics, for traversing the edge in one or the other direction. For simplicity, we assume that all edge costs are set to one. Routers that are connected to multicast group members have a dark shade. In addition, the router that connects to the source is labeled with an `*S*'. We assume that there is one multicast source, and that the source is also a member of the multicast group.

In this graph, the task of multicast routing is the embedding of a tree into the graph such that all multicast group members are connected by the tree. So, how would an ideal embedding look like? One can think immediately of two objectives to build a *good* tree, which are shown in Figure 10.11(b) and Figure 10.11(c). In the figures, the thick lines indicate the distribution trees. The first objective is to build a tree that minimizes the path cost from the source to each receiver. Such a tree is called a *shortest-path tree* or *source-based tree* (Figure 10.11(b)). A shortest-path tree can be relatively quickly computed using a shortest-path algorithm, but has the drawback that the tree is dependent on the source of the multicast tree. Note in Figure 10.11(b), that the shortest-path tree is different when a different node is the source in the multicast group. Therefore, a distribution tree must be computed separately for each source.

The second objective is to build a tree that minimizes the total cost of the edges, called a *minimum-cost tree* (Figure 10.11(c)). Different from a shortest-path tree, a minimum-cost tree

does not change when a different node becomes the source. Thus, the same tree can be shared by all sources. The main drawback of a minimum-cost tree is that its computation is prohibitively expensive in most cases. In fact, the problem of calculating a minimum-cost tree is known to be an NP-complete problem, meaning that the computation of the tree is intractable unless the network is small. The two objectives for multicast routing, shortest-path tree and minimum-cost tree, represent a set of trade-off for multicast routing. Shortest-path trees minimize the cost of each receiver, whereas the minimum-cost tree minimizes the cost of the total tree. A single minimum-cost tree can be shared by all senders, whereas, a shortest-path trees must be built for each source. Lastly, a shortest-path tree can be computed relatively quickly, whereas the computation of a minimum cost-tree is not tractable in large networks.

Some of the above trade-offs of multicast routing are reflected in the multicast routing protocols for the Internet. However, routing protocols have to satisfy additional constraints, which are not expressed in the formulation of a graph problem. For example, routing protocols must be able to adapt the distribution tree quickly when hosts join and leave a multicast group. Additionally, most multicast routing protocols require computing the distribution tree in a distributed fashion without any central coordination. Finally, most multicast routing protocols try to leverage off unicast routing protocols, by constructing the distribution tree using information from the unicast routing tables. This imposes additional constraints on a multicast routing protocol.

3.2. Reverse Path Forwarding

The majority of routing protocols in the Internet built either source-based trees or core-based trees. A source-based tree is essentially a shortest-path tree, with a multicast source at the root of the distributions tree. With source-based trees, one distribution tree must be built for each multicast source. Protocols that built core-based trees avoid the need for multiple distribution trees by building a single distribution tree that is shared by all sources. Core-based tree routing protocols do not attempt to construct a minimum-cost tree. Instead, one router is selected as *core* or *rendezvous-point*, and a shortest-path tree is constructed with the core router as the root of the tree. Thus, both source-based trees and core-based trees build shortest path trees. For reasons that will become clear in a moment, routing protocols generally minimize the paths from the receivers to the source, as opposed to minimizing the path from the source to the receiver.

A reverse shortest path tree can be built by applying a concept called *reverse path forwarding (RPF).* RPF is a mechanism to build a shortest path tree in a distributed fashion by taking of the unicast routing tables. The idea of RPF is simple. Given the address of the root of the tree, a router selects as its upstream neighbor in the tree the router which is the next-hop neighbor for forwarding unicast packets to the root. The network interface which is used to reach this upstream neighbor is called the *RPF interface* and the neighbor router is called the RPF *neighbor*. An illustration of the basic concept of reverse path forwarding is given in Figure 10.12. Here, host H1 is the root of the distribution tree. Router R3 determines that R2 is its RPF neighbor for H1 from a lookup in its routing table. The interface that connects to R2 is the RPF interface.

If all routers determine their upstream neighbor using reverse path forwarding, the resulting distribution tree is such the packets from the root to a receiver are forwarded on the reverse path taken by unicast packets sent from the receiver to the root. This is where the name *reverse path forwarding* comes from. Since unicast routing tables are set so that the path from the receiver to the root is minimal, the tree generated by reverse path forwarding is a *reverse shortest path.* Reverse path forwarding is applied to construct source-based, as well as core-based trees. In the former case, the root of the tree is a multicast source. In the latter case, the root of the tree is the core.

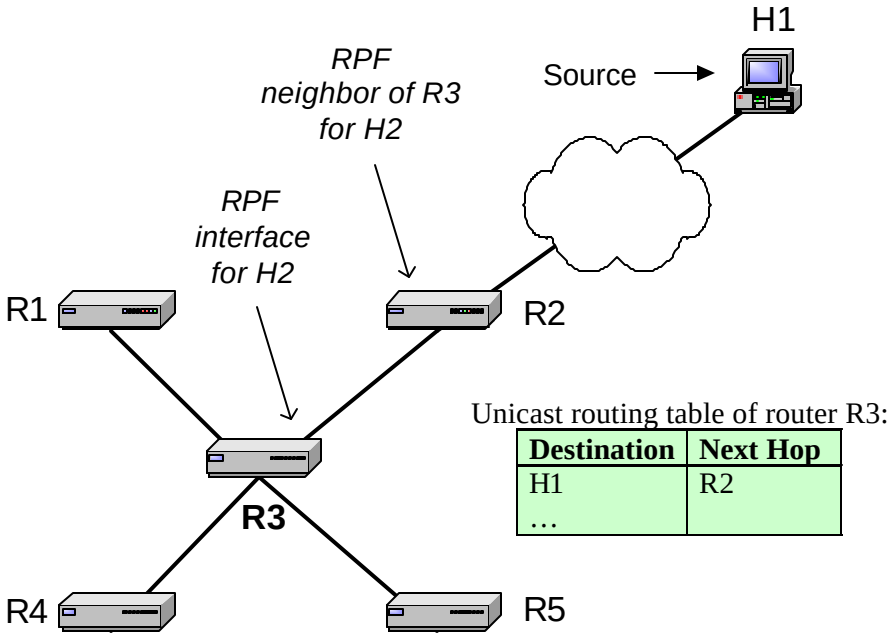


Figure 10.12. Reverse path forwarding.

The general structure of a multicast routing table is shown in Figure 10.13. There are columns for a source IP address, a multicast address, an incoming interface and a list of outgoing interfaces.

Routing table entries for source-based trees and for core-based trees are different. In a source-based tree, a routing table entry contains a source address as well as a group address. This is often called a (Source, Group) or (S, G) entry. Since a source-based tree has one distribution tree for each source there may be multiple routing entries for the same multicast group. Routing table entries for a core-based tree do not list a source IP address, which is indicated with a '*' (star) in the source IP address column. The corresponding entry is called a (*, G) entry. In an (S, G) entry, the incoming interface entry is the RPF interface for address S. In an (*, G) entry, the incoming interface is set to the RPF interface with respect to the core.

An arriving multicast packet must match the source address and the group address in an (S, G) entry, and the group address in an (*, G) entry. If there are matches for both (S, G) and (*, G) entries, the (S, G) entries take preference. When a match is found in the routing table, the router verifies that the packet arrived on the incoming interface listed in the incoming interface column of the routing table. This is called an *RPF check*. If an RPF check is successful, one copy of the datagram is forwarded on each interface listed in the outgoing interface list. If there is no match in the routing table or if the RPF check failed, the packet is discarded.

Source IP address	Multicast group	Incoming interface (RPF interface)	Outgoing interface list
S1	G1	I1	I2, I3
*	G2	I2	I1, I3

Figure 10.13. General structure of a multicast routing table.

3.3. Source-based trees

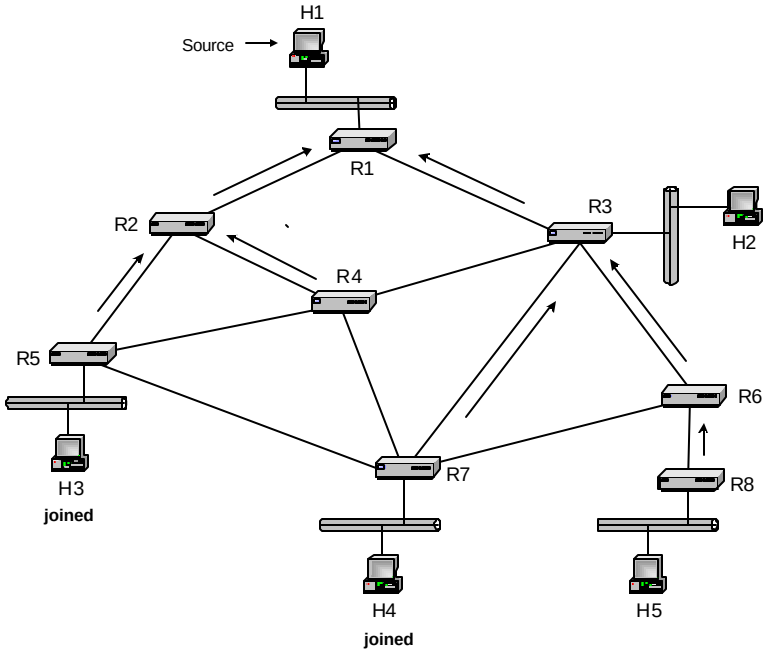


Figure 10.14. Multicast group with RPF forwarding tree.

We illustrate the construction of a source-based tree for the network shown in Figure 10.14. Each host is connected to a router via an Ethernet network and routers are connected by point-to-point links. H1 is the source of a multicast group and hosts H3 and H4 are receivers that have joined the multicast group. The arrows in the figure indicate the reverse shortest paths for all routers in the network. Data transmissions in the source-based distribution tree for H1 follow

the opposite direction of the reverse shortest paths. The paths for H3 and H4 are H1→ R1→ R2 → R5 → H3 and H1→ R1→ R3 → R7 → H4, respectively.

Since the RPF interface tells each router the upstream neighbor in the distribution tree, but not the downstream neighbors in a distribution tree, additional protocol mechanisms are needed to determine the outgoing interfaces in the multicast routing tables. One method to achieve this is *flood-and-prune*, which starts out by forwarding multicast packets on all its interfaces, and then deletes interfaces which are not part of the distribution tree. Another method, called *explicit join*, requires that multicast receivers initiate the process of getting connected to the distribution tree. We will describe both methods.

In flood-and-prune, the outgoing interface list in a routing table entry initially lists all interfaces other than the RPF interface. When a router receives a multicast packet from source address S for a multicast group G, on the RPF interface for source S, it forwards the packet on all interfaces, with exception of the RPF interface. When a router receives a packet on an interface that is different from the RPF interface for S, it discards the packet. Figure 10.15 shows how multicast packets from H1 are forwarded with this method. Arrows indicate the transmission of packets and crosses show where packets are discarded. Figure 10.15 shows that due the flooding of messages, many routers receive multiple copies of the same packet. All but the copy arriving on the RPF interface are discarded.

A router connected to a LAN, forwards a datagram to the LAN only if some hosts on the LAN are multicast group members. In Figure 10.15, these are the LANs where the multicast group members H3 and H4 reside. A router knows about group members on a LAN from IGMP messages.

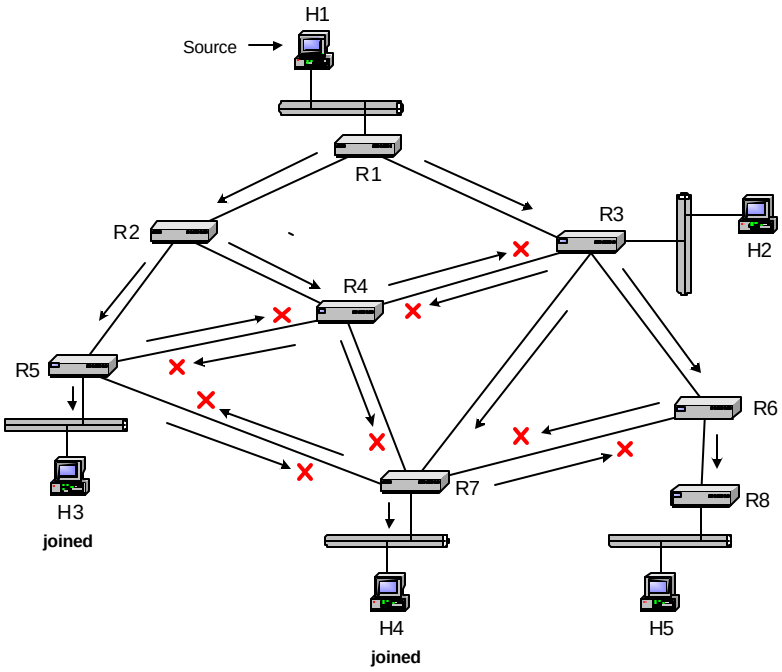


Figure 10.15. Flood-and-prune.

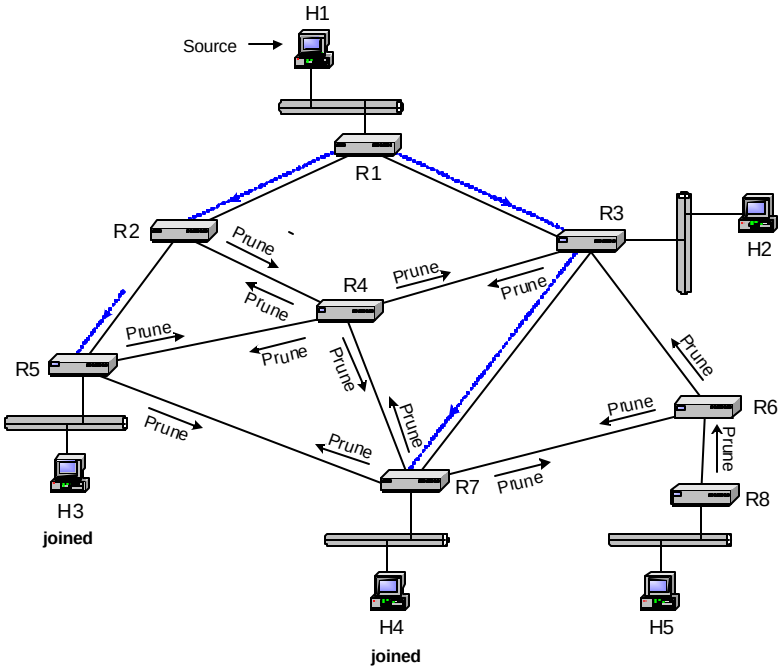


Figure 10.16. Prune messages.

The second phase of the flood-and-prune method ensures datagrams are delivered to a router only if the router is on a path to a multicast group member. This second phase is called *pruning* and works as follows. Whenever, a router receives a multicast packet on an interface that is not the RPF interface, the router sends a *prune* message on that interface. A router sends a prune message on its RPF interface if it is not connected to a LAN with group members and one of the following holds: (1) the router does not have a downstream neighbor in the distribution tree, or (2) the router has received prune messages from all downstream neighbors. When a router receives a prune message for an (S, G) entry on an interface, it stops forwarding multicast packets for group G from source S on that interface.

Figure 10.16 illustrates the transmission of prune messages. R2 sends a prune message to R4 since the packet from R4 is not the RPF for source H1. R8 sends a prune message because it does not have a group member in its local network, and it also does not have downstream neighbors. The prune message from R6 to R3 is sent because R6 has received prune messages from all downstream neighbors.

As a result of pruning, routers receive multicast messages only if they have group members in their connected LANs, or if they are on the reverse shortest path from a multicast group member to the source. Since a multicast group is dynamic and new hosts may join or leave the multicast group, prune message stop the forwarding on an interface only temporarily and reinstate a pruned link, and packets are forwarded over a previously pruned link. This may trigger the transmission of another prune message.

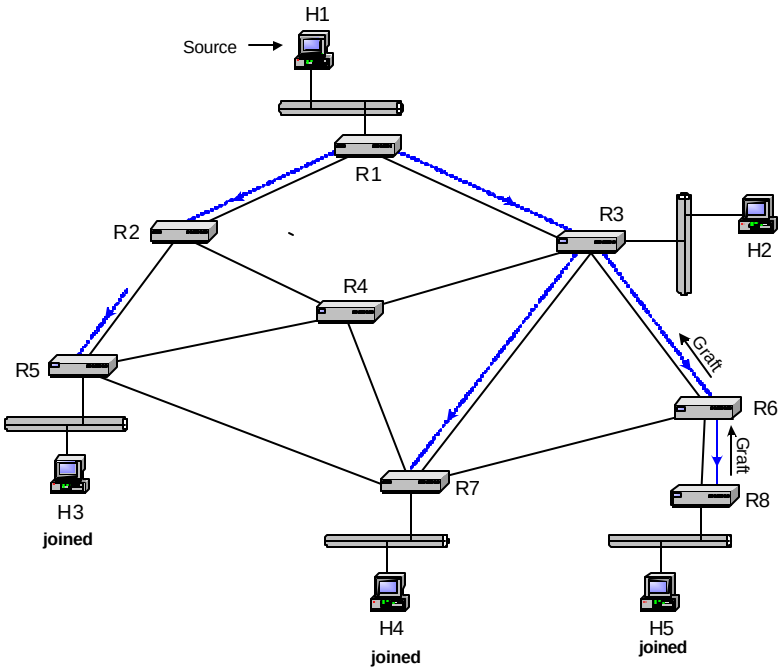


Figure 10.17. Graft messages.

Since the waiting time until the timeout of a pruned link may be long, a second method exists by which a pruned link is reinstated before the timeout of the pruned link. This is called *grafting*. A *graft* message is sent on the RPF interface and voids the pruning of a link. A router that receives a graft message forwards the graft message on its RPF interface, until a router is reached that is part of the distribution tree. The transmission of graft messages is shown in Figure 10.17, when host H5 joins the multicast group.

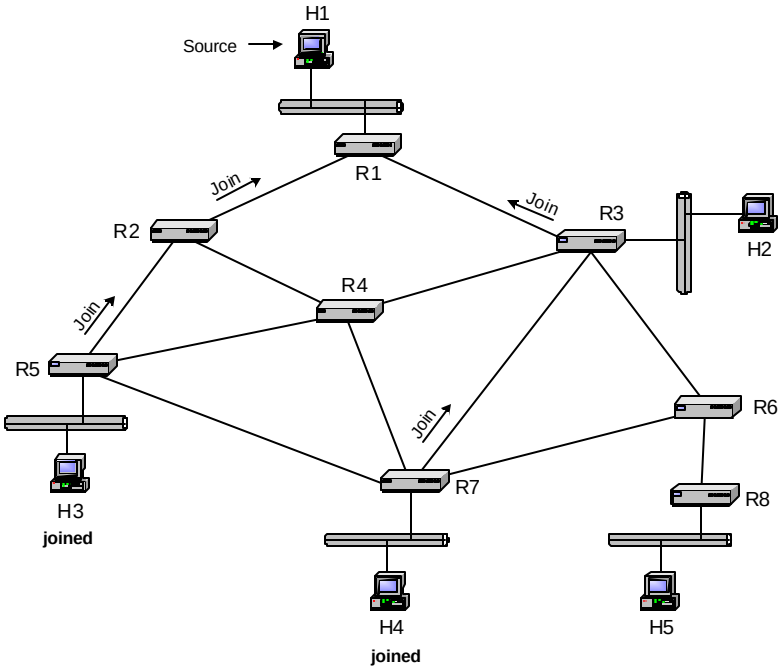


Figure 10.18. Source-based tree with explicit join.

An alternative method to build a source-based tree is to let receivers initiate the construction of links in the distribution tree. This method is called *explicit join*. Here, routers send *join* messages for an (S, G) entry on their RPF interfaces for S, similar to the *graft* messages in Figure 10.17. In Figure 10.18, when hosts H3 and H4 want to receive multicast packets from source S for group G, the connected routers, R5 and R7, send a join message to the router on their RPF interfaces. When a router receives a *join* message, it adds the interface where the *join* message was received to the list of outgoing interfaces of the (S, G) routing table entry. If an (S, G) routing entry does not exist, it will be created when the *join* message arrives. In Figure 10.18, when R2 receives a *join* message from R5, it adds the interface that connects to R5 to the outgoing interface list. As with *graft* messages, a router forwards a *join* message on its RPF interface only if the router is not part of the distribution tree. When a router in the distribution tree receives a *join* message, it does not forward the join message. This method of explicit join messages avoids the transmission of duplicate packets as in flood-and-prune. On the other hand, building source-based trees with explicit join messages assume that routers know the identity of hosts that transmit to a multicast group.

3.4. Core-based Trees

Routing protocols that built core-based trees designate a router, called the *core*, and built a reverse shortest path tree with the core as the root of the tree. The core becomes the central hub for disseminating multicast packets sent to the group. When a source transmits a packet to a multicast group, the packet is sent to the core. When the packet reaches the core, it is forwarded using the reverse shortest path tree.

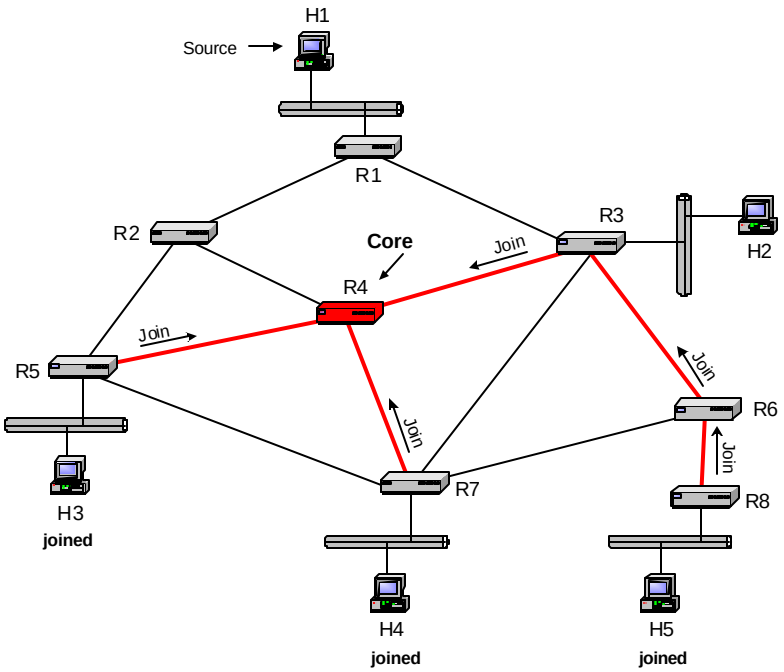


Figure 10.19. Construction of a core-based tree.

The construction of a core-based tree for a multicast group G is illustrated in Figure 10.19. Each receiver that joins a multicast group sends a *join* message to the core router of the group. The message is sent on the RPF interface with respect to the core. If the join message reaches a router that is not part of the tree, the router adds an (*, G) entry to the multicast routing table, adds the interface where the *join* message arrived to the outgoing interface list, and sets the incoming interface to the RPF interface. Then, the router forwards the *join* message on its RPF interface in the direction of the core. If the router is already part of the shared tree for group G, the router, upon receiving a *join* message, adds the interface where the *join* message arrived to the outgoing interface list of the corresponding (*, G) entry, but does not forward the *join* message. Since *join* messages are sent on the reverse shortest path, the resulting distribution tree is a reverse shortest-path tree rooted at the core. Figure 10.19 illustrates the transmission of join messages with router R4 as core, and hosts H3, H4, and H5 has members of the multicast group.

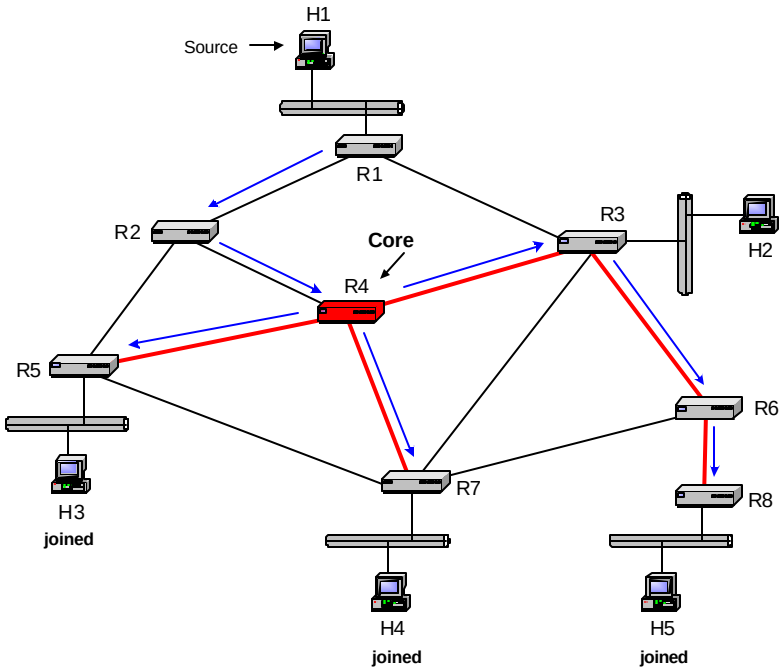


Figure 10.20. Multicast forwarding in a core-based tree.

Forwarding of multicast packets in core-based trees is illustrated in Figure 10.20. When H1 transmits a multicast packet, the packet is forwarded on the unicast shortest path to the core. A possible route is H1→ R2 → R4. When the packet reaches the tree at the core, it is sent downstream in the core-based tree. Sending packets always to the core and inserting packets into the distribution tree only at the core can be inefficient. For example, suppose the path from R1 to R4 is through R3. Then, a packet from H1 to H5 takes the route H1→ R1 → R3 → R4 → R3 → → R6 → R8. Here, it would be better to have R3, instead of forwarding the packet to the core, forward the datagram using the core-based tree. R3 should forward the datagram downstream the tree to R6, as well as upstream the tree to R4. However, such bidirectional trees, where packets are forwarded both upstream and downstream a core-based tree, are prone to routing loops, and special precautions are necessary to avoid such loops.

The main advantage of using core-based trees is that only one distribution tree is required for each multicast group. This reduces the complexity of the routing tables, as well as the volume of

multicast routing protocol messages. The main disadvantage of core-based trees is that, dependent on the placement of the core, the paths from the sender to the receivers through the core can be much longer than the direct path between a source and a receiver.

3.5. Overview of Multicast Routing Protocols

Here is a brief overview of the multicast routing protocols that have been developed for the Internet.

Distance Vector Multicast Routing Protocol (DVMRP):[5] DVMRP was the first multicast routing protocol developed for the Internet. DVMRP can operate in an environment where some, but not all routers in the network are capable of multicast forwarding and routing. This is achieved by having DVMRP run a separate unicast routing algorithm, similar to RIP, to determine the shortest-paths between all multicast-capable routers. DVMRP uses flood-and-prune to set up source-based trees. DVMRP messages are encapsulated in IGMP messages, where the type field is set to 3.

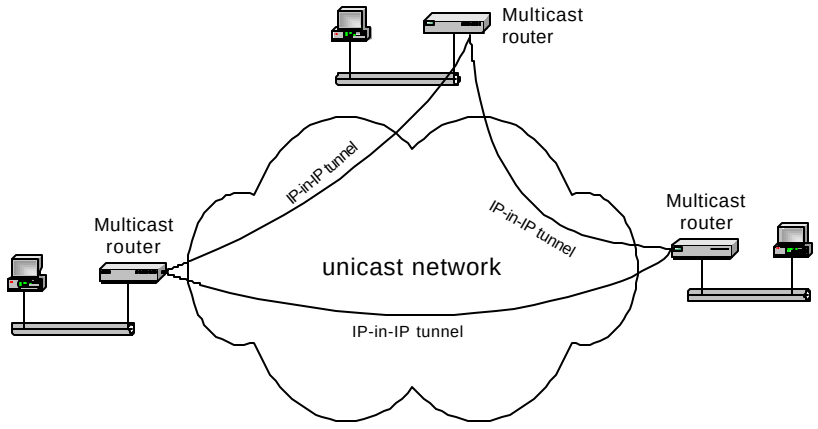


Figure 10.21. Tunnel-based topology in the MBONE.

DVMRP played an important role in the early deployment of IP multicast. IP multicast deployment in the Internet began in the early 1990s with the creation of the Multicast Backbone (MBONE). The multicast routing algorithm in the MBONE is DVMRP. The MBONE solved the problem of wide-area IP multicast routing on the Internet where only few routers were capable of IP multicast routing, by setting up a virtual network of multicast routers that are connected by unicast path. These multicast routers exchanged multicast IP datagram that were encapsulated in IP unicast datagrams, using the IP-in-IP option in the IP header, as shown in

[5] RFC 1075

Figure 10.21. As a result of the encapsulation, the MBONE is a virtual network, where each link between two multicast routers consists of a complete unicast path. As more and more routers provide native support for IP multicast, meaning that they are capable of forwarding IP multicast traffic and running a multicast routing protocol, the need for a virtual multicast network has all but disappeared.

Multicast Open Shortest Path First (MOSPF):[6] MOSPF consists of multicast extensions to the unicast routing protocol OSPF, and requires that OSPF is used for unicast routing. In MOSPF, multicast routers broadcast link state advertisements (LSAs) to all other multicast routers. Then, as in unicast OSPF, each multicast router calculates routes independently. MOSPF computes shortest-path trees for each sender in the multicast group. A router computes a shortest-path tree for a source only if there is traffic from that sender.

Core Based Tree (CBT):[7] CBT was the first routing protocol for the Internet that took a core-based tree approach. CBT builds a shared tree using reverse-path forwarding, without making assumptions on the unicast routing protocol used. The core of a group is either statically configured, or determined as the outcome of a selection process from a candidate set. Different multicast groups may use different core-bases trees. Distribution trees in CBT are bidirectional, that is, routers are capable of forwarding multicast packets downstream away from the core as well as upstream towards the core.

Protocol Independent Multicast (PIM):[8] Protocol independent multicast consists of two multicast routing protocols: *PIM Dense Mode* (PIM-DM) and *PIM Sparse Mode* (PIM-SM). PIM-DM builds source-based trees using flood-and-prune, and is intended for large multicast groups where most networks have a group member. PIM-SM builds core-based trees as well as source-based trees with explicit joins. PIM-DM and PIM-SM, respectively, are in several aspects similar to DVMRP and CBT. Just like CBT, PIM can operate on top of any unicast routing protocol, hence the name *protocol independent multicast.* A consequence of this is that PIM must assume that all routers in the network are multicast enabled. An important difference between the core-based trees of PIM and CBT is that the trees in PIM are unidirectional, that is, sources always forward packets to the core, and the core transmits packets downstream the core-based tree.

All of the above multicast routing protocols follow the service model of IP multicast where any host can transmit to all multicast groups. Since this service model makes multicast routing complex, recently there has been support for a source-specific multicast service model, where hosts join a multicast group separately for each source. This service model is called *Source-Specific Multicast (SSM)*. SSM requires that a host, when it joins a multicast group G, also

---

[6] RFC 1584

[7] RFC 2189, RFC 2201

[8] RFC2362

specifies the source S from which it wishes to receive multicast packets. This can be accommodated by IGMPv3[9], a recently completed new revision IGMP, which allows hosts to join a multicast group for specific sources. SSM does not require new multicast routing protocols. In fact, routing for SSM can be realized with a subset of most existing protocols.

4. Protocol Independent Multicast

This section provides an overview of the PIM version 2 (PIMv2) multicast routing protocol., which is available on Cisco routers. Both PIM-DM and PIM-SM will be encountered in the lab exercises of Lab 10. The presentation leaves out many details of PIM, but provides enough information to master the lab exercises.

4.1. PIM Messages

PIMv2 protocol messages are encapsulated in IP datagrams with protocol number 103. PIM messages can be sent as unicast or multicast packets. When sent as multicast packets, PIM uses the multicast IP address 224.0.0.13, which is reserved as the *ALL-PIM-Routers* group.

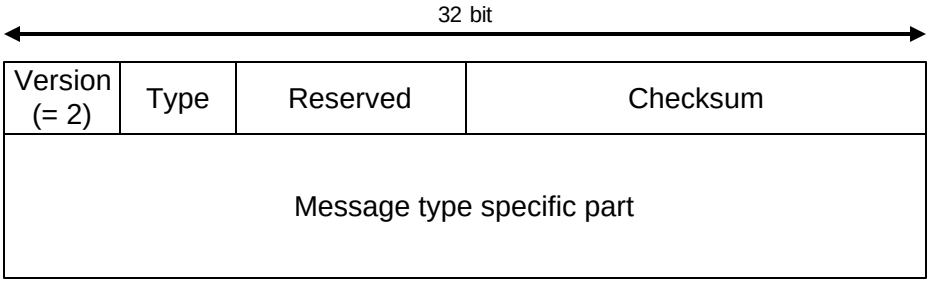


Figure 10.22. PIMv2 Message format

PIM-DM messages	Type	PIM-DM	PIM-SM
Hello	0	✓	✓
Register	1		✓
Register-Stop	2		✓
Join/Prune	3	✓	✓
Bootstrap	4		✓
Assert	5	✓	✓
Graft	6	✓	

[9] RFC 3376

Graft-Ack	7	✓	
Candidate-RP-Advertisement	8		✓

Table 7.1. PIMv2 Message Types

The format of a PIMv2 message is shown in Figure 10.22. All messages have a four byte long header, which is followed by type-specific fields. The first field of the header is the version field, which is set to 2 in PIMv2. The Type field specifies one of the message types listed in Table 7.1. Message types *Hello, Join/Prune, Bootstrap, Assert*, and *Graft* are sent as multicast packets to address 224.0.0.13 with the TTL field in the IP header set to 1. Note that the same message type is used for *Join* and *Prune* messages. All other message types are sent as unicast messages. The third byte of the header is unused. The last two bytes of the header are a checksum. We refer to[10] for a description of the type-specific fields of PIM messages.

4.2. Designated Router and Designated Forwarder

Each PIM router periodically sends *Hello* messages on all PIM enabled interfaces to detect neighboring routers. Hello messages are also used to select a Designated Router (DR) on a network with multiple routers. For a given network, the role of the DR is to send *Join/Prune* and *Register* messages on behalf of the hosts that are on the network. When multiple routers send *Hello* messages, the sender with the largest IP address assumes the role of the DR. As a result, the DR can be different from the querier, which is the router that sends IGMP queries on a network.

The role of the PIM *Assert* messages in is to determine the forwarder on a network with multiple routers. Recall from Subsection 2.1 that the forwarder is the router that forwards multicast packet on network with hosts that are multicast group members. The forwarder is generally identical to the DR, but not necessarily.

A router sends an *Assert* message when it receives a multicast packet on an interface that is listed in the outgoing interface list of the matching routing entry. Receiving a message on an outgoing interface is an indication that packets are duplicated, that is, more than one router forwards the same multicast packets to a network. Such a situation is illustrated in Figure 10.23, where both routers R1 and R2 forward multicast packet for the same (S, G) entry on a network. Here, both routers will *Assert* messages on the Ethernet network.
An *Assert* message contains, in addition to a source address and group address, a unicast cost metric for sending packets to the source, and a preference metric for the a unicast cost. The preference metric, which corresponds to the administrative distance that we saw in unicast routing tables of IOS, can be used to express a preference between unicast routing protocols. The router with the smallest preference metric becomes the forwarder. If the preference metrics

[10] RFC 2362

are equal, the router that sent the lowest unicast cost metric becomes the forwarder. If the unicast metrics are also equal, the router with the highest IP address becomes the forwarder. After the transmission of *Assert* messages, only the forwarder continues to forward messages on the network.

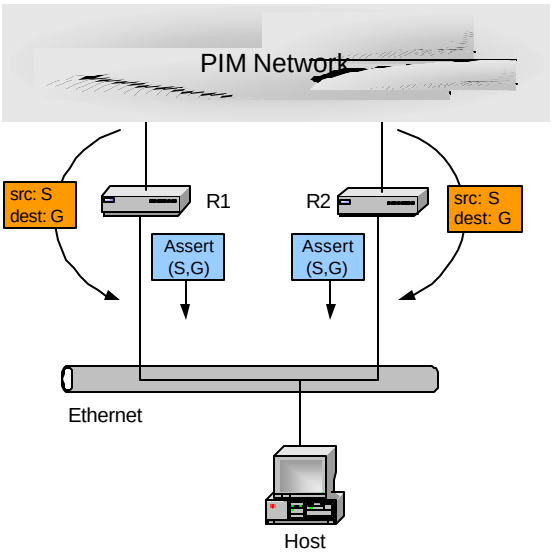


Figure 10.23. Election of a forwarder using Assert messages. Multicast packets are shown in orange, and PIM messages are shown in blue.

4.3. PIM Dense Mode (PIM-DM)

PIM-DM is a relatively straightforward implementation of the flood-and-prune protocol presented in Subsection 3.3. For an illustration, in Figure 10.24 host S1 is a multicast source for group G and host H2 is a group member. When source S1 starts sending IP datagrams to group G, the datagram is flooded to all routers. Each router that processes the packet, creates an (S1,G) multicast routing table entry, where the incoming interface is set to the RPF interface for source S1, and the outgoing interface list contains all other interfaces. For example, at router R2, the multicast routing entry is:

Source IP address	Multicast group	Incoming interface	Outgoing interface list
S1	G	I1	I2, I3

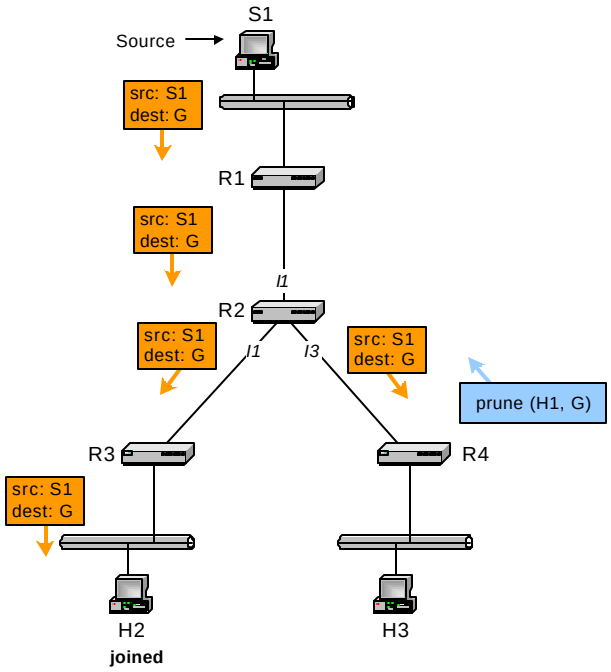


Figure 10.24. Reverse Path Forwarding and pruning in PIM-DM.

According to the flood-and-prune algorithm, router R4 which is not connected to a network with group members sends a *Prune* message on its RPF interface to router R2. The *Prune* message contains the multicast address G and the address of the pruned source S.[11] When R2 receives the prune message, it marks the corresponding outgoing interface as pruned:

Source IP address	Multicast group	Incoming interface	Outgoing interface list
H1	G	S1	I2, I3(pruned)

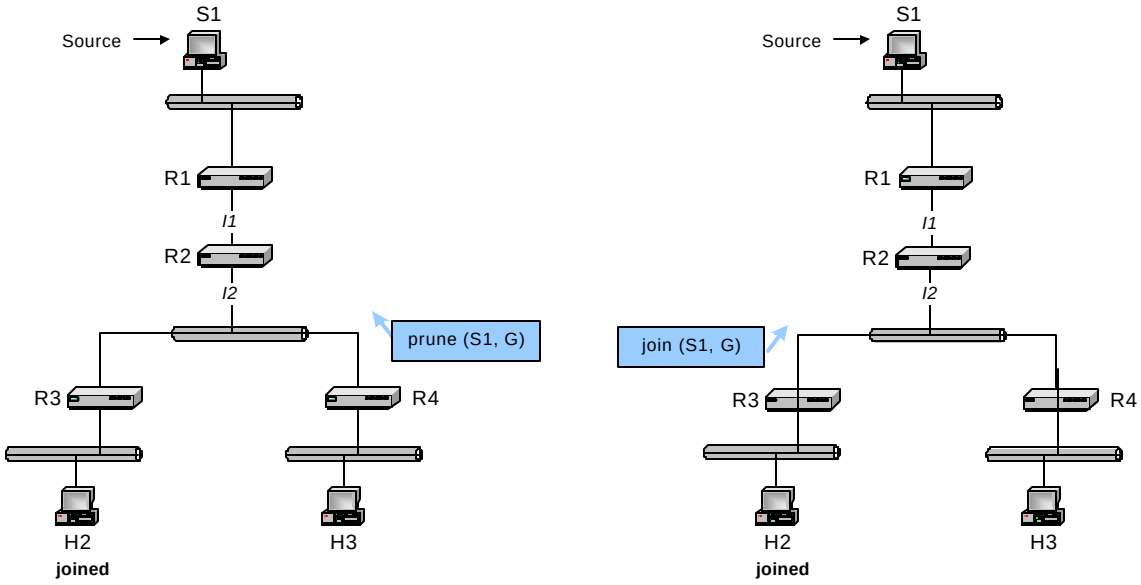
A router does not forward multicast packets on pruned interfaces. There is a timer associated with each pruned interface. When the time expires, the router reinstates interface I3, and resumes sending datagrams the interface.

When host H5 wants to join multicast group G, it sends an IGMP message to R4. R4, in turn, sends a *Graft* message for (S1,G) on the RPF interface for S1. R4 must keep track of all previously sent *Prune* message, otherwise, R4 does not know that it is necessary to send a *Graft* message. A *Graft* message has the effect of reinstating a pruned entry. The *Graft* message is

[11] We will write *Prune message* and *Join message*, even though PIM has only one message type *Prune/Join*. Each *Prune/Join* message has a list of pruned entries, and a list for joined entries. When we refer to a *Prune message*, we mean that the list of joined routing entries is empty.

forwarded on the reverse shortest path until a router is reached that has an (S1,G) entry. In PIM-DM, *Graft* messages are acknowledged with *Graft-ACK* messages.

Pruning is not straightforward when multiple routers are reachable on an outgoing interface of a multicast routing table. The problem is illustrated in Figure 10.25, where router R2 has two downstream routers, R3 and R4, that are reachable with the outgoing interface I2. Here, even when R4 sends a *Prune* message to R2, R2 should continue forwarding datagrams, since H4 is a multicast receiver. In PIM-DM, this issue is addressed in a procedure which is called *prune override*. When H5 sends a *Prune*(S1,G) message (Figure 10.25(a)), router R2 does not immediately disable the outgoing interface I2, but waits for a few seconds. Since PIM *Join/Prune* messages are sent to a multicast address, H4 sees the *Prune*(S1,G) message from H5, and sends a *Join*(S1,G) message to R2. The *Join* message overrides the prune message, and R2 continues forwarding datagrams to the Ethernet network.



(a) Prune Message from R4

(b) Join Message from R3

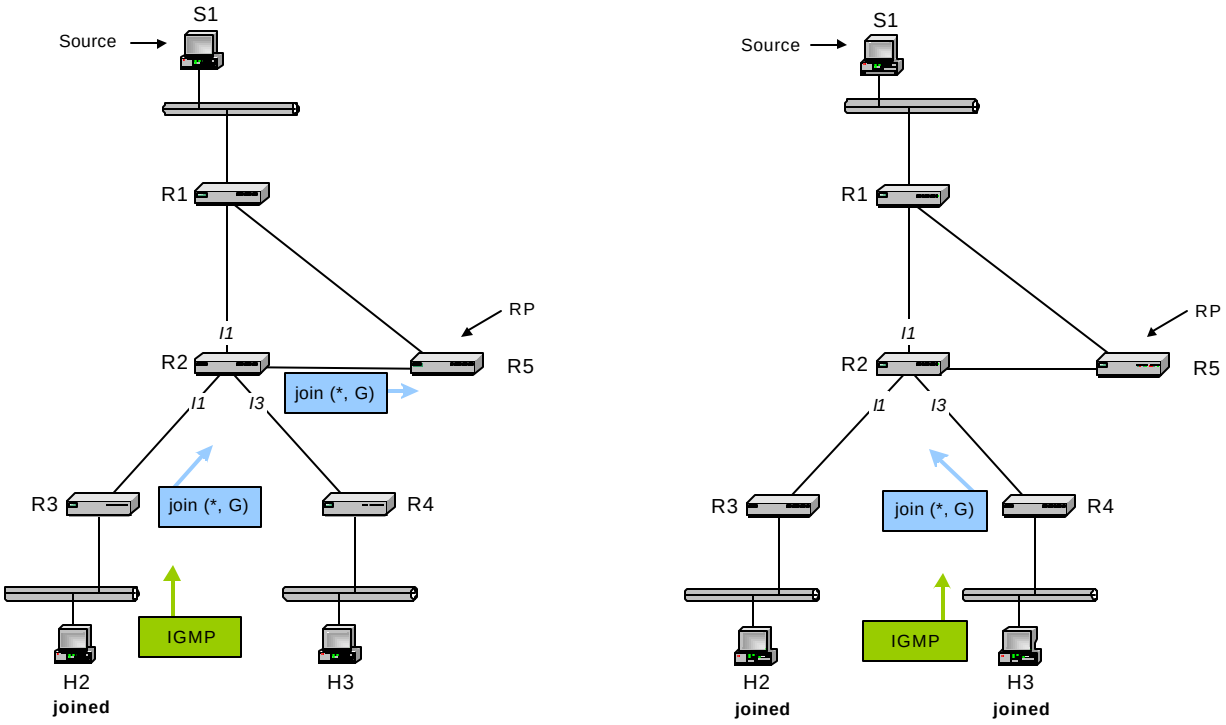
Figure 10.25. Prune Override in PIM-DM.

4.4. PIM Sparse Mode (PIM-SM)

PIM-SM is more complex than PIM-DM. PIM-SM initially builds a core-based tree routing protocol for a multicast group, which is shared by all sources. When the traffic from a source exceeds a threshold, PIM constructs a source-based tree for that source.

We next discuss how the core-based tree is constructed, how a source transmits data, and how PIM-SM switches to a source-based tree. In PIM-SM, a core router is called *rendezvous point (RP)* and a core-based tree is called *rendezvous-point tree (RPT)*. Henceforth, we will use the PIM-SM terminology. The RPT constructed in PIM-SM is similar as described in Section 3.4. The RPT in PIM-SM is unidirectional, that is, multicast packets are forwarded only downstream in the RPT and sources send multicast packet to the RP. The RPs can either be statically configured or it can be dynamically determined via an advertisement mechanism, which is part of PIM-SM. Also, a network can have a different RP for each multicast group. To keep the discussion of PIM-SM concise, we assume that there is only one statically configured RP in the network.

In PIM a router can have both an (S, G) and an (*, G) entry for the same multicast group. In this case, when a packet arrives with source address S for group G the router first tries to find a match in the routing table for an entry (S, G). If no match is found, the router looks for an (*, G) entry. If no matching entry is found, the router tries to forward the packet to the RP for that group. If no RP is known, the datagram is discarded. When a match is found, the router performs an RP check, that is, it verifies that the datagram was received on the incoming interface that is listed in the routing table entry, and then forwards the data to the outgoing interface list.



(a) PIM-SM: H2 joins

(b PIM-DM: H3 joins

Figure 10.26. PIM-SM. Building the core-based tree.

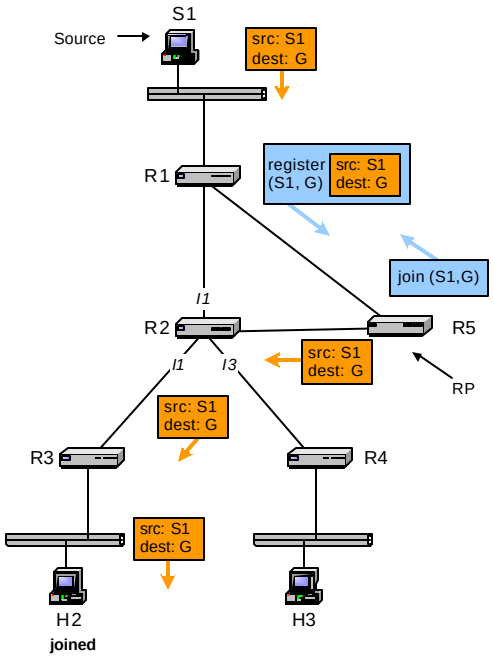
When a host joins a multicast group, the DR for that host sends a *Join* message on the reverse path to the RP. In Figure 10.26(a), the DR for host H2, router R3, sends a *Join*(*, G) message to R2, its RPF neighbor for the RP. Assuming that R2 does not have a multicast entry for (*, G), when R2 receives the *Join* message, it sets up a (*, G) routing table entry, and sends a *Join*(*, G) message to R5. When the RP receives the *Join* message, it adds the outgoing interface that connects to R2 to the routing table entry for (*, G). As with *Prune* messages, *Join* messages are sent periodically, about once per minute. The routing entries created with a *Join* message have a limited lifetime, and a router deletes an entry if no new *Join* message has been received for some time.

When host H3 joins the multicast group G, as shown in Figure 10.26(b), router R4 sends a *Join*(*, G) message to R2. Since R2 already has an entry for (*, G), the R2 does not forward the *Join* message to the RP, and simply adds interface I2 to the outgoing interface list for the (*, G) routing table entry.

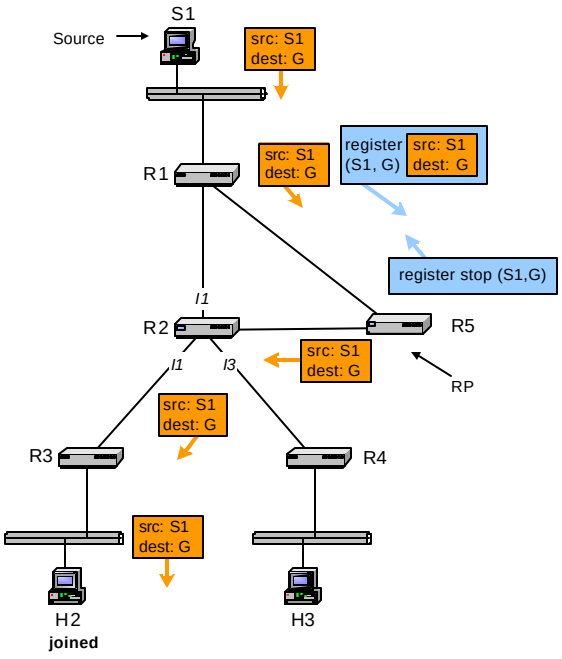
Next we explore data delivery in an RPT. When a source wants to transmit data to a multicast group, the DR for the source sends a *Register* message to the RP of the group, and encapsulates

the multicast packet in the *Register* message. The RP that receives a *Register* message extracts the multicast packets, and sends the multicast packet downstream in the RPT. If the RPT does not have any downstream nodes, the RP discards the multicast packet. When the *Register* message with the first datagram message arrives at the RP, the RP sends a *Join*(S,G) message to the source. The *Join* message travels to the RP on the reverse shortest path and sets up a (S, G) routing table entries on all routers between the source and the RP. When the *Join* message reaches the DR of the source, the DR sends multicast packets from the source to the RP. Once the RP receives the multicast packets from the reverse path, it sends a *Register-stop* message to the DR to stop the transmission of encapsulated datagrams. For a certain time, between the reception of the *Join* message and the *Register-stop* message, the DR of the source receives two copies of each multicast packet from the source, one encapsulated in a *Register* message and one native multicast packet.

Refer to Figure 10.27. for an illustration. In Figure 10.27(a), host S1 sends a multicast packet to its DR, R1. R1 encapsulated the datagram in a Register message, and sends the message as a unicast message to the RP. The RP extracts the multicast packet from the register message, and forwards the message in the RPT. In the figure, we assume that only host H1 has joined the multicast group G, and the RPT consists of the path R5→ R2→ R3. Then, the RP sends a *Join*(S1,G) to the DR of the source. When R1 receives the *Join* message it sets up an (S1, G) routing table entry, and begins forwarding native multicast packets, that is, regular IP datagrams with the destination address set to G, to the RP. As shown in Figure 10.27(b), R1 continues sending multicast packets encapsulated in *Register* messages to the RP. As soon, as the first native multicast packet reaches the R5, R5 sends a *Register-stop* message to R1. This stops the transmission of *Register* messages with encapsulated multicast packets.

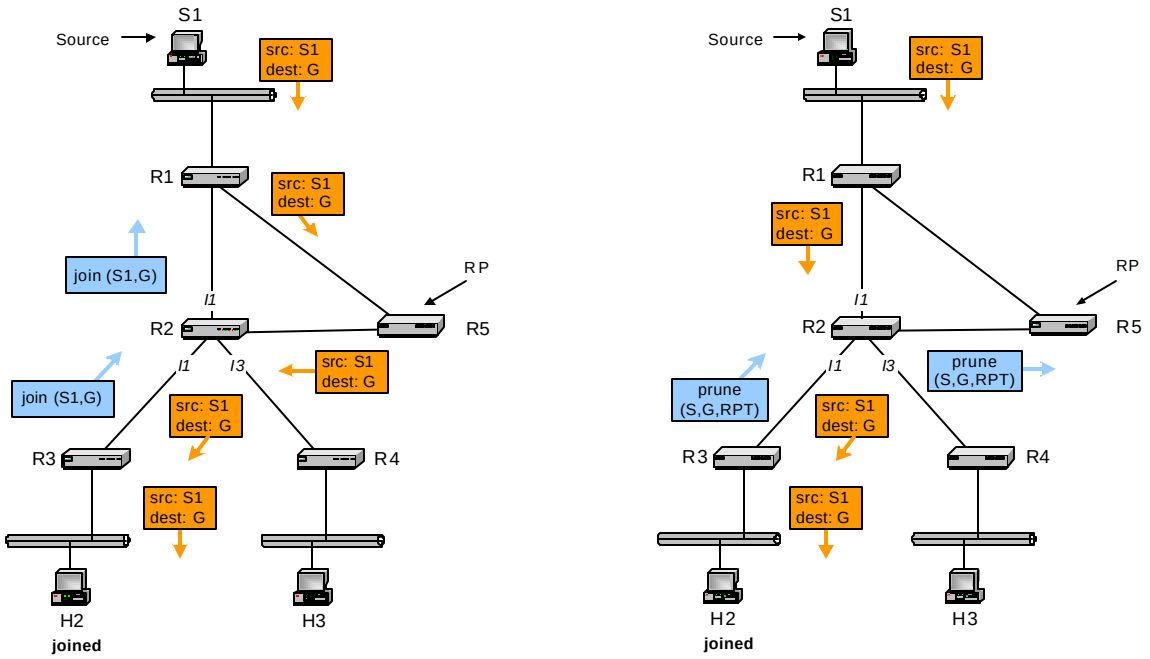


(a) PIM-SM: Register message to RP



(b) PIM-SM: Register Stop message to S1

Figure 10.27. Data transmission in PIM-SM.



(a) PIM-SM: R3 switches to a SPT

(b) PIM-SM: Data follows a SPT

Figure 10.28. Switching to a reverse shortest-path tree in PIM -SM.

PIM-SM distribution trees always start out as an RPT, but when the data rate from a source exceeds a certain threshold, the distribution tree is switched over to a source-based tree. In the context of PIM-SM, the source-based tree is called *shortest path tree* (SPT). The switching from an RPT tree to a source-based tree is initiated by the DRs connected to multicast group members by sending a *Join* message on the RPF interface to the source. When the SPT is established, a router may receive two copies of the multicast packet, one from the RPT and one from the SPT. To stop the duplicate transmissions, the router sends a *Prune* message to the RP, as soon as it receives a packet from the SPT. The *Prune* message removes the router from the RPT.

Switching to a source-based tree occurs even if the RP is on the reverse path to a source. By default, in Cisco IOS, a DR switches to a SPT tree as soon as the first multicast packet arrives from a source.

We illustrate the switching from the RPT to an SPT in Figure 10.28. The DR of host H2, R3, initiates a switch to a SPT for source S1, by creating an (S1, G) routing table entry and by sending a *Join*(S1, G) message on its RPF interface for S1 (see Figure 10.28(a)). When R2 receives the *Join*(S1, G), it creates an (S1,G) routing table entry and sends a *Join*(S1, G) to R1.

As soon as the *Join* message arrives at R1, R1 adds the interface to R2 to is outgoing interface list and starts forwarding multicast packets to R2, as well as to the RP, R5.

When a router receives a packet from the SPT, it sends a Prune(S1,G,RP) message to the RP (see Figure 10.28(b)). This is a special prune message which is not sent to the source S1, but to the RP. The Prune message stops the forwarding of data from source S1 for group G on the RPT at R2, and then at R5. After the prune message is processed at the RP, H1 receives multicast packets from S1 only from the SPT.

5. Interdomain multicast routing

All multicast routing protocols discussed so far, particularly, PIM-DM and PIM-SM, are intradomain routing protocols that operate in a single autonomous system. Additional protocol solutions are needed to enable multicast across multiple domains. In unicast, interdomain routing issues are solved in a hierarchical fashion. An intradomain routing protocol, for example, OSPF, determines routes within a domain, and an interdomain routing protocol, for example, BGP, determines routes between domains. A similar hierarchical approach to interdomain routing is being developed for multicasting, in the form of the *Border Gateway Multicast Protocol (BGMP),* which can construct a distribution tree between domains. However, BGMP is not yet deployed in the Internet.

The current solution to interdomain multicast routing involves three protocols. One of them is PIM-SM, which we discussed earlier. The other two are the *Multicast Source Discovery Protocol (MSDP)* and the *Multiprotocol Extensions for BGP (MBGP)*[12]. We briefly discuss how this interdomain solution works.

In PIM-SM, a domain is composed of a set of routers that use the same RPs for multicasting. Generally, the scope of a PIM-SM domain is an autonomous system. In each domain, PIM-SM intradomain multicasting routing builds a multicast distribution tree. In Figure 10.29, we illustrate two PIM-SM domains, AS1 and AS2. The rendezvous points in the domains are R2 in AS1 and R3 in AS2.

The protocol MSDP is used by an RP to inform the RPs in other domains about currently active sources. Specifically, when an RP receives a PIM *Register* message from a new source, it sends an *MSDP Source Active* message to a remote RP. The *MSDP Source Active* message contains an encapsulated multicast packet.

---

[12] RFC 2858

This is illustrated in Figure 10.29. Suppose that H1 becomes a new multicast source for group G. Then, the DR of H1, R1, sends a *PIM Register* message to R2, the RP in AS 1. The RP sends a *MSDP Source Active* message to R6, the RP in AS 2. MSDP messages are sent over TCP connections that are established between RPs. Once the MSDP message is received by the remote RP, the multicast packet is extracted and transmitted downstream the RPT. When the DR for a multicast group member, for example, router R7, receives a multicast packet from H1, it sends a *PIM Join*(H1,G) message on the RPF interface for H1. This initiates setting up a source-based tree from H1 to H2. As shown in Figure 10.29, the *Join* message traverses the path R7→R5→R4→R1. In this way, a source-based tree is setup from R1 to R7.

The role of MBGP is to provide control over the routing paths between domains. Suppose that, in the network in Figure 10.29, for some administrative reason, multicast traffic between AS1 and AS2 should traverse the link between R3 and R5, instead of the link between R4 and R5. Since *PIM Join* messages from receivers always follow the reverse shortest path, there is a need for a protocol which advertises a reverse path that is different from the unicast routing path selected by PIM-SM. This is accomplished by MBGP. MBGP is an extension to BGP, which permits a router to advertise multicast routes that are different from unicast routes. Using MBGP, router R3 can advertise itself as the as the next hop for the reverse path from AS 2 to AS 1.

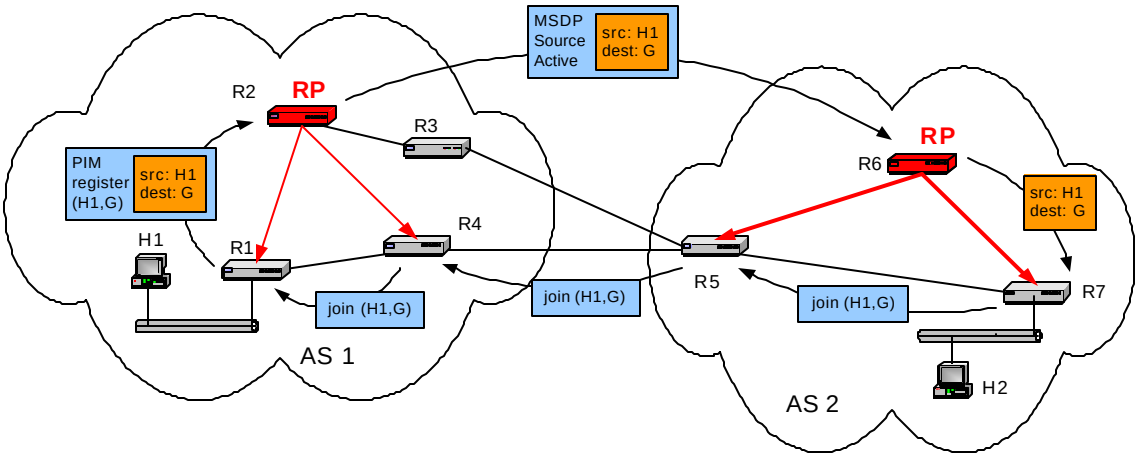


Figure 10.29. Interdomain multicast routing.

6. Multicast Routing in IOS

Cisco IOS supports the intradomain multicast protocols PIM-DM and PIM-SM. The protocols DVMRP, CBT, and MOSPF are not available in IOS, but IOS can interoperate with DVMRP. In this section we discuss a basic set of commands for configuring IP multicast in IOS, which are used in the lab exercises in Lab 10.

6.1. Basic configuration for IP multicast

The IOS commands to enable multicast routing in IOS are similar to the corresponding commands for unicast routing. In IOS, IP multicast forwarding is enabled with the global configuration command

```
Router1(config) ip multicast-routing
```

and disabled with the command

```
Router1(config) no ip multicast-routing
```

Multicast routing protocols must be enabled separately for each interface, and a router can run different routing protocols on its interfaces. IOS fully supports PIM-DM and PIM-SM. Since PIM relies on an underlying unicast routing protocol, a protocol, such as RIP or OSPF must be running on each interface that has PIM enabled. The commands to start PIM-DM on an interface, say, interface Ethernet0/0, are

```
Router1(config) interface Ethernet0/0
Router1(config-if) ip pim dense-mode
```

The corresponding commands to enable PIM-SM are

```
Router1(config) interface Ethernet0/0
Router1(config-if) ip pim sparse-mode
```

The commands to disable multicast routing is done with the commands *no ip pim dense-mode* or *ip pim sparse-mode.*

A router that performs multicast forwarding responds to IGMPv2 messages on all connected local area networks. Even if multicast forwarding is disabled, the router responds to IGMPv2 messages on all interfaces where a multicast routing protocol is running. The following sequence of commands, which will be used in Lab 10, enables IGMPv2 on interface Ethernet0/0 at a router that has multicast forwarding disabled:

```
Router1(config) ip routing
Router1(config) no ip multicast-routing
Router1(config) interface Ethernet0/0
```

```
Router1(config-if) ip pim dense-mode
Router1(config-if) end
```

The first command enables IP forwarding and the second command disables IP multicast forwarding at the router. Then, the interface configuration mode is entered for interface *Ethernet0/0*, and PIM-DM is started. In this configuration, the router sends and receives IGMPv2 messages on interface *Ethernet0/0*, but does not forward multicast packets.

The following sequence of commands is a typical configuration of multicast routing on an interface. Since PIM assumes that a unicast routing protocol is running, the command sequence includes the configuration of a unicast protocol, RIPv2, in this case.

```
(1)  Router1> enable
(2)  Password: <enable secret>
(3)  Router1# configure terminal
(4)  Router1(config)# no ip routing
(5)  Router1(config)# ip routing
(6)  Router1(config)# router rip
(7)  Router1(config-router)# version 2
(8)  Router1(config-router)# network 10.0.0.0
(9)  Router1(config-router)# interface ethernet0
(10) Router1(config)#ip multicast-routing
(11) Router1(config)# interface Ethernet0/0
(12) Router1(config-if)# no shutdown
(13) Router1(config-if)# ip address 10.0.2.1 255.255.255.0
(14) Router1(config-if)# ip pim dense-mode
(15) Router1(config-if)# end
```

Commands (1)–(3) put the router into the global configuration mode. Commands (4)–(5) enable IP forwarding. As discussed in Chapter 3, issuing the command *no ip routing* resets the current IP configuration. Commands (6)–(8) configure RIPv2 as unicast routing protocol. Command (10) enables IP multicast forwarding. The remaining commands configure the interface *Ethernet0/0*. Command (12) enables the interface, (13) sets the IP address, and (14) starts PIM-DM. PIM-SM is started with the command *ip pim sparse-mode*. For PIM-SM, additional commands are needed to specify the RP.

The following list summarizes the most important commands for a multicast IP configuration in IOS.

IOS Mode: Privileged Exec mode

clear ip mroute *

Delete all multicast routing table entries.

IOS Mode: Global configuration

ip multicast-routing
no ip multicast-routing

The first command enables IP multicast forwarding and the second command disables multicast forwarding.

IOS Mode: Interface configuration

ip pim dense-mode
ip pim sparse-mode
ip pim sparse-dense-mode

Enables the multicast routing protocol PIM-DM or PIM-SM for an interface. The option sparse-dense-mode uses PIM-SM if an RP is known, and PIM-DM when an RP is not known.

6.2. Displaying multicast configuration

IOS has numerous commands to view the current IP multicast configuration of a router. The following list shows some of the most important commands.

IOS mode: Privileged EXEC

show ip igmp groups
show ip igmp interface *interface*

Displays the multicast groups that have group members on the interfaces of the router. The first command displays information on all multicast groups that have members. The second command displays multicast groups that have group members on the network attached to the given interface.

show ip mroute
show ip mroute *groupaddress*

Displays the multicast routing table for all groups. When a multicast IP address is given as an argument, only the entries for that group are displayed.

show ip rpf *IPaddress*

Displays information on the neighbor router that is reached through the RPF interface for the given IP address. This router is called an RPF neighbor.

As an example, here is the output of the command *show ip igmp groups* at a router.

```
Router1#show ip igmp groups
IGMP Connected Group Membership
Group Address    Interface            Uptime    Expires   Last Reporter
224.2.2.2        Ethernet0/1          00:16:47  00:02:46  10.0.3.20
224.1.1.1        Ethernet0/0          00:17:03  00:02:35  10.0.2.21
```

The table lists the multicast IP address, the interface on which IGMP membership reports have been received, and the IP address of the host that transmitted the most recent IGMP group membership report. Also displayed are the age of the IGMP entry (*Uptime*) and the maximum time that the router will keep the entry if no further IGMP membership reports are received (*Expires*).

The command to display the multicast routing table is *show ip mroute*. Below we show the routing table entries for multicast group 224.1.1.1 at a multicast router that runs PIM-SM.

```
router1#show ip mroute
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, C - Connected, L - Local, P - Pruned
       R - RP-bit set, F - Register flag, T - SPT-bit set, J - Join SPT
       X - Proxy Join Timer Running
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.1.1), 01:02:42/00:02:59, RP 10.0.3.2, flags: SJCF
  Incoming interface: Ethernet0, RPF nbr 10.0.2.2
  Outgoing interface list:
    Ethernet0/1, Forward/Sparse, 01:00:35/00:02:07

(10.0.1.11, 224.1.1.1), 01:00:51/00:03:28, flags: CFT
  Incoming interface: Ethernet1, nbr 10.0.5.4
  Outgoing interface list:
    Serial1/0, Forward/Sparse, 00:00:41/00:02:48
    Ethernet0/0, Forward/Sparse, 01:00:51/00:02:43
```

The first lines of the output explain the format of the routing table. The flags specify the state of the routing table entry or how the entry was created. The present routing table has three entries for the group 224.1.1.1, one (*, G) entry and one (S, G) entries. The (*, G) entry is a routing entry for an RPT. The incoming interface is Ethernet0/0 with 10.0.2.2 as RPF

neighbor in the shared tree, and the outgoing interface is Ethernet0/1. The rendezvous point (RP) for this group is given as 10.0.3.2. The S flag indicates that this is a sparse mode entry, the J flag tells that the router will join a shortest-path tree, the C flag indicates that a multicast group member is reachable via the outgoing interface, and the F flag tells that the router may need to send a *PIM Register*. The outgoing interface lists has only one interface. The (S, G) entry is for an SPT for source 10.0.1.11. The T flag indicates that the router is part of the SPT. This routing entry has two outgoing interfaces.

Since multicast forwarding is generally done using reverse path forwarding, the following command can be used to display the results of the RPF check for a given source address.

```
Router1#show ip rpf 10.0.1.11
RPF information for ? (10.0.1.11)
  RPF interface: Ethernet0/0
  RPF neighbor: ? (10.0.5.1)
  RPF route/mask: 10.0.1.0/255.255.255.0
  RPF type: unicast
```

Some of the information on the RPF neighbors can also be obtained from the multicast routing table.

6.3. Configuring PIM in IOS

There are numerous IOS commands available for configuring PIM. Lab 10 only uses two of these commands. The first command statically sets the RP. The command is

```
Router1(config)# ip pim RPaddress
```

where *RPaddress* is the IP address of the RP. All routers in the network need to issue this command, and they must select the same router as RP.

The other configuration command is used in the configuration of PIM-SM. As discussed earlier, PIM-SM switches from an RPT to a SPT, when the transmission rate from a source exceeds a threshold. By default, Cisco IOS switches from to a SPT when the first multicast packet arrives at the designated router of a multicast group member. The default can be changed by setting the parameter *spt-threshold*. When issuing the command

```
Router1(config)# ip pim spt-threshold datarate
```

where *datarate* is the threshold rate, measured in kilobits per second, at which a PIM-SM router switches from a RPT to a SPT. By typing

```
Router1(config)# ip pim spt-threshold infinity
```

PIM-SM is configured so that it never switches to a source-based tree. The following summarizes the commands.

IOS Mode: Global configuration

ip pim *rp-address*
Sets the RP of PIM-SM to IP address *rp-address*.

ip pim spt-threshold *datarate*
ip pim spt-threshold infinity
Sets the threshold data rate to *datarate*, measured in kilobit per second, at which a PIM-SM router switches from a RPT to a SPT. The second command enforces that PIM-SM never switches to a SPT.

There are several IOS commands to query the status of a PIM router, and to display debugging information for PIM-SM. The following list summarizes some of the commands.

IOS mode: Privileged EXEC

show ip pim interface
Displays information about PIM enabled interfaces.
show ip pim neighbor
Lists the neighbors that a router has discovered via PIM *Hello* messages.
show ip pim rp mapping
Lists the mapping of multicast groups to RPs at a router.

Here is the output of the command *show ip pim interface* at a router:

```
Router1#show ip pim interface
Address          Interface          Version/Mode    Nbr    Query     DR
                                                    Count  Intvl
10.0.2.1         Ethernet0/0        v2/Sparse        1     30      10.0.2.2
10.0.1.1         Ethernet0/1        v2/Sparse        0     30      10.0.1.1
10.0.3.1         Serial1/0          v2/Sparse        1     30      0.0.0.0
```

The command lists the PIM enabled interfaces, with the version of the PIM protocol that is running, the number of neighbors that have been detected via PIM *Hello* messages, the configured value of the interval for the *Hello* messages, and the IP address of the designated

router. The output shows that there are three PIM-enabled interfaces, and that the router has discovered one neighbor on two of the interfaces.

To display the IP addresses of the neighbors, one can use the command *show ip pim neighbor:*

```
Router1#show ip pim neighbor
PIM Neighbor Table
Neighbor Address  Interface          Uptime    Expires   Ver  Mode
10.0.2.2          Ethernet0/0        03:31:50  00:01:44  v2   (DR)
10.0.5.2          Serial1/0          00:43:29  00:01:18  v2
```

The command *show ip pim rp mapping* displays the RPs that are known at this router. For example, the command

```
Router1#show ip pim rp mapping
PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static
     RP: 10.0.3.2 (?)
```

shows that 10.0.3.2 has been statically configured as the RP for the prefix 224.0.0.0/4, which includes all multicast addresses.

7. Tools and Utilities

The software tools for Lab 10 include Linux commands to send and receive multicast packets on a Linux PC (*msend* and *mreceive*), multicast options for known Linux command (*ping*, *netstat*), the commands *mrinfo* and *mtrace*, available on both Linux and IOS, which query multicast routers about their multicast configuration, and the Linux command *map-mbone*, which displays how multicast routers are connected to each other.

7.1. msend and mreceive

The Linux commands *msend* and *mreceive* send and receive, respectively, multicast packets for a given multicast group on Linux. These commands are the only software tools that have been developed specifically for the Internet Lab. The command *msend* periodically transmits UDP datagrams to a given multicast group. An *msend* program does not join the multicast group, unless the option *–join* is given as an argument. The command *mreceive* joins a multicast group and displays the payload of received multicast messages to the standard output. The commands *msend* and *mreceive* are minimal applications, which trivialize the task to start multicast senders and receivers on a network. The command

```
PC1% msend –g 224.1.1.1 –p 4444 –text "PC1"
```

sends UDP datagrams which contain the string "PC1" to multicast group 224.1.1.1 at port number 4444, and displays the payload of UDP datagrams that are transmitted by other senders as a text string. By default, *msend* transmits one datagram per second and the TTL value of the datagram is set to 1. A multicast receiver is started with the command

```
PC2% mreceive –g 224.1.1.1 –p 4444
```

This command displays all data sent to multicast address 224.1.1.1 at port number 4444 to the standard output. The *msend* and *mreceive* are independent, in the sense that a multicast sender can be started without a multicast receiver, and vice versa. The number of multicast senders and multicast receivers for the same IP address and port number is not limited.

The default values of *msend* and *mreceive* can be modified using various options. The complete set of options is provided in the appendix of this chapter. For example, the commands

```
PC1% msend –join –g 224.1.1.1 –p 4444 –text "PC1" –ttl 10 –i eth0
PC2% mreceive –g 224.1.1.1 –p 4444 –i eth1
```

starts a multicast sender for group 224.1.1.1, and which sends multicast packets to a given interface (*eth0*), with a fixed TTL value (10), and a multicast receiver that listens to multicast packets for group 224.1.1.1 on interface *eth1*.

7.2. Multicast options of Linux commands: ping and netstat –g

The Linux commands *ping* and *netstat* can be useful when working with IP multicast. *Ping* commands can be used for multicast addresses by providing a multicast address as argument, as follows:

```
PC1%ping 224.1.1.1
```

All hosts that have joined a multicast group, respond to the *ping*. By default the TTL field of a multicast ping is set to one. This can be modified with the *–t* option. For example, the following command sets the TTL field in the multicast ping to 5:

```
PC1%ping –t 5 224.1.1.1
```

Running *netstat* with the *–g* option on a Linux system displays the set of all multicast groups that the Linux system has joined. The output of the command is as follows.

```
PC1%netstat -g
IPv6/IPv4 Group Memberships
Interface       RefCnt Group
--------------- ------ ---------------------
lo              1      224.0.0.1
eth0            1      224.1.1.1
```

```
eth0            1       224.0.0.1
```

Here, the Linux PC has joined group 224.0.0.1 on the loopback interface, and has joined groups 224.0.0.1 and 224.1.1.1 on interface *eth0*.

7.3. mrinfo

The command *mrinfo* is a query utility that retrieve multicast configuration information from routers. *Mrinfo* is available both on Linux and on IOS.

IOS (User EXEC mode) and Linux

mrinfo
Displays information on the local multicast configuration of a router.

mrinfo *Ipaddress*
Sends a query to the specified IP address, which should be the address of a multicast router, and retrieves data on the multicast configuration.

Running *mrinfo* on a Cisco router without an argument yields information on the local system:

```
Router1#mrinfo
  10.0.4.3 [version  12.0] [flags: PMA]:
  10.0.4.3 -> 10.0.4.4 [1/0/pim]
  10.0.3.3 -> 10.0.3.2 [1/0/pim/querier]
```

The first line lists the IP address of the local router, the IOS version number, and a set of flags. The following lines list the multicast enabled interfaces and the IP address of the neighboring router that the interface is connected to. At the end of the lines is additional information about the neighbor. In *[1/0/pim]*, the first number is the metric of the link, the second number indicates that the router does not enforce a TTL threshold, and *pim* indicates the routing protocol. The output *[1/0/pim/querier]* in the last line additionally specifies that the neighbor is the querier on the connected network.

When *mrinfo* is given the IP address of a router as an argument, *mrinfo* queries the state of the remote router. The following command queries router 10.0.4.4 from a Linux PC:

```
PC1% mrinfo 10.0.4.4
10.0.4.4 (10.0.4.4) [version 12.0]:
  10.0.4.4 -> 10.0.4.3 (10.0.4.3) [1/0/pim/querier]
  10.0.5.4 -> 10.0.5.1 (10.0.5.1) [1/0/pim]
```

Here, a query is sent from PC1 to IP address 10.0.4.4, and a response is sent back to PC1. Both query and the response is sent in a DVMRP message, which in turn is encapsulated in an IGMP

message. The query is sent as a *DVMRP Ask-Neighbors* message and the response is sent as a *DVMRP Neighbors-Reply* message. Recall that DVMRP is a multicast routing protocol. Routers and hosts that implement *mrinfo* are capable of parsing these DVMRP messages even if they do not run DVMRP as multicast routing protocol.

7.4. mtrace

The *mtrace* command, the multicast analogue to *traceroute*, traces a multicast route from a multicast receiver to a specified source address. *Mtrace* utility is available in Linux and in IOS.

IOS (User EXEC mode) or Linux

mtrace *DstIPaddress SrcIPaddress GroupIPaddress*
Displays information on the route of the reverse path from IP address *DstIPaddress* system to IP address *SrcIPaddress* for group address *GroupIPaddress.*

mtrace *DstIPaddress SrcIPaddress*
The route is traced for the default group address 224.2.0.1.

mtrace *SrcIPaddress*
The route is traced using address 224.2.0.1 as default group and the IP address of the local system as destination address.

The *mtrace* program sends a query to the router that is the forwarder for the given destination address. The router generates a report, which includes the IP addresses, packet counters, and routing error conditions, and passes the report to its RPF neighbor with respect to the source address. This router adds its information to the report packet and forwards the report to its own RPF neighbor. In this fashion, all routers on the reverse shortest path to the source address are visited. When the router that is directly connected to the source is reached, the report is sent to the querying IP address.

The following is the output of issuing *mtrace* in IOS from a router with IP address 10.0.4.3.

```
Router3#mtrace 10.0.1.11
Type escape sequence to abort.
Mtrace from 10.0.1.11 to 10.0.4.3 via RPF
From source (?) to destination (?)
Querying full reverse path...
 0  10.0.4.3
-1  10.0.4.3 PIM  thresh^ 0  4 ms   [10.0.1.0/24]
-2  10.0.4.4 PIM  thresh^ 0  -94692 ms   [10.0.1.0/24]
```

```
-3  10.0.5.1 PIM  thresh^ 0  24820 ms   [10.0.1.0/24]
-4  10.0.1.11
```

The first column lists the number of hops from IP address 10.0.4.3 to the address 10.0.1.11. The next column is the IP address of the current hop. The additional information is the routing protocol used, the TTL threshold at the router for forwarding multicast traffic. The time value gives the time until the report trace is forwarded. From the output, it is questionable that the time values are reliable. The negative value of -94692 ms appears to be due to a rounding error, and the value of 24820 ms appears excessively high.

The output of *mtrace* in Linux is similar. Here is the output of *mtrace* from IP address 10.0.3.31 to source address 10.0.4.41 for group address 224.1.1.1.

```
PC1% mtrace 10.0.4.41 224.1.1.1
Mtrace from 10.0.4.41 to 10.0.3.31 via group 224.1.1.1
Querying full reverse path...
  0  ? (10.0.3.31)
 -1  ? (10.0.3.3)  PIM  thresh^ 0
 -2  ? (10.0.4.41)
Round trip time 4 ms; total ttl of 1 required.
```

We note that the output of *mtrace* in Linux has a second section, which provides a pictorial view of the multicast forwarding path with additional information, including loss rate and forwarding delay. For a description of this second section, we refer to the manual page of *mrinfo.* The query and the reports of *mtrace* are transmitted in special IGMP message types that are designated for this purpose.

7.5. Map-mbone

The Linux command *map-mbone* can recursively display all multicast routers that are reachable from a local system. The *map-mbone* utility contacts neighboring multicast routers by sending *DVMRP Ask-Neighbors* message to multicast routers. If this multicast router responds, with a *DVMRP Neighbors-reply* message, the version number and a list of neighboring multicast router addresses is included in the response. When the requests results in the discovery of new multicast routers, *map-mbone* sends DVMRP Ask-Neighbors to these newly detected routers. The search for new routers continues until no new multicast routers are reported in a response.

The output of the command is shown below. The *–n* option suppresses a DNS lookup and the *–f* option sets a flooding option, to start a recursive search of neighboring routers.

```
PC1% map-mbone -f -n 10.0.1.1

Multicast Router Connectivity:

10.0.1.1: <v12.0>
    10.0.5.1:  10.0.5.4 [1/0]
    10.0.1.1:  10.0.1.1 [1/0/querier]
```

```
    10.0.2.1:  10.0.2.2 [1/0]

10.0.2.1: alias for 10.0.1.1

10.0.5.1: alias for 10.0.1.1

10.0.2.2: <v12.0>
    10.0.3.2:  10.0.3.3 [1/0]
    10.0.2.2:  10.0.2.1 [1/0/querier]

10.0.3.2: alias for 10.0.2.2

10.0.3.3: <v12.0>
    10.0.3.3:  10.0.3.2 [1/0/querier]
    10.0.4.3:  10.0.4.4 [1/0]

10.0.4.3: alias for 10.0.3.3

10.0.4.4: alias for 10.0.5.4

10.0.5.4: <v12.0>
    10.0.5.4:  10.0.5.1 [1/0]
    10.0.4.4:  10.0.4.3 [1/0/querier]
```

The command lists the IP address of each router and the IP address of the neighbor that is reachable via an interface. The output [1/0] or [1/0/querier] has the same interpretation as discussed in *mrinfo.*

.

RFCs:

[RFC 1075] Distance Vector Multicast Routing Protocol (DVMRP). D. Waitzman, C. Partridge, S. Deering, RFC 1075, November 1988.

[RFC 1112] Host Extensions for IP Multicasting, S. Deering, RFC 1112, August 1989. Obsoletes: RFC 988, RFC 1054.

[RFC 1584] Multicast Extensions to OSPF, J. Moy, RFC 1584, March 1994.

[RFC 2189] Core Based Trees (CBT version 2) Multicast Routing: Protocol Specification, A. Ballardie, RFC 2189, September 1997.

[RFC 2201] Core Based Trees (CBT) Multicast Routing Architecture, A. Ballardie, RFC 2201, September 1997.

[RFC 2236] Internet Group Management Protocol, Version 2, W. Fenner, RFC 2236, November 1997.
Obsoletes: RFC 1112.

[RFC 2365] Administratively Scoped IP Multicast, D. Meyer, RFC 2365, July 1998.

[draft-ietf-idmr-dvmrp-v3-10] Distance Vector Multicast Routing Protocol, T. Pusateri, Internet Draft, draft-ietf-idmr-dvmrp-v3-10, August 2000.
Obsoletes: RFC 1075

[RFC2362] Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification, D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, L. Wei, RFC2362, June 1998.
Obsoletes: 2117.

[RFC2858] Multiprotocol Extensions for BGP-4, T. Bates,Y. Rekhter, R. Chandra, D. Katz, RFC 2858, June 2000.
Obsoletes: RFC 2283.

[draft-ietf-msdp-spec-14] Multicast Source Discovery Protocol (MSDP), David Meyer, Bill Fenner (Editors), Internet Draft, <draft-ietf-msdp-spec-14.txt>, November 2002.

[RFC 3180] GLOP Addressing in 233/8, D. Meyer and P. Lothberg, RFC 3180, September 2001.
Obsoletes: RFC 2770.

[RFC 3376] Internet Group Management Protocol, Version 3, B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan, RFC 3376, October 2002.
Obsoletes: RFC 2236.

References

[Rodriguez01] Adolfo Rodriguez , John Gatrell, John Karas, Roland Peschke. TCP/IP Tutorial and Technical Overview, IBM Reedbook, Publisher: Prentice Hall, 2nd edition, October 2001.

[Almeroth99] Kevin Almeroth, The Evolution of Multicast: From the MBone to Inter-Domain Multicast to Internet2 Deployment, IEEE Network, January/February 2000, Pages 1-20.

[Doyle98] Jeff Doyle, Jennifer DeHave Carroll. Routing TCP/IP, Volume II, Cisco Press, 2001.

[Abritton99] John Abritton, Cisco IOS Essentials, McGraw-Hill 1999.

[Aweya00] James Aweya. On the design of IP routers Part 1: Router architectures, Journal of Systems Architecture 46 (2000) pp.483-511.

[Chappell99] Laura Chappel (Editor), Introduction to Cisco Router Configuration, McMillan Technical Publishing, 1999.

[Edwards02] Brian M. Edwards, Leonard A. Guiliano, Brian R. Wright, Interdomain Multicast Routing, Addison-Wesley, 2002.

[Kirch99] Olaf Kirch, Terry Dawson. Linux Network Administrator's Guide. 2nd edition. O'Reilly and Associates, 1999.

[Stevens94] W. Richard Stevens, TCP/IP Illustrated, Volume 1. The Protocols. Addison-Wesley Publishing Company, 1994.

[Wang2002] Jianping Wang, Yvan Pointurier, Jorg Liebeherr, Msend/Mreceive, Version 2.1, *(software)*, *http://www.cs.virginia.edu/~itlab/book/DNS-conf/mSendReceive.tar.gz*, 2002.

Appendix: Manual Page for msend and mreceive (version 2.1).

```
NAME
 msend - send UDP messages to a multicast group

SYNOPSIS
 msend -g group -p port [-t ttl] [-i ip] [-P period] [-text \"text\"|-n]
 msend –v


DESCRIPTION
  Continuously send UDP packets to a multicast group specified in the
   -g and -p options.

  -g     (Required) Specify the IP multicast address to which packets are sent.

  -p     (Required) Specify the UDP port number used by the multicast group.
```

```
  -t     Specify the TTL value in the message. The default value is 1.

  -i     Specify the IP address of the interface to be used to send the packets. The
         default value is INADDR_ANY which implies that the default interface
         selected by the system will be used.

  -join  Multicast sender will also join the multicast group (By default, a
         multicast member does not join).

  -P     Specify the interval between two transmitted packets. The default value is
         1000 (second).

  -text  Specify a string which is sent as the payload of the packets and is
         displayed by the mreceive command. The default value is empty string.

  -n      Interpret the contents of the message as a number (messages sent
          with send -n) instead of a string of characters. It should be
          specified while running msend with -n option.

  -v     Print version information.

SEE ALSO
           mreceive
```

```
NAME
 mreceive - receive UDP multicast messages and display them

SYNOPSIS
 mreceive -g group -p port [-i ip] ... [-i ip] [-n]
 mreceive -v

DESCRIPTION
 Join a multicast group specified by the -g and -p options, receive
 and display the multicast packets sent to this group by the msend
 command.

  -g     Specify the IP multicast address from which the packets are received.

  -p     Specify the UDP port number used by the multicast group.

  -i     Specify the IP addresses of one or more interfaces to receive multicast
         packets.

  -n     Interpret the contents of the message as a number (messages sent with send -
         n) instead of a string of characters.

  -v      Print version information.

SEE ALSO
 msend
```

0.50