CHAPTER 22

Network Layer:
Delivery, Forwarding, and Routing

This chapter describes the delivery, forwarding, and routing of IP packets to their final destinations. Delivery refers to the way a packet is handled by the underlying networks under the control of the network layer. Forwarding refers to the way a packet is delivered to the next station. Routing refers to the way routing tables are created to help in forwarding.

Routing protocols are used to continuously update the routing tables that are consulted for forwarding and routing. In this chapter, we also briefly discuss common unicast and multicast routing protocols.

22.1   DELIVERY

The network layer supervises the handling of the packets by the underlying physical networks.We define this handling as the delivery of a packet.

Direct Versus Indirect Delivery

The delivery of a packet to its final destination is accomplished by using two different methods of delivery, direct and indirect, as shown in Figure 22.1.
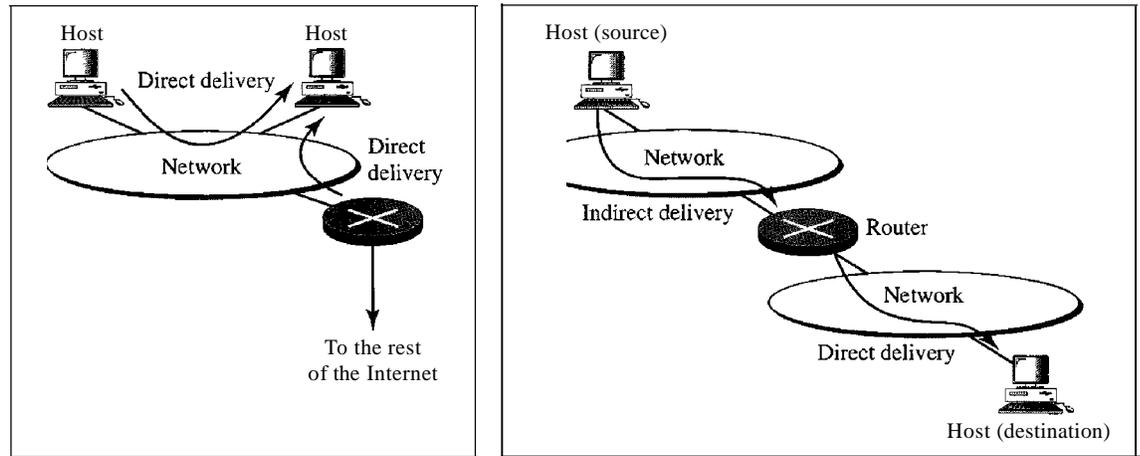
Direct Delivery

In a direct delivery, the final destination of the packet is a host connected to the same physical network as the deliverer. Direct delivery occurs when the source and destination of the packet are located on the same physical network or when the delivery is between the last router and the destination host.

The sender can easily determine if the delivery is direct. It can extract the network address of the destination (using the mask) and compare this address with the addresses of the networks to which it is connected. If a match is found, the delivery is direct.

Indirect Delivery

If the destination host is not on the same network as the deliverer, the packet is delivered indirectly. In an indirect delivery, the packet goes from router to router until it reaches the one connected to the same physical network as its final destination. Note

Figure 22.1   *Direct and indirect delivery*

a. Direct delivery

b. Indirect and direct delivery

that a delivery always involves one direct delivery but zero or more indirect deliveries. Note also that the last delivery is always a direct delivery.

22.2 FORWARDING

Forwarding means to place the packet in its route to its destination. Forwarding requires a host or a router to have a routing table. When a host has a packet to send or when a router has received a packet to be forwarded, it looks at this table to find the route to the final destination. However, this simple solution is impossible today in an internetwork such as the Internet because the number of entries needed in the routing table would make table lookups inefficient.

Forwarding Techniques

Several techniques can make the size of the routing table manageable and also handle issues such as security. We briefly discuss these methods here.
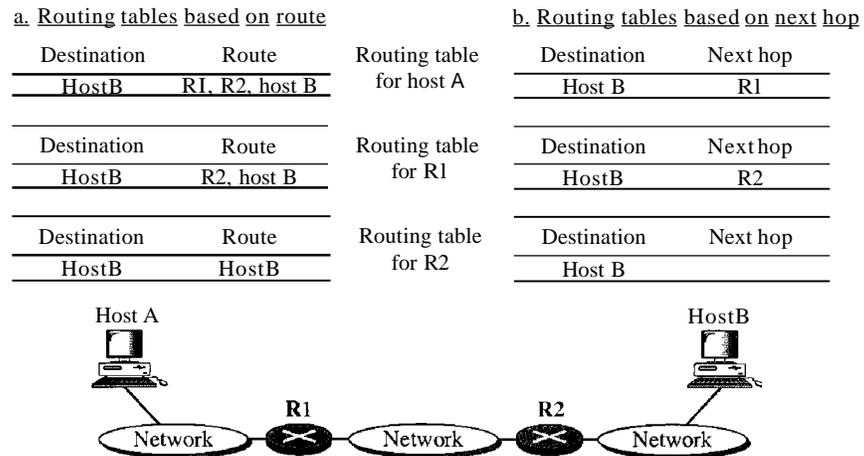
Next-Hop Method Versus Route Method

One technique to reduce the contents of a routing table is called the next-hop method. In this technique, the routing table holds only the address of the next hop instead of information about the complete route (route method). The entries of a routing table must be consistent with one another. Figure 22.2 shows how routing tables can be simplified by using this technique.

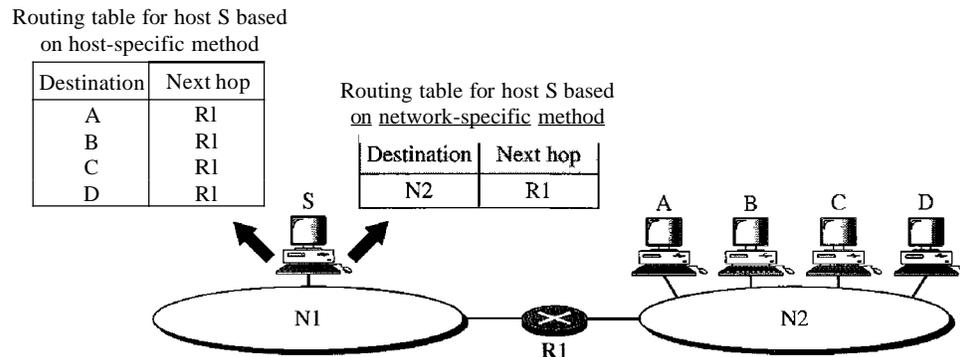#### *Network-Specific Method Versus Host-Specific Method*

A second technique to reduce the routing table and simplify the searching process is called the network-specific method. Here, instead of having an entry for every destination host connected to the same physical network (host-specific method), we have

Figure 22.2   *Route method versus next-hop method*

a. Routing tables based on route

Destination	Route
HostB	RI, R2, host B

Routing table for host A

Destination	Route
HostB	R2, host B

Routing table for R1

Destination	Route
HostB	HostB

Routing table for R2

b. Routing tables based on next hop

Destination	Next hop
Host B	R1

Destination	Next hop
HostB	R2

Destination	Next hop
Host B	

Host A   HostB

Network   R1   Network   R2   Network

only one entry that defines the address of the destination network itself. In other words, we treat all hosts connected to the same network as one single entity. For example, if 1000 hosts are attached to the same network, only one entry exists in the routing table instead of 1000. Figure 22.3 shows the concept.

Figure 22.3   *Host-specific versus network-specific method*

Routing table for host S based on host-specific method

Destination	Next hop
A	R1
B	R1
C	R1
D	R1

Routing table for host S based on network-specific method

Destination	Next hop
N2	R1

S   N1   R1   N2   A B C D

Host-specific routing is used for purposes such as checking the route or providing security measures.

Default Method

Another technique to simplify routing is called the default method. In Figure 22.4 host A is connected to a network with two routers. Router R1 routes the packets to hosts connected to network N2. However, for the rest of the Internet, router R2 is used. So instead of listing all networks in the entire Internet, host A can just have one entry called the *default* (normally defined as network address 0.0.0.0).

Figure 22.4   *Default method*
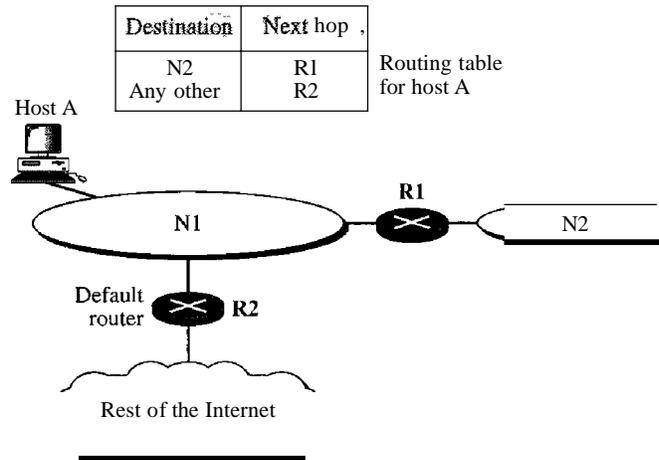
Forwarding Process

Let us discuss the forwarding process. We assume that hosts and routers use classless addressing because classful addressing can be treated as a special case of classless addressing. In classless addressing, the routing table needs to have one row of information for each block involved. The table needs to be searched based on the network address (first address in the block). Unfortunately, the destination address in the packet gives no clue about the network address. To solve the problem, we need to include the mask *(In)* in the table; we need to have an extra column that includes the mask for the corresponding block. Figure 22.5 shows a simple forwarding module for classless addressing.

Figure 22.5   *Simplified forwarding module in classless address*
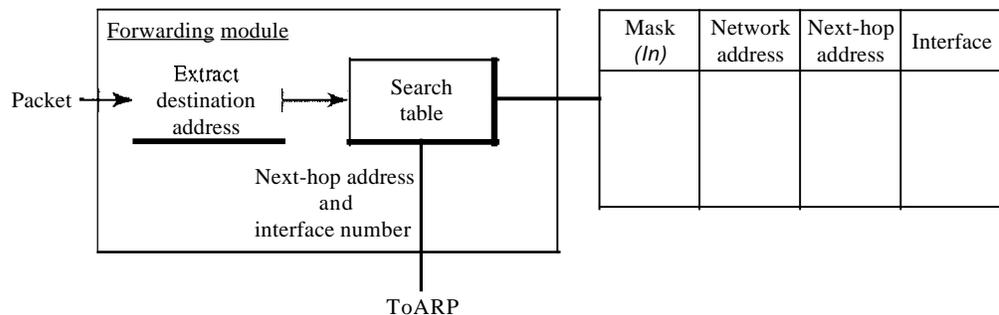
Note that we need at least four columns in our routing table; usually there are more.

In classless addressing, we need at least four colwnns in a routing table.

Example 22.1

Make a routing table for router R1, using the configuration in Figure 22.6.
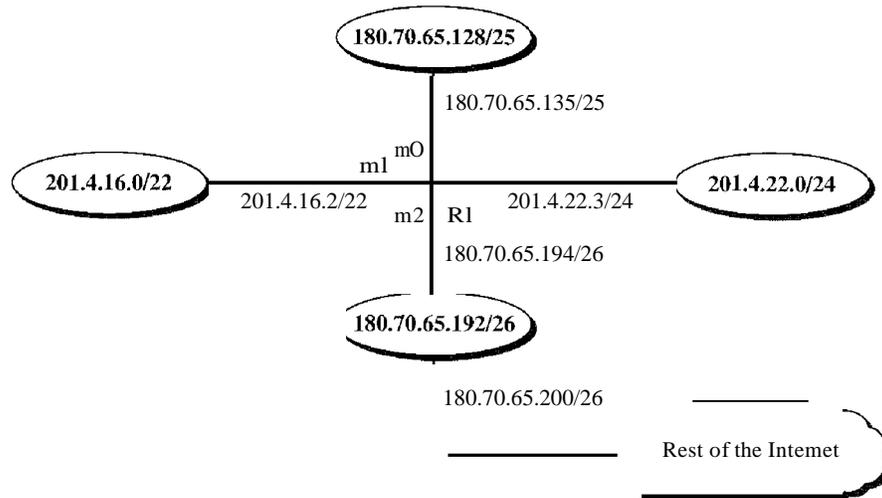
Figure 22.6   *Configuration for Example 22.1*

Solution

Table 22.1 shows the corresponding table.

Table 22.1   *Routing table for router RI in Figure 22.6*

Mask	Network Address	Next Hop	Interface
/26	180.70.65.192	-	m2
/25	180.70.65.128	-	mO
/24	201.4.22.0	-	m3
/22	201.4.16.0	....	m1
Any	Any	180.70.65.200	m2

Example 22.2

Show the forwarding process if a packet arrives at R1 in Figure 22.6 with the destination address 180.70.65.140.

Solution

The router performs the following steps:

1. The first mask (/26) is applied to the destination address. The result is 180.70.65.128, which does not match the corresponding network address.

2. The second mask (/25) is applied to the destination address. The result is 180.70.65.128, which matches the corresponding network address. The next-hop address (the destination address of the packet in this case) and the interface number mO are passed to ARP for further processing.

Example 22.3

Show the forwarding process if a packet arrives at R1 in Figure 22.6 with the destination address 201.4.22.35.

Solution

The router performs the following steps:

1. The first mask (/26) is applied to the destination address. The result is 201.4.22.0, which does not match the corresponding network address (row 1).
2. The second mask (/25) is applied to the destination address. The result is 201.4.22.0, which does not match the corresponding network address (row 2).
3. The third mask (/24) is applied to the destination address. The result is 201.4.22.0, which matches the corresponding network address. The destination address of the packet and the interface number m3 are passed to ARP.

Example 22.4

Show the forwarding process if a packet arrives at R1 in Figure 22.6 with the destination address 18.24.32.78.

Solution

This time all masks are applied, one by one, to the destination address, but no matching network address is found. When it reaches the end of the table, the module gives the next-hop address 180.70.65.200 and interface number m2 to ARP. This is probably an outgoing package that needs to be sent, via the default router, to someplace else in the Internet.
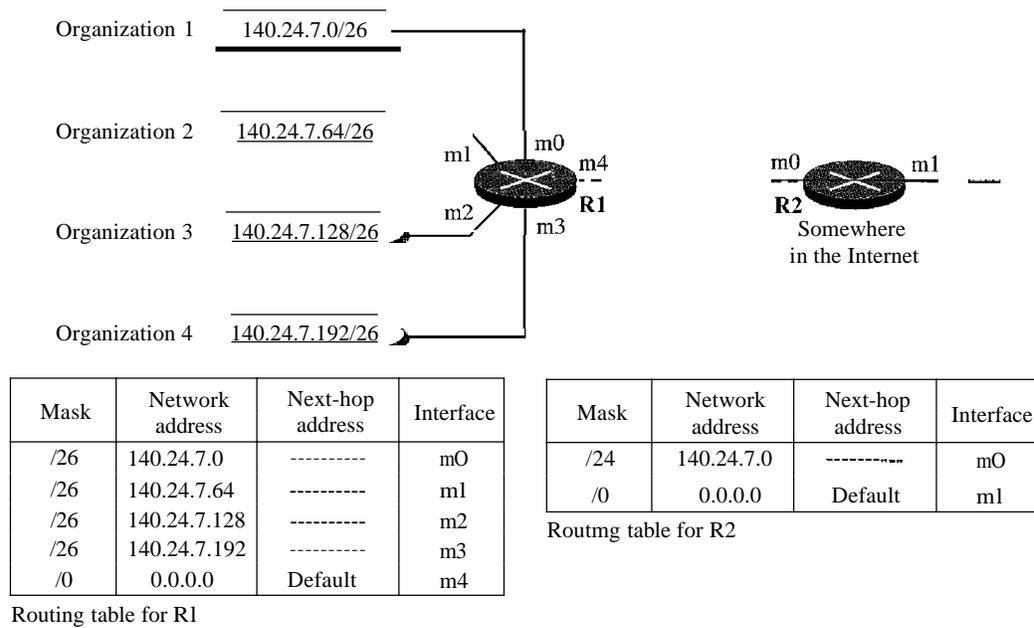
Address Aggregatioll

When we use classless addressing, it is likely that the number of routing table entries will increase. This is so because the intent of classless addressing is to divide up the whole address space into manageable blocks. The increased size of the table results in an increase in the amount of time needed to search the table. To alleviate the problem, the idea of address aggregation was designed. In Figure 22.7 we have two routers.

Router R1 is connected to networks of four organizations that each use 64 addresses. Router R2 is somewhere far from R1. Router R1 has a longer routing table because each packet must be correctly routed to the appropriate organization. Router R2, on the other hand, can have a very small routing table. For R2, any packet with destination 140.24.7.0 to 140.24.7.255 is sent out from interface mO regardless of the organization number. This is called address aggregation because the blocks of addresses for four organizations are aggregated into one larger block. Router R2 would have a longer routing table if each organization had addresses that could not be aggregated into one block.

Note that although the idea of address aggregation is similar to the idea of subnetting, we do not have a common site here; the network for each organization is independent. In addition, we can have several levels of aggregation.

Longest Mask Matching

What happens if one of the organizations in Figure 22.7 is not geographically close to the other three? For example, if organization 4 cannot be connected to router R1 for some reason, can we still use the idea of address aggregation and still assign block 140.24.7.192/26 to organization 4?

Figure 22.7   *Address aggregation*

Mask	Network address	Next-hop address	Interface
/26	140.24.7.0	----------	mO
/26	140.24.7.64	----------	ml
/26	140.24.7.128	----------	m2
/26	140.24.7.192	----------	m3
/0	0.0.0.0	Default	m4

Routing table for Rl

Mask	Network address	Next-hop address	Interface
/24	140.24.7.0	----------	mO
/0	0.0.0.0	Default	ml

Routmg table for R2

The answer is yes because routing in classless addressing uses another principle, **longest** mask **matching.** This principle states that the routing table is sorted from the longest mask to the shortest mask. In other words, if there are three masks *127, 126,* and *124,* the mask /27 must be the first entry and *124* must be last. Let us see if this principle solves the situation in which organization 4 is separated from the other three organizations. Figure 22.8 shows the situation.
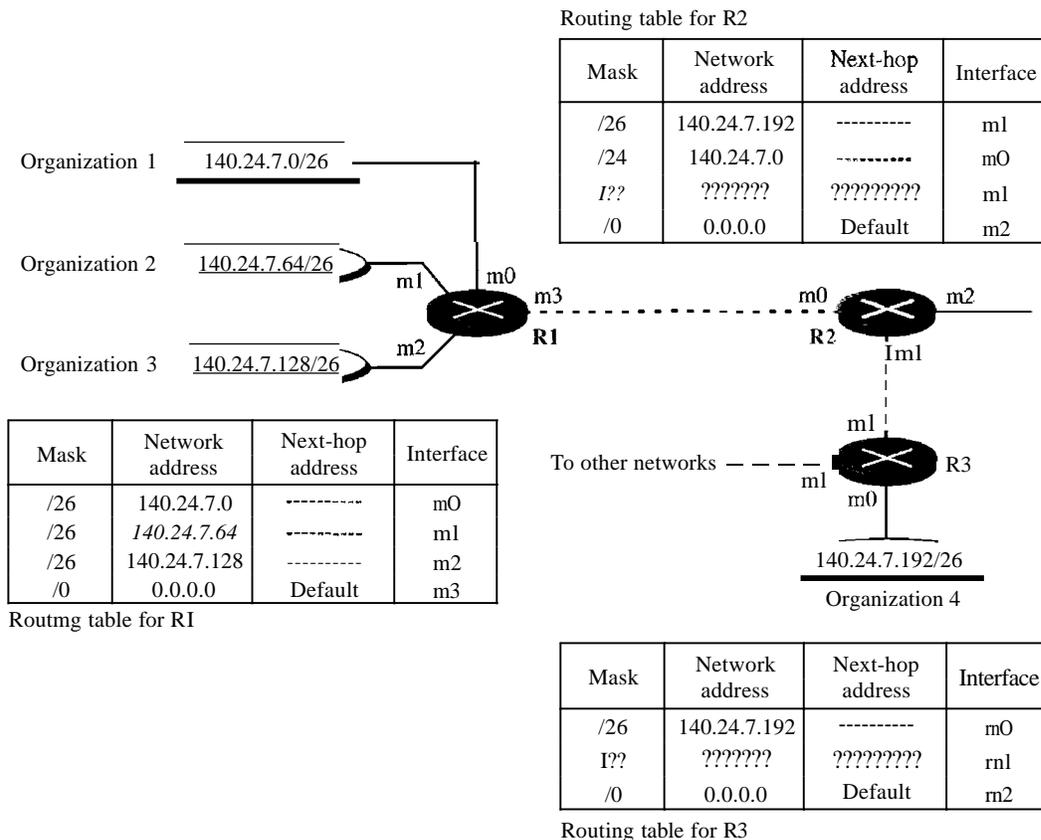
Suppose a packet arrives for organization 4 with destination address 140.24.7.200. The first mask at router R2 is applied, which gives the network address 140.24.7.192. The packet is routed correctly from interface ml and reaches organization 4. If, however, the routing table was not stored with the longest prefix first, applying the /24 mask would result in the incorrect routing of the packet to router Rl.

Hierarchical Routing

To solve the problem of gigantic routing tables, we can create a sense of hierarchy in the routing tables. In Chapter 1, we mentioned that the Internet today has a sense of hierarchy. We said that the Internet is divided into international and national ISPs. National ISPs are divided into regional ISPs, and regional ISPs are divided into local ISPs. If the routing table has a sense of hierarchy like the Internet architecture, the routing table can decrease in size.

Let us take the case of a local ISP. A local ISP can be assigned a single, but large block of addresses with a certain prefix length. The local ISP can divide this block into smaller blocks of different sizes and can assign these to individual users and organizations, both large and small. If the block assigned to the local ISP starts with a.b.c.dln, the ISP can create blocks starting with eJ.g.h/m, where *m* may vary for each customer and is greater than *n.*

Figure 22.8   *Longest mask matching*

Routing table for R2

Mask	Network address	Next-hop address	Interface
/26	140.24.7.192	----------	ml
/24	140.24.7.0	----------	mO
I??	*???????*	*?????????*	ml
/0	0.0.0.0	Default	m2

Organization 1   140.24.7.0/26

Organization 2   140.24.7.64/26   ml   mO

Organization 3   140.24.7.128/26   m2

m3   R1   m0   R2   m2

Iml

ml

To other networks — — —   ml   R3

m0

140.24.7.192/26

Organization 4

Mask	Network address	Next-hop address	Interface
/26	140.24.7.0	----------	mO
/26	*140.24.7.64*	----------	ml
/26	140.24.7.128	----------	m2
/0	0.0.0.0	Default	m3

Routmg table for RI

Mask	Network address	Next-hop address	Interface
/26	140.24.7.192	----------	mO
I??	???????	?????????	rnl
/0	0.0.0.0	Default	rn2

Routing table for R3

How does this reduce the size of the routing table? The rest of the Internet does not have to be aware of this division. All customers of the local ISP are defined as a.b.c.dln to the rest of the Internet. Every packet destined for one of the addresses in this large block is routed to the local ISP. There is only one entry in every router in the world for all these customers. They all belong to the same group. Of course, inside the local ISp, the router must recognize the subblocks and route the packet to the destined customer. If one of the customers is a large organization, it also can create another level of hierarchy by subnetting and dividing its subblock into smaller subblocks (or sub-subblocks). In classless routing, the levels of hierarchy are unlimited so long as we follow the rules of classless addressing.

Example 22.5

As an example of hierarchical routing, let us consider Figure 22.9. A regional ISP is granted 16,384 addresses starting from 120.14.64.0. The regional ISP has decided to divide this block into four subblocks, each with 4096 addresses. Three of these subblocks are assigned to three local ISPs; the second subblock is reserved for future use. Note that the mask for each block is *120* because the original block with mask /18 is divided into 4 blocks.

The first local ISP has divided its assigned subblock into 8 smaller blocks and assigned each to a small ISP. Each small ISP provides services to 128 households (HOOI to HI28), each using four addresses. Note that the mask for each small ISP is now *123* because the block is further divided into 8 blocks. Each household has a mask of *130,* because a household has only four addresses (2^{32-30} is 4).

Figure 22.9 *Hierarchical routing with ISPs*
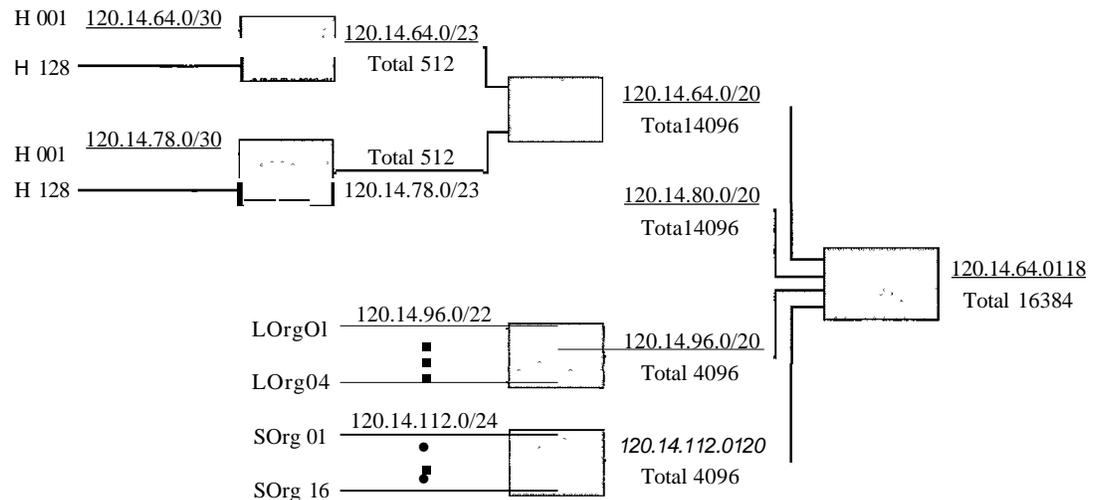
The second local ISP has divided its block into 4 blocks and has assigned the addresses to four large organizations (LOrg01 to LOrg04). Note that each large organization has 1024 addresses, and the mask is /22.

The third local ISP has divided its block into 16 blocks and assigned each block to a small organization (SOrg01 to SOrg16). Each small organization has 256 addresses, and the mask is /24.

There is a sense of hierarchy in this configuration. All routers in the Internet send a packet with destination address 120.14.64.0 to 120.14.127.255 to the regional ISP.

The regional ISP sends every packet with destination address 120.14.64.0 to 120.14.79.255 to local ISP1. Local ISP1 sends every packet with destination address 120.14.64.0 to 120.14.64.3 to H001.

Geographical Routing

To decrease the size of the routing table even further, we need to extend hierarchical routing to include geographical routing. We must divide the entire address space into a few large blocks. We assign a block to North America, a block to Europe, a block to Asia, a block to Africa, and so on. The routers of ISPs outside Europe will have only one entry for packets to Europe in their routing tables. The routers of ISPs outside North America will have only one entry for packets to North America in their routing tables. And so on.

Routing Table

Let us now discuss routing tables. A host or a router has a routing table with an entry for each destination, or a combination of destinations, to route IP packets. The routing table can be either static or dynamic.

Static Routing Table

A **static routing table** contains information entered manually. The administrator enters the route for each destination into the table. When a table is created, it cannot update

automatically when there is a change in the Internet. The table must be manually altered by the administrator.

A static routing table can be used in a small internet that does not change very often, or in an experimental internet for troubleshooting. It is poor strategy to use a static routing table in a big internet such as the Internet.

Dynamic Routing Table

A dynamic routing table is updated periodically by using one of the dynamic routing protocols such as RIP, OSPF, or BGP. Whenever there is a change in the Internet, such as a shutdown of a router or breaking of a link, the dynamic routing protocols update all the tables in the routers (and eventually in the host) automatically.

The routers in a big internet such as the Internet need to be updated dynamically for efficient delivery of the IP packets. We discuss in detail the three dynamic routing protocols later in the chapter.

Format

As mentioned previously, a routing table for classless addressing has a minimum of four columns. However, some of today's routers have even more columns. We should be aware that the number of columns is vendor-dependent, and not all columns can be found in all routers. Figure 22.10 shows some common fields in today's routers.

Figure 22.10   *Common fields in a routing table*

Mask	Network address	Next-hop address	Interlace		Reference count	Use

O   Mask. This field defines the mask applied for the entry.

O   Network address. This field defines the network address to which the packet is finally delivered. In the case of host-specific routing, this field defines the address of the destination host.

O   Next-hop address. This field defines the address of the next-hop router to which the packet is delivered.

D   Interface. This field shows the name of the interface.

D   Flags. This field defines up to five flags. Flags are on/off switches that signify either presence or absence. The five flags are U (up), G (gateway), H (host-specific), D (added by redirection), and M (modified by redirection).

   a. U (up). The U flag indicates the router is up and running. If this flag is not present, it means that the router is down. The packet cannot be forwarded and is discarded.

   b. G (gateway). The G flag means that the destination is in another network. The packet is delivered to the next-hop router for delivery (indirect delivery). When this flag is missing, it means the destination is in this network (direct delivery).

    c. H (host-specific). The H flag indicates that the entry in the network address field is a host-specific address. When it is missing, it means that the address is only the network address of the destination.

    d. D (added by redirection). The D flag indicates that routing information for this destination has been added to the host routing table by a redirection message from ICMP. We discussed redirection and the ICMP protocol in Chapter 21.

    e. M (modified by redirection). The M flag indicates that the routing information for this destination has been modified by a redirection message from ICMP. We discussed redirection and the ICMP protocol in Chapter 21.

O Reference count. This field gives the number of users of this route at the moment. For example, if five people at the same time are connecting to the same host from this router, the value of this column is 5.

O Use. This field shows the number of packets transmitted through this router for the corresponding destination.

Utilities

There are several utilities that can be used to find the routing information and the contents of a routing table. We discuss *netstat* and *ifconfig.*

Example 22.6

One utility that can be used to find the contents of a routing table for a host or router is *netstat* in UNIX or LINUX. The following shows the list of the contents of a default server. We have used two options, r and n. The option r indicates that we are interested in the routing table, and the option n indicates that we are looking for numeric addresses. Note that this is a routing table for a host, not a router. Although we discussed the routing table for a router throughout the chapter, a host also needs a routing table.

```
$ netstat-rn
Kernel IP routing table
```

| Destination	Gateway	Mask	Flags	Iface
153.18.16.0	0.0.0.0	255.255.240.0	U	ethO
127.0.0.0	0.0.0.0	255.0.0.0	U	10
0.0.0.0	153.18.31.254	0.0.0.0	UO	ethO

    Note also that the order of columns is different from what we showed. The destination column here defines the network address. The term *gateway* used by UNIX is synonymous with *router.* This column acmally defines the address of the next hop. The value 0.0.0.0 shows that the delivery is direct. The last entry has a flag of G, which means that the destination can be reached through a router (default router). The *Iface* defines the interface. The host has only one real interface, ethO, which means interface 0 connected to an Ethernet network. The second interface, la, is actually a virmalloopback interface indicating that the host accepts packets with loopback address 127.0.0.0.

    More information about the IP address and physical address of the server can be found by using the *ifconfig* command on the given interface (ethO).

```
$ ifconfig ethO
ethO   Link encap:Ethemet  HWaddr 00:BO:DO:DF:09:5D
inet addr:153.18.17.11  Bcast: 153.18.31.255  Mask:255.255.240.0
```

From the above information, we can deduce the configuration of the server, as shown in Figure 22.11.

Figure 22.11   *Configuration of the server for Example 22.6*
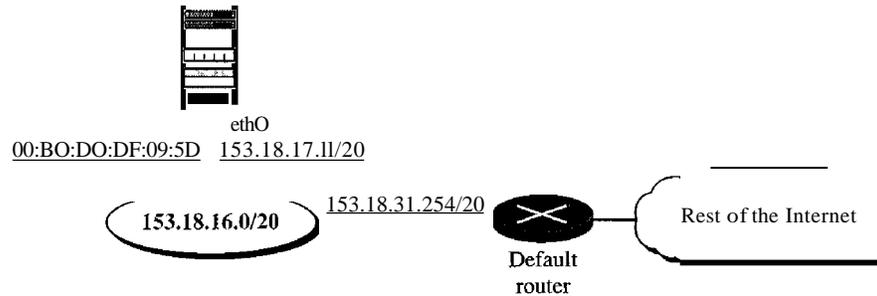
Note that the *ifconfig* command gives us the IP address and the physical (hardware) address of the interface.

22.3   UNICAST ROUTING PROTOCOLS

A routing table can be either static or dynamic. A *static table* is one with manual entries. A *dynamic table,* on the other hand, is one that is updated automatically when there is a change somewhere in the internet. Today, an internet needs dynamic routing tables. The tables need to be updated as soon as there is a change in the internet. For instance, they need to be updated when a router is down, and they need to be updated whenever a better route has been found.

Routing protocols have been created in response to the demand for dynamic routing tables. A routing protocol is a combination of rules and procedures that lets routers in the internet inform each other of changes. It allows routers to share whatever they know about the internet or their neighborhood. The sharing of information allows a router in San Francisco to know about the failure of a network in Texas. The routing protocols also include procedures for combining information received from other routers.

Optimization

A router receives a packet from a network and passes it to another network. A router is usually attached to several networks. When it receives a packet, to which network should it pass the packet? The decision is based on optimization: Which of the available pathways is the optimum pathway? What is the definition of the term *optimum?*

One approach is to assign a cost for passing through a network. We call this cost a metric. However, the metric assigned to each network depends on the type of protocol. Some simple protocols, such as the Routing Information Protocol (RIP), treat all networks as equals. The cost of passing through a network is the same; it is one hop count. So if a packet passes through 10 networks to reach the destination, the total cost is 10 hop counts.

Other protocols, such as Open Shortest Path First (OSPF), allow the administrator to assign a cost for passing through a network based on the type of service required. A route through a network can have different costs (metrics). For example, if maximum through-put is the desired type of service, a satellite link has a lower metric than a fiber-optic line. On the other hand, if minimum delay is the desired type of service, a fiber-optic line has a lower metric than a satellite link. Routers use routing tables to help decide the best route. OSPF protocol allows each router to have several routing tables based on the required type of service.

Other protocols define the metric in a totally different way. In the Border Gateway Protocol (BGP), the criterion is the policy, which can be set by the administrator. The policy defines what paths should be chosen.

Intra- and Interdomain Routing

Today, an internet can be so large that one routing protocol cannot handle the task of updating the routing tables of all routers. For this reason, an internet is divided into autonomous systems. An autonomous system (AS) is a group of networks and routers under the authority of a single administration. Routing inside an autonomous system is referred to as intradomain routing. Routing between autonomous systems is referred to as interdomain routing. Each autonomous system can choose one or more intrado-main routing protocols to handle routing inside the autonomous system. However, only one interdomain routing protocol handles routing between autonomous systems (see Figure 22.12).

Figure 22.12   *Autonomous systems*
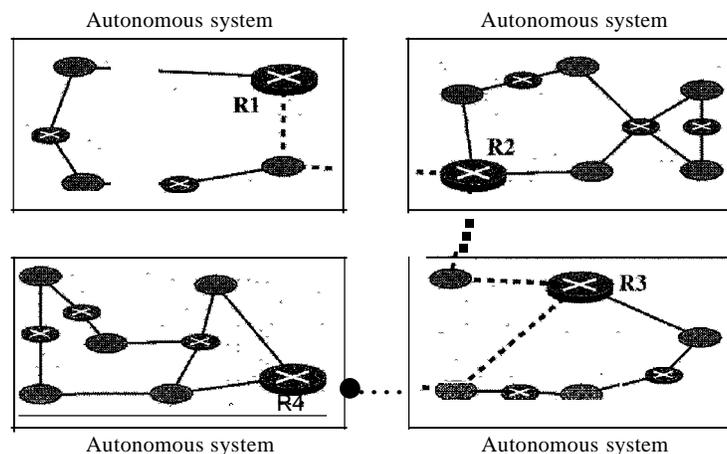
Several intradomain and interdomain routing protocols are in use. In this section, we cover only the most popular ones. We discuss two intradomain routing protocols: distance vector and link state. We also introduce one interdomain routing protocol: path vector (see Figure 22.13).

Routing Information Protocol (RIP) is an implementation of the distance vector protocol. Open Shortest Path First (OSPF) is an implementation of the link state proto-col. Border Gateway Protocol (BGP) is an implementation of the path vector protocol.

Figure 22.13   *Popular routing protocols*
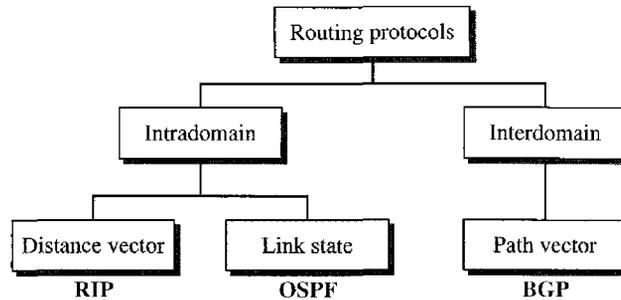
Distance Vector Routing

In distance vector routing, the least-cost route between any two nodes is the route with minimum distance. In this protocol, as the name implies, each node maintains a vector (table) of minimum distances to every node. The table at each node also guides the packets to the desired node by showing the next stop in the route (next-hop routing).

We can think of nodes as the cities in an area and the lines as the roads connecting them. A table can show a tourist the minimum distance between cities.

In Figure 22.14, we show a system of five nodes with their corresponding tables.

Figure 22.14   *Distance vector routing tables*
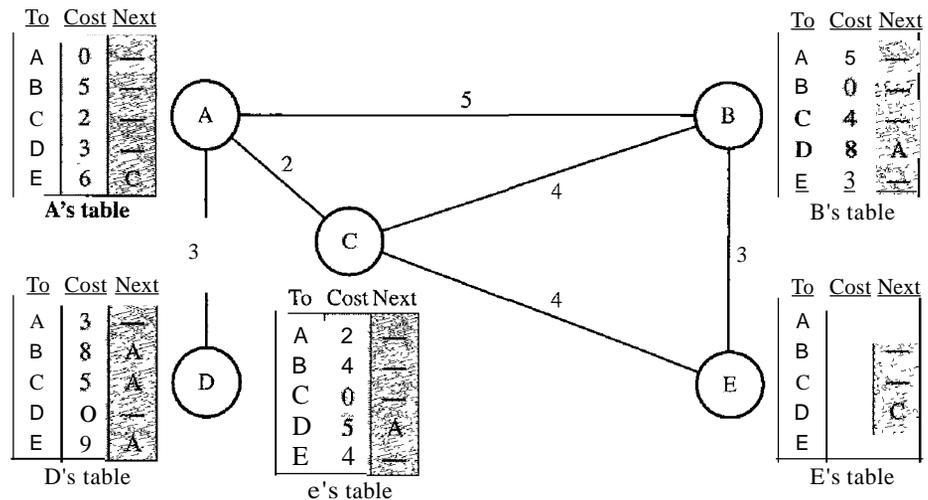
The table for node A shows how we can reach any node from this node. For example, our least cost to reach node E is 6. The route passes through C.

Initialization

The tables in Figure 22.14 are stable; each node knows how to reach any other node and the cost. At the beginning, however, this is not the case. Each node can know only

the distance between itself and its immediate neighbors, those directly connected to it. So for the moment, we assume that each node can send a message to the immediate neighbors and find the distance between itself and these neighbors. Figure 22.15 shows the initial tables for each node. The distance for any entry that is not a neighbor is marked as infinite (unreachable).

Figure 22.15   *Initialization of tables in distance vector routing*
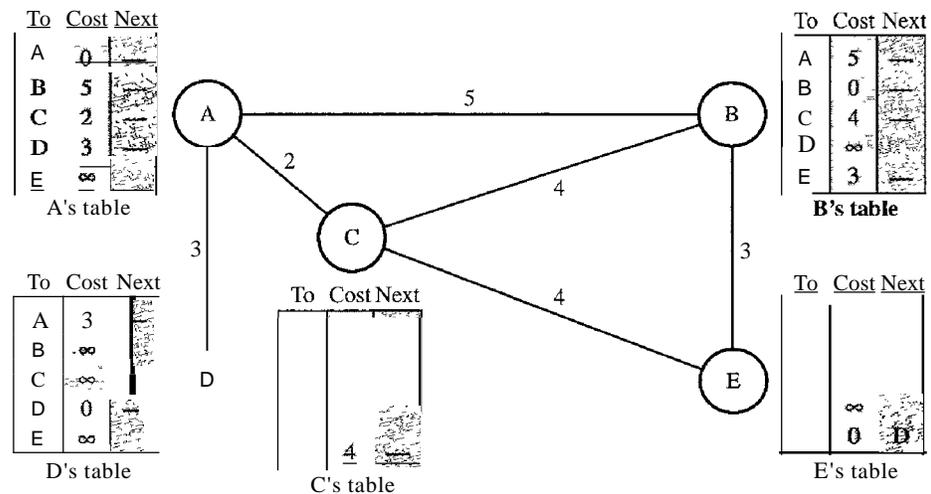
Sharing

The whole idea of distance vector routing is the sharing of information between neighbors. Although node A does not know about node E, node C does. So if node C shares its routing table with A, node A can also know how to reach node E. On the other hand, node C does not know how to reach node D, but node A does. If node A shares its routing table with node C, node C also knows how to reach node D. In other words, nodes A and C, as immediate neighbors, can improve their routing tables if they help each other.

There is only one problem. How much of the table must be shared with each neighbor? A node is not aware of a neighbor's table. The best solution for each node is to send its entire table to the neighbor and let the neighbor decide what part to use and what part to discard. However, the third column of a table (next stop) is not useful for the neighbor. When the neighbor receives a table, this column needs to be replaced with the sender's name. If any of the rows can be used, the next node is the sender of the table. A node therefore can send only the first two columns of its table to any neighbor. In other words, sharing here means sharing only the first two columns.

In distance vector routing, each node shares its routing table with its immediate neighbors periodically and when there is a change.


Updating

When a node receives a two-column table from a neighbor, it needs to update its routing table. Updating takes three steps:

1. The receiving node needs to add the cost between itself and the sending node to each value in the second column. The logic is clear. If node C claims that its distance to a destination is x mi, and the distance between A and C is y mi, then the distance between A and that destination, via C, is $x + y$ mi.

2. The receiving node needs to add the name of the sending node to each row as the third column if the receiving node uses information from any row. The sending node is the next node in the route.

3. The receiving node needs to compare each row of its old table with the corresponding row of the modified version of the received table.

   a. If the next-node entry is different, the receiving node chooses the row with the smaller cost. If there is a tie, the old one is kept.

   b. If the next-node entry is the same, the receiving node chooses the new row. For example, suppose node C has previously advertised a route to node X with distance 3. Suppose that now there is no path between C and X; node C now advertises this route with a distance of infinity. Node A must not ignore this value even though its old entry is smaller. The old route does not exist any more. The new route has a distance of infinity.

Figure 22.16 shows how node A updates its routing table after receiving the partial table from node C.

Figure 22.16   *Updating in distance vector routing*
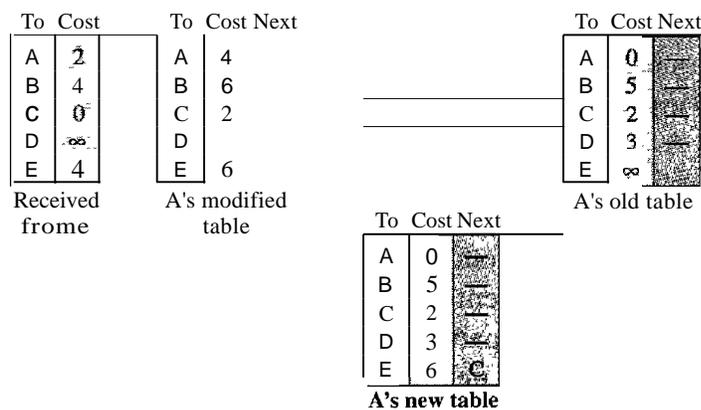
There are several points we need to emphasize here. First, as we know from mathematics, when we add any number to infinity, the result is still infinity. Second, the modified table shows how to reach A from A via C. If A needs to reach itself via C, it needs to go to C and come back, a distance of 4. Third, the only benefit from this updating of node A is the last entry, how to reach E. Previously, node A did not know how to reach E (distance of infinity); now it knows that the cost is 6 via C.

Each node can update its table by using the tables received from other nodes. In a short time, if there is no change in the network itself, such as a failure in a link, each node reaches a stable condition in which the contents of its table remains the same.

When to Share

The question now is, When does a node send its partial routing table (only two columns) to all its immediate neighbors? The table is sent both periodically and when there is a change in the table.

Periodic Update   A node sends its routing table, normally every 30 s, in a periodic update. The period depends on the protocol that is using distance vector routing.

Triggered Update   A node sends its two-column routing table to its neighbors any-time there is a change in its routing table. This is called a triggered update. The change can result from the following.

1. A node receives a table from a neighbor, resulting in changes in its own table after updating.
2. A node detects some failure in the neighboring links which results in a distance change to infinity.

Two-Node Loop Instability

A problem with distance vector routing is instability, which means that a network using this protocol can become unstable. To understand the problem, let us look at the scenario depicted in Figure 22.17.
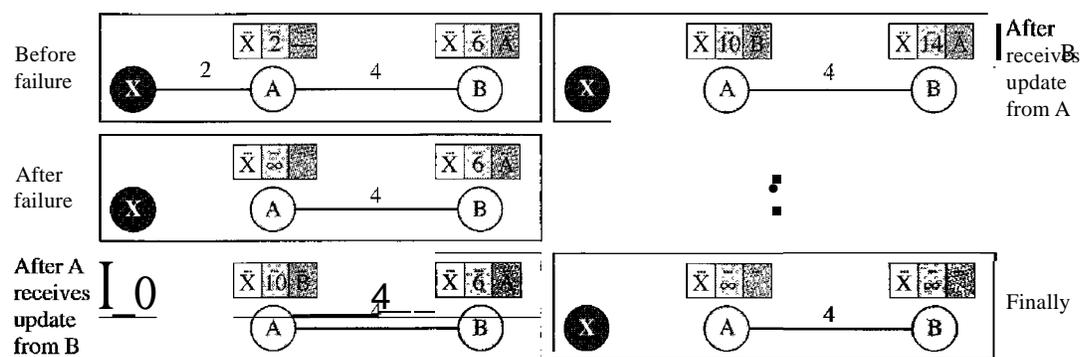
Figure 22.17   *Two-node instability*

Figure 22.17 shows a system with three nodes. We have shown only the portions of the routing table needed for our discussion. At the beginning, both nodes A and B know how to reach node X. But suddenly, the link between A and X fails. Node A changes its table. **If** A can send its table to B immediately, everything is fine. However, the system becomes unstable if B sends its routing table to A before receiving A's routing table. Node A receives the update and, assuming that B has found a way to reach X, immedi-ately updates its routing table. Based on the triggered update strategy, A sends its new

update to B. Now B thinks that something has been changed around A and updates its routing table. The cost of reaching X increases gradually until it reaches infinity. At this moment, both A and B know that X cannot be reached. However, during this time the system is not stable. Node A thinks that the route to X is via B; node B thinks that the route to X is via A. If A receives a packet destined for X, it goes to B and then comes back to A. Similarly, if B receives a packet destined for X, it goes to A and comes back to B. Packets bounce between A and B, creating a two-node loop problem. A few solutions have been proposed for instability of this kind.

Defining Infinity   The first obvious solution is to redefine infinity to a smaller number, such as 100. For our previous scenario, the system will be stable in less than 20 updates. As a matter of fact, most implementations of the distance vector protocol define the distance between each node to be I and define 16 as infinity. However, this means that the distance vector routing cannot be used in large systems. The size of the network, in each direction, can not exceed 15 hops.

Split Horizon   Another solution is called split horizon. In this strategy, instead of flooding the table through each interface, each node sends only part of its table through each interface. If, according to its table, node B thinks that the optimum route to reach X is via A, it does not need to advertise this piece of information to A; the information has corne from A (A already knows). Taking information from node A, modifying it, and sending it back to node A creates the confusion. In our scenario, node B eliminates the last line of its routing table before it sends it to A. In this case, node A keeps the value of infinity as the distance to X. Later when node A sends its routing table to B, node B also corrects its routing table. The system becomes stable after the first update: both node A and B know that X is not reachable.

Split Horizon and Poison Reverse   Using the split horizon strategy has one drawback. Normally, the distance vector protocol uses a timer, and if there is no news about a route, the node deletes the route from its table. When node B in the previous scenario eliminates the route to X from its advertisement to A, node A cannot guess that this is due to the split horizon strategy (the source of information was A) or because B has not received any news about X recently. The split horizon strategy can be combined with the poison reverse strategy. Node B can still advertise the value for X, but if the source of information is A, it can replace the distance with infinity as a warning: "Do not use this value; what I know about this route comes from you."

Three-Node Instability

The two-node instability can be avoided by using the split horizon strategy combined with poison reverse. However, if the instability is between three nodes, stability cannot be guaranteed. Figure 22.18 shows the scenario.

Suppose, after finding that X is not reachable, node A sends a packet to B and C to inform them of the situation. Node B immediately updates its table, but the packet to C is lost in the network and never reaches C. Node C remains in the dark and still thinks that there is a route to X via A with a distance of 5. After a while, node C sends to B its routing table, which includes the route to X. Node B is totally fooled here. It receives information on the route to X from C, and according to the algorithm, it updates its

Figure 22.18   *Three-node instability*
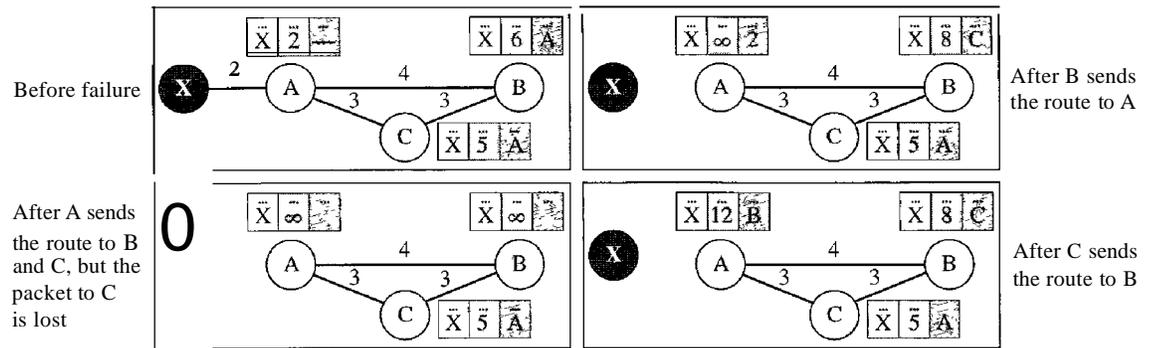
table, showing the route to X via C with a cost of 8. This information has come from C, not from A, so after awhile node B may advertise this route to A. Now A is fooled and updates its table to show that A can reach X via B with a cost of 12. Of course, the loop continues; now A advertises the route to X to C, with increased cost, but not to B. Node C then advertises the route to B with an increased cost. Node B does the same to A. And so on. The loop stops when the cost in each node reaches infinity.

RIP

The Routing Information Protocol (RIP) is an intradomain routing protocol used inside an autonomous system. It is a very simple protocol based on distance vector routing. RIP implements distance vector routing directly with some considerations:

1. In an autonomous system, we are dealing with routers and networks (links). The routers have routing tables; networks do not.
2. The destination in a routing table is a network, which means the first column defines a network address.
3. The metric used by RIP is very simple; the distance is defined as the number of links (networks) to reach the destination. For this reason, the metric in RIP is called a hop count.
4. Infinity is defined as 16, which means that any route in an autonomous system using RIP cannot have more than 15 hops.
5. The next-node column defines the address of the router to which the packet is to be sent to reach its destination.

Figure 22.19 shows an autonomous system with seven networks and four routers. The table of each router is also shown. Let us look at the routing table for R1. The table has seven entries to show how to reach each network in the autonomous system. Router R1 is directly connected to networks 130.10.0.0 and 130.11.0.0, which means that there are no next-hop entries for these two networks. To send a packet to one of the three networks at the far left, router R1 needs to deliver the packet to R2. The next-node entry for these three networks is the interface of router R2 with IP address 130.10.0.1. To send a packet to the two networks at the far right, router R1 needs to send the packet to the interface of router R4 with IP address 130.11.0.1. The other tables can be explained similarly.

Figure 22.19   *Example of a domain using RIP*

Dest.	Hop	Next
130.10.0.0	1	
130.11.0.0	2	130.10.0.2
195.2.4.0	1	
195.2.5.0	1	
195.2.6.0	2	195.2.5.2
205.5.5.0	3	130.10.0.2
205.5.6.0	3	130.10.0.2

R2 Table

Dest.	Hop	Next
130.10.0.0	2	195.2.5.1
130.11.0.0	3	195.2.5.1
195.2.4.0	2	195.2.5.1
195.2.5.0	1	
195.2.6.0	1	
205.5.5.0	4	195.2.5.1
205.5.6.0	4	195.2.5.1

R3 Table

Des!	Hop	Next
130.10.0.0	1	— —
130.11.0.0	1	— —
195.2.4.0	2	130.10.0.1
195.2.5.0	2	130.10.0.1
195.2.6.0	3	130.10.0.1
205.5.5.0	2	130.11.0.1
205.5.6.0	2	130.11.0.1

RI Table

Dest.	Hop	Next
130.10.0.0	2	130.11.0.2
130.11.0.0	1	
195.2.4.0	3	130.11.0.2
195.2.5.0	3	130.11.0.2
195.2.6.0	4	130.11.0.2
205.5.5.0	1	
205.5.6.0	1	

R4 Table

195.2.4.0   130.10.0.0   130.11.0.0   205.5.5.0
195.2.4.1   130::0.0.2   R1   205.5.5.1   R4
130.10.0.1   130.11.0.2   130.11.0.1   205.5.6.1
195.2.5.1   195.2.5.0   195.2.5.2   205.5.6.0
R2   R3
195.2.6.1
195.2.6.0

Link State Routing

Link state routing has a different philosophy from that of distance vector routing. In link state routing, if each node in the domain has the entire topology of the domain-the list of nodes and links, how they are connected including the type, cost (metric), and condition of the links (up or down)-the node can use Dijkstra's algorithm to build a routing table. Figure 22.20 shows the concept.

Figure 22.20   *Concept of link state routing*
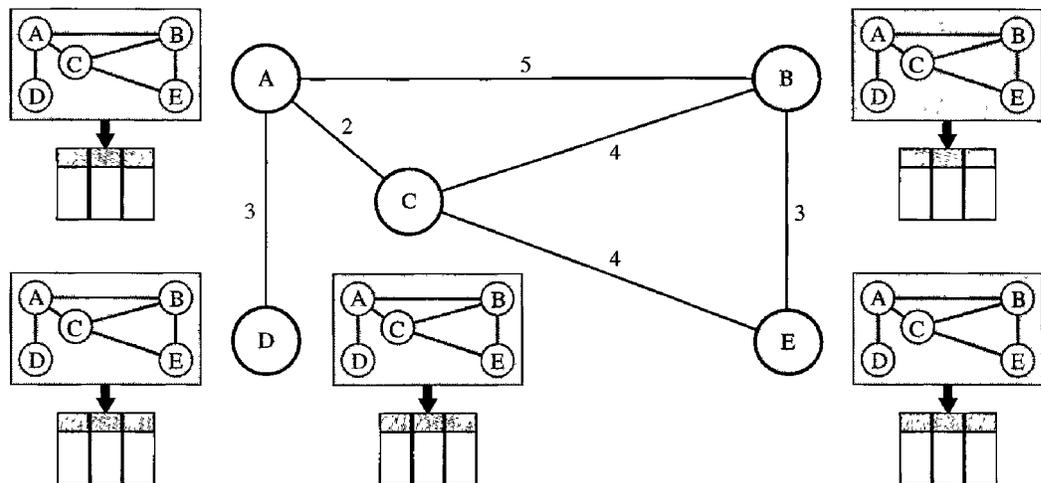
The figure shows a simple domain with five nodes. Each node uses the same topology to create a routing table, but the routing table for each node is unique because the calculations are based on different interpretations of the topology. This is analogous to a city map. While each person may have the same map, each needs to take a different route to reach her specific destination.

The topology must be dynamic, representing the latest state of each node and each link. If there are changes in any point in the network (a link is down, for example), the topology must be updated for each node.

How can a common topology be dynamic and stored in each node? No node can know the topology at the beginning or after a change somewhere in the network. Link state routing is based on the assumption that, although the global knowledge about the topology is not clear, each node has partial knowledge: it knows the state (type, condition, and cost) of its links. **In** other words, the whole topology can be compiled from the partial knowledge of each node. Figure 22.21 shows the same domain as in Figure 22.20, indicating the part of the knowledge belonging to each node.

Figure 22.21   *Link state knowledge*
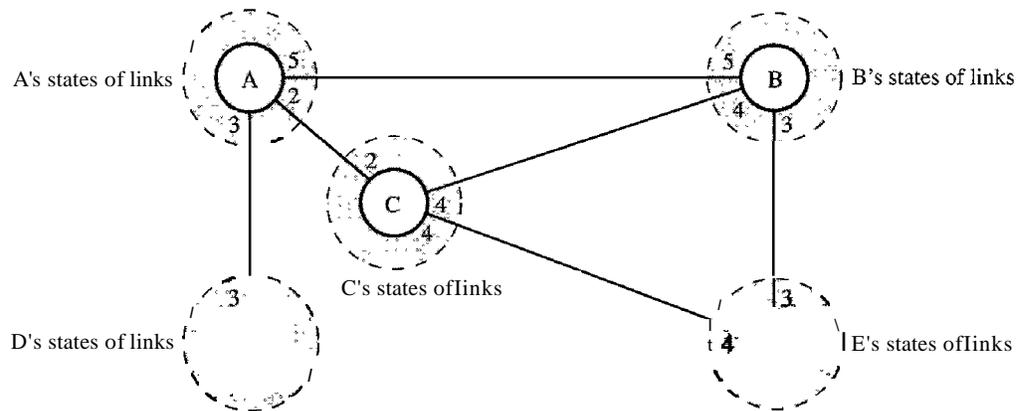
Node A knows that it is connected to node B with metric 5, to node C with metric 2, and to node D with metric 3. Node C knows that it is connected to node A with metric 2, to node B with metric 4, and to node E with metric 4. Node D knows that it is connected only to node A with metric 3. And so on. Although there is an overlap in the knowledge, the overlap guarantees the creation of a common topology-a picture of the whole domain for each node.

Building Routing Tables

In link state routing, four sets of actions are required to ensure that each node has the routing table showing the least-cost node to every other node.

1. Creation of the states of the links by each node, called the link state packet (LSP).
2. Dissemination of LSPs to every other router, called **flooding,** in an efficient and reliable way.
3. Formation of a shortest path tree for each node.
4. Calculation of a routing table based on the shortest path tree.

Creation of Link State Packet (LSP)   A link state packet can carry a large amount of information. For the moment, however, we assume that it carries a minimum amount

of data: the node identity, the list of links, a sequence number, and age. The first two, node identity and the list of links, are needed to make the topology. The third, sequence number, facilitates flooding and distinguishes new LSPs from old ones. The fourth, age, prevents old LSPs from remaining in the domain for a long time. LSPs are generated on two occasions:

1. *When there is a change in the topology of the domain.* Triggering of LSP dissemination is the main way of quickly informing any node in the domain to update its topology.

2. *On a periodic basis.* The period in this case is much longer compared to distance vector routing. As a matter of fact, there is no actual need for this type of LSP dissemination. It is done to ensure that old information is removed from the domain. The timer set for periodic dissemination is normally in the range of 60 min or 2 h based on the implementation. A longer period ensures that flooding does not create too much traffic on the network.

Flooding of LSPs   After a node has prepared an LSP, it must be disseminated to all other nodes, not only to its neighbors. The process is called flooding and based on the following:

1. The creating node sends a copy of the LSP out of each interface.

2. A node that receives an LSP compares it with the copy it may already have. If the newly arrived LSP is older than the one it has (found by checking the sequence number), it discards the LSP. If it is newer, the node does the following:

   a. It discards the old LSP and keeps the new one.

   b. It sends a copy of it out of each interface except the one from which the packet arrived. This guarantees that flooding stops somewhere in the domain (where a node has only one interface).

Formation of Shortest Path Tree: Dijkstra Algorithm   After receiving all LSPs, each node will have a copy of the whole topology. However, the topology is not sufficient to find the shortest path to every other node; a shortest path tree is needed.

A tree is a graph of nodes and links; one node is called the root. All other nodes can be reached from the root through only one single route. A shortest path tree is a tree in which the path between the root and every other node is the shortest. What we need for each node is a shortest path tree with that node as the root.

The Dijkstra algorithm creates a shortest path tree from a graph. The algorithm divides the nodes into two sets: tentative and permanent. It finds the neighbors of a current node, makes them tentative, examines them, and if they pass the criteria, makes them permanent. We can informally define the algorithm by using the flowchart in Figure 22.22.

Let us apply the algorithm to node A of our sample graph in Figure 22.23. To find the shortest path in each step, we need the cumulative cost from the root to each node, which is shown next to the node.

The following shows the steps. At the end of each step, we show the permanent (filled circles) and the tentative (open circles) nodes and lists with the cumulative costs.

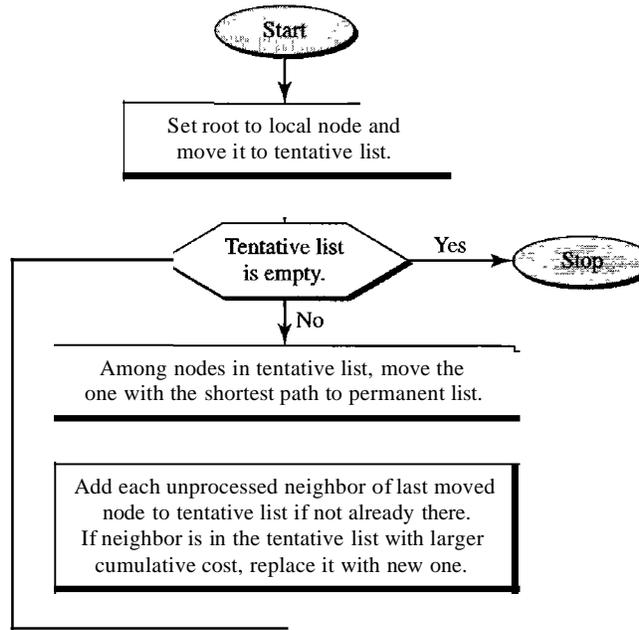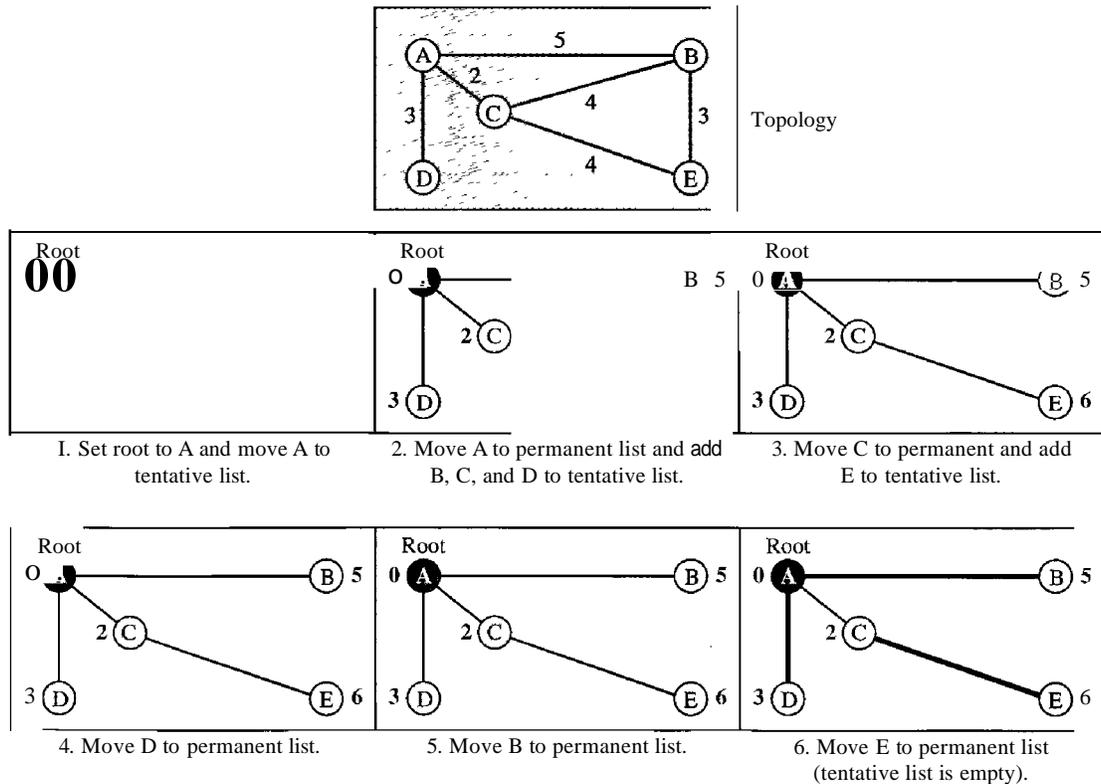**Figure 22.22** *Dijkstra algorithm*

Figure 22.23 *Example of formation of shortest path tree*

1. We make node A the root of the tree and move it to the tentative list. Our two lists are

   Permanent list: empty   Tentative list: A(O)

2. Node A has the shortest cumulative cost from all nodes in the tentative list. We move A to the pennanent list and add all neighbors of A to the tentative list. Our new lists are

   Permanent list: A(O)   Tentative list: B(5), C(2), D(3)

3. Node C has the shortest cumulative cost from all nodes in the tentative list. We move C to the permanent list. Node C has three neighbors, but node A is already processed, which makes the unprocessed neighbors just B and E. However, B is already in the tentative list with a cumulative cost of 5. Node A could also reach node B through C with a cumulative cost of 6. Since 5 is less than 6, we keep node B with a cumulative cost of 5 in the tentative list and do not replace it. Our new lists are

   Permanent list: A(O), e(2)   Tentative list: B(5), 0(3), E(6)

4. Node D has the shortest cumulative cost of all the nodes in the tentative list. We move D to the permanent list. Node D has no unprocessed neighbor to be added to the tentative list. Our new lists are

   Permanent list: A(O), C(2), 0(3)   Tentative list: B(5), E(6)

5. Node B has the shortest cumulative cost of all the nodes in the tentative list. We move B to the permanent list. We need to add all unprocessed neighbors of B to the tentative list (this is just node E). However, E(6) is already in the list with a smaller cumulative cost. The cumulative cost to node E, as the neighbor of B, is 8. We keep node E(6) in the tentative list. Our new lists are

   Permanent list: A(O), B(5), C(2), 0(3)   Tentative list: E(6)

6. Node E has the shortest cumulative cost from all nodes in the tentative list. We move E to the permanent list. Node E has no neighbor. Now the tentative list is empty. We stop; our shortest path tree is ready. The final lists are

   Permanent list: A(0), B(5), C(2), D(3), E(6)   Tentative list: empty

Calculation of Routing Table from Shortest Path Tree   Each node uses the shortest path tree protocol to construct its routing table. The routing table shows the cost of reaching each node from the root. Table 22.2 shows the routing table for node A.

Table 22.2   *Routing table for node A*

Node	Cost	Next Router
A	0	-
B	5	-
C	2	-
D	3	-
E	6	C

Compare Table 22.2 with the one in Figure 22.14. Both distance vector routing and link state routing end up with the same routing table for node A.

OSPF

The Open Shortest Path First or OSPF protocol is an intradomain routing protocol based on link state routing. Its domain is also an autonomous system.

Areas   To handle routing efficiently and in a timely manner, OSPF divides an autonomous system into areas. An area is a collection of networks, hosts, and routers all contained within an autonomous system. An autonomous system can be divided into many different areas. All networks inside an area must be connected.

Routers inside an area flood the area with routing information. At the border of an area, special routers called area border routers summarize the information about the area and send it to other areas. Among the areas inside an autonomous system is a special area called the *backbone;* all the areas inside an autonomous system must be connected to the backbone. In other words, the backbone serves as a primary area and the other areas as secondary areas. This does not mean that the routers within areas cannot be connected to each other, however. The routers inside the backbone are called the backbone routers. Note that a backbone router can also be an area border router.

If, because of some problem, the connectivity between a backbone and an area is broken, a virtual link between routers must be created by an administrator to allow continuity of the functions of the backbone as the primary area.

Each area has an area identification. The area identification of the backbone is zero. Figure 22.24 shows an autonomous system and its areas.

Figure 22.24   *Areas in an autonomous system*
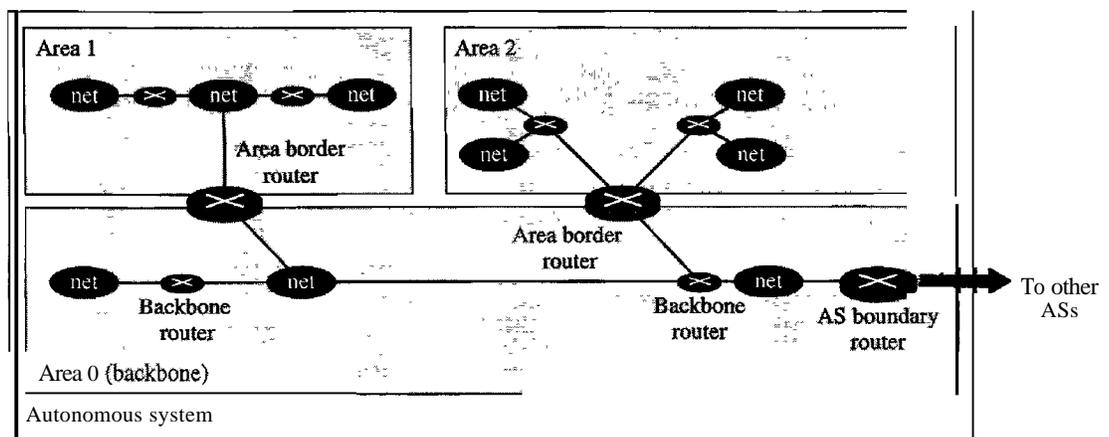
Metric   The OSPF protocol allows the administrator to assign a cost, called the metric, to each route. The metric can be based on a type of service (minimum delay, maximum throughput, and so on). As a matter of fact, a router can have multiple routing tables, each based on a different type of service.

Types of Links   In OSPF terminology, a connection is called a *link.* Four types of links have been defined: point-to-point, transient, stub, and virtual (see Figure 22.25).

Figure 22.25   *Types of links*
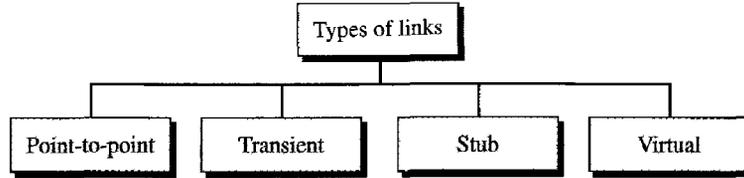
A point-to-point link connects two routers without any other host or router in between. In other words, the purpose of the link (network) is just to connect the two routers. An example of this type of link is two routers connected by a telephone line or a T line. There is no need to assign a network address to this type of link. Graphically, the routers are represented by nodes, and the link is represented by a bidirectional edge connecting the nodes. The metrics, which are usually the same, are shown at the two ends, one for each direction. In other words, each router has only one neighbor at the other side of the link (see Figure 22.26).

Figure 22.26   *Point-to-point link*
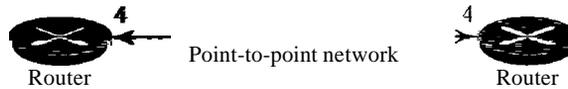
A transient link is a network with several routers attached to it. The data can enter through any of the routers and leave through any router. All LANs and some WANs with two or more routers are of this type. In this case, each router has many neighbors. For example, consider the Ethernet in Figure 22.27a. Router A has routers B, C, D, and E as neighbors. Router B has routers A, C, D, and E as neighbors. If we want to show the neighborhood relationship in this situation, we have the graph shown in Figure 22.27b.

Figure 22.27   *Transient link*

a. Transient network   b. Unrealistic representation   c. Realistic representation
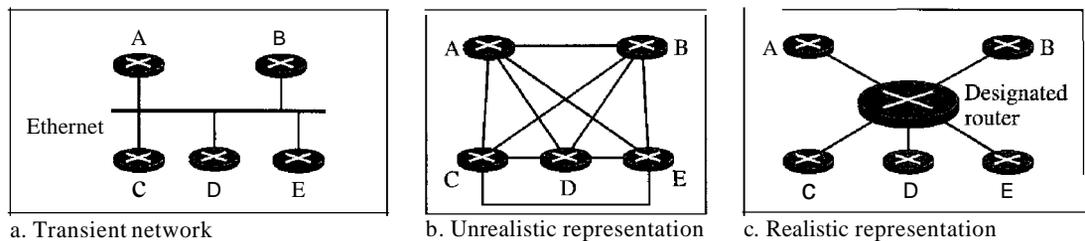
This is neither efficient nor realistic. It is not efficient because each router needs to advertise the neighborhood to four other routers, for a total of 20 advertisements. It is

not realistic because there is no single network (link) between each pair of routers; there is only one network that serves as a crossroad between all five routers.

To show that each router is connected to every other router through one single network, the network itself is represented by a node. However, because a network is not a machine, it cannot function as a router. One of the routers in the network takes this responsibility. It is assigned a dual purpose; it is a true router and a designated router. We can use the topology shown in Figure 22.27c to show the connections of a transient network.

Now each router has only one neighbor, the designated router (network). On the other hand, the designated router (the network) has five neighbors. We see that the number of neighbor announcements is reduced from 20 to 10. Still, the link is represented as a bidirectional edge between the nodes. However, while there is a metric from each node to the designated router, there is no metric from the designated router to any other node. The reason is that the designated router represents the network. We can only assign a cost to a packet that is passing through the network. We cannot charge for this twice. When a packet enters a network, we assign a cost; when a packet leaves the network to go to the router, there is no charge.

A **stub link** is a network that is connected to only one router. The data packets enter the network through this single router and leave the network through this same router. This is a special case of the transient network. We can show this situation using the router as a node and using the designated router for the network. However, the link is only one-directional, from the router to the network (see Figure 22.28).

Figure 22.28   *Stub link*


a. Stub network   b. Representation
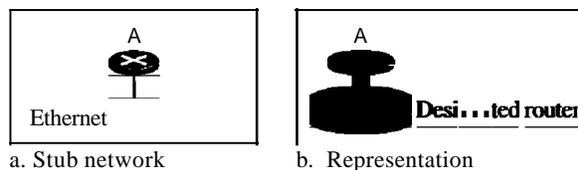
When the link between two routers is broken, the administration may create a **virtual link** between them, using a longer path that probably goes through several routers.

Graphical Representation   Let us now examine how an AS can be represented graphically. Figure 22.29 shows a small AS with seven networks and six routers. Two of the networks are point-to-point networks. We use symbols such as N1 and N2 for transient and stub networks. There is no need to assign an identity to a point-to-point network. The figure also shows the graphical representation of the AS as seen by OSPF.

We have used square nodes for the routers and ovals for the networks (represented by designated routers). However, OSPF sees both as nodes. Note that we have three stub networks.

Figure 22.29   *Example of an AS and its graphical representation in OSPF*

a. Autonomous system

b. Graphical representation
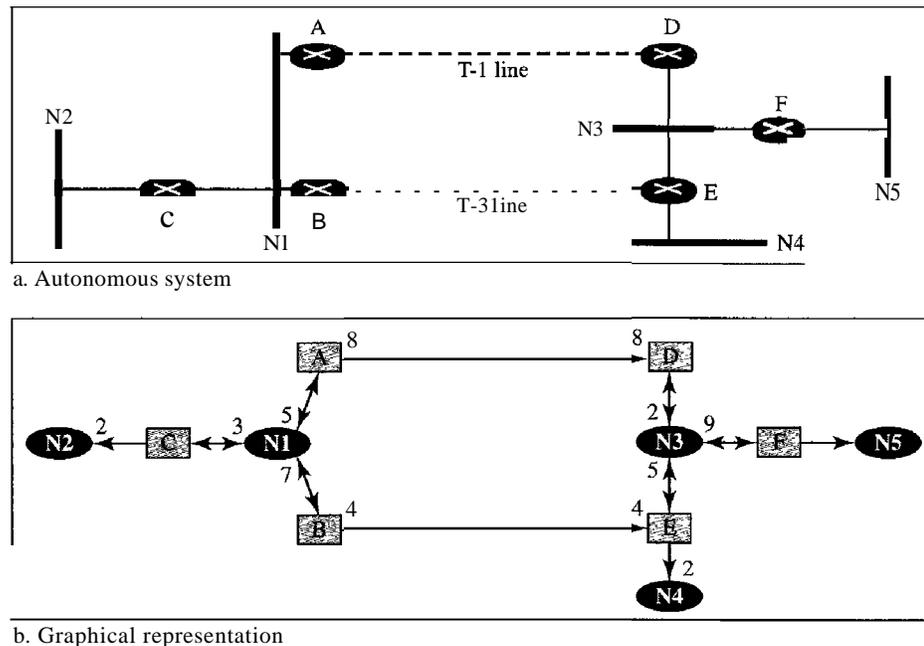
Path Vector Routing

Distance vector and link state routing are both intradomain routing protocols. They can be used inside an autonomous system, but not between autonomous systems. These two protocols are not suitable for interdomain routing mostly because of scalability. Both of these routing protocols become intractable when the domain of operation becomes large. Distance vector routing is subject to instability if there are more than a few hops in the domain of operation. Link state routing needs a huge amount of resources to calculate routing tables. It also creates heavy traffic because of flooding. There is a need for a third routing protocol which we call path vector routing.

Path vector routing proved to be useful for interdomain routing. The principle of path vector routing is similar to that of distance vector routing. In path vector routing, we assume that there is one node (there can be more, but one is enough for our conceptual discussion) in each autonomous system that acts on behalf of the entire autonomous system. Let us call it the speaker node. The speaker node in an AS creates a routing table and advertises it to speaker nodes in the neighboring ASs. The idea is the same as for distance vector routing except that only speaker nodes in each AS can communicate with each other. However, what is advertised is different. A speaker node advertises the path, not the metric of the nodes, in its autonomous system or other autonomous systems.

Initialization

At the beginning, each speaker node can know only the reachability of nodes inside its autonomous system. Figure 22.30 shows the initial tables for each speaker node in a system made of four ASs.

Figure 22.30 *Initial routing tables in path vector routing*
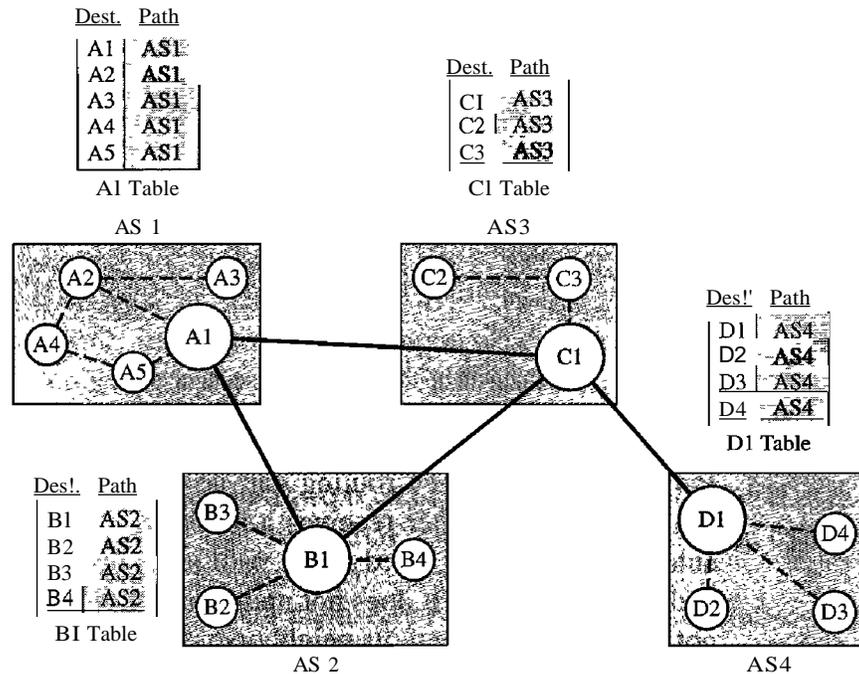
Node A1 is the speaker node for AS1, B1 for AS2, C1 for AS3, and D1 for AS4. Node A1 creates an initial table that shows A1 to A5 are located in ASI and can be reached through it. Node B1 advertises that B1 to B4 are located in AS2 and can be reached through B1. And so on.

Sharing   Just as in distance vector routing, in path vector routing, a speaker in an autonomous system shares its table with immediate neighbors. In Figure 22.30, node A1 shares its table with nodes B1 and C1. Node C1 shares its table with nodes D1, B1, andA1. Node B1 shares its table with C1 andA1. Node D1 shares its table with C1.

Updating   When a speaker node receives a two-column table from a neighbor, it updates its own table by adding the nodes that are not in its routing table and adding its own autonomous system and the autonomous system that sent the table. After a while each speaker has a table and knows how to reach each node in other ASs. Figure 22.31 shows the tables for each speaker node after the system is stabilized.

According to the figure, if router A1 receives a packet for nodes A3, it knows that the path is in ASI (the packet is at home); but if it receives a packet for D1, it knows that the packet should go from AS1, to AS2, and then to AS3. The routing table shows the path completely. On the other hand, if node D1 in AS4 receives a packet for node A2, it knows it should go through AS4, AS3, and AS 1.

O   Loop prevention. The instability of distance vector routing and the creation of loops can be avoided in path vector routing. When a router receives a message, it checks to see if its autonomous system is in the path list to the destination. If it is, looping is involved and the message is ignored.

Figure 22.31   *Stabilized tables for three autonomous systems*

Oest.	Path
A1	AS1
AS	AS1
B1	-AS2
B4	AS1-AS2
C1	AS1-AS3
C3	AS1-AS3
01	AS1-AS2-AS4
D4	AS1-AS2-AS4

A1 Table

Oest.	Path
A1	AS2-AS1
A5	AS2-AS1
B1	AS2
B4	AS2
C1	AS2-AS3
C3	AS2-AS3
D1	AS2-AS3-AS4
D4	AS2-AS3-AS4

B1 Table

Oest.	Path
A1	AS3-AS1
A5	AS3-AS1
B1	AS3-AS2
B4	AS3-AS2
C1	AS3
C3	AS3
D1	AS3-AS4
D4	AS3-AS4

Cl Table

Dest.	Path
A1	AS4-AS3-AS1
A5	AS4-AS3-AS1
B1	AS4-AS3-AS2
B4	AS4-AS3-AS2
C1	AS4-AS3
C3	AS4-AS3
D1	AS4
D4	AS4

D1 Table

O **Policy routing.** Policy routing can be easily implemented through path vector routing. When a router receives a message, it can check the path. **If** one of the autonomous systems listed in the path is against its policy, it can ignore that path and that destination. It does not update its routing table with this path, and it does not send this message to its neighbors.

O **Optimum path.** What is the optimum path in path vector routing? We are looking for a path to a destination that is the best for the organization that runs the autonomous system. We definitely cannot include metrics in this route because each autonomous system that is included in the path may use a different criterion for the metric. One system may use, internally, RIP, which defines hop count as the metric; another may use OSPF with minimum delay defined as the metric. The optimum path is the path that fits the organization. In our previous figure, each autonomous system may have more than one path to a destination. For example, a path from AS4 to ASI can be AS4-AS3-AS2-AS1, or it can be AS4-AS3-ASI. For the tables, we chose the one that had the smaller number of autonomous systems, but this is not always the case. Other criteria, such as security, safety, and reliability, can also be applied.

BGP

Border Gateway Protocol (BGP) is an interdomain routing protocol using path vector routing. It first appeared in 1989 and has gone through four versions.

Types **of Autonomous** Systems   As we said before, the Internet is divided into hierarchical domains called autonomous systems. For example, a large corporation that manages its own network and has full control over it is an autonomous system. A local ISP that provides services to local customers is an autonomous system. We can divide autonomous systems into three categories: stub, multihomed, and transit.

O **Stub** AS. A stub AS has only one connection to another AS. The interdomain data traffic in a stub AS can be either created or terminated in the AS. The hosts in the AS can send data traffic to other ASs. The hosts in the AS can receive data coming from hosts in other ASs. Data traffic, however, cannot pass through a stub AS. A stub AS

is either a source or a sink. A good example of a stub AS is a small corporation or a small local ISP.

O Multihomed AS. A multihomed AS has more than one connection to other ASs, but it is still only a source or sink for data traffic. It can receive data traffic from more than one AS. It can send data traffic to more than one AS, but there is no transient traffic. It does not allow data coming from one AS and going to another AS to pass through. A good example of a multihomed AS is a large corporation that is connected to more than one regional or national AS that does not allow transient traffic.

O Transit AS. A transit AS is a multihomed AS that also allows transient traffic. Good examples of transit ASs are national and international ISPs (Internet backbones).

Path Attributes   In our previous example, we discussed a path for a destination network. The path was presented as a list of autonomous systems, but is, in fact, a list of attributes. Each attribute gives some information about the path. The list of attributes helps the receiving router make a more-informed decision when applying its policy.

Attributes are divided into two broad categories: well known and optional. A well-known attribute is one that every BGP router must recognize. An optional attribute is one that needs not be recognized by every router.

Well-known attributes are themselves divided into two categories: mandatory and discretionary. A *well-known mandatory attribute* is one that must appear in the description of a route. A *well-known discretionary attribute* is one that must be recognized by each router, but is not required to be included in every update message. One well-known mandatory attribute is ORIGIN. This defines the source of the routing information (RIP, OSPF, and so on). Another well-known mandatory attribute is AS_PATH. This defines the list of autonomous systems through which the destination can be reached. Still another well-known mandatory attribute is NEXT-HOP, which defines the next router to which the data packet should be sent.

The optional attributes can also be subdivided into two categories: transitive and nontransitive. An *optional transitive attribute* is one that must be passed to the next router by the router that has not implemented this attribute. An *optional nontransitive attribute* is one that must be discarded if the receiving router has not implemented it.

BGP Sessions   The exchange of routing information between two routers using BGP takes place in a session. A session is a connection that is established between two BGP routers only for the sake of exchanging routing information. To create a reliable environment, BGP uses the services of TCP. In other words, a session at the BGP level, as an application program, is a connection at the TCP level. However, there is a subtle difference between a connection in TCP made for BGP and other application programs. When a TCP connection is created for BGP, it can last for a long time, until something unusual happens. For this reason, BGP sessions are sometimes referred to as *semi-pennanent connections*.

External and Internal BGP   If we want to be precise, BGP can have two types of sessions: external BGP (E-BGP) and internal BGP (I-BGP) sessions. The E-BGP session is used to exchange information between two speaker nodes belonging to two different autonomous systems. The I-BGP session, on the other hand, is used to exchange routing information between two routers inside an autonomous system. Figure 22.32 shows the idea.

Figure 22.32 *Internal and external BGP sessions*
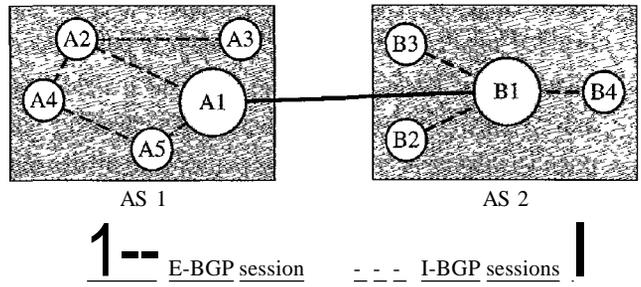
The session established between AS 1 and AS2 is an E-BOP session. The two speaker routers exchange information they know about networks in the Internet. However, these two routers need to collect information from other routers in the autonomous systems. This is done using I-BOP sessions.

22.4   MULTICAST ROUTING PROTOCOLS

In this section, we discuss multicasting and multicast routing protocols. We first define the term *multicasting* and compare it to unicasting and broadcasting. We also briefly discuss the applications of multicasting. Finally, we move on to multicast routing and the general ideas and goals related to it. We also discuss some common multicast routing protocols used in the Internet today.

Unicast, Multicast, **and** Broadcast

A message can be unicast, multicast, or broadcast. Let us clarify these terms as they relate to the Internet.

Unicasting

In unicast communication, there is one source and one destination. The relationship between the source and the destination is one-to-one. In this type of communication, both the source and destination addresses, in the IP datagram, are the unicast addresses assigned to the hosts (or host interfaces, to be more exact). In Figure 22.33, a unicast

Figure 22.33   *Unicasting*
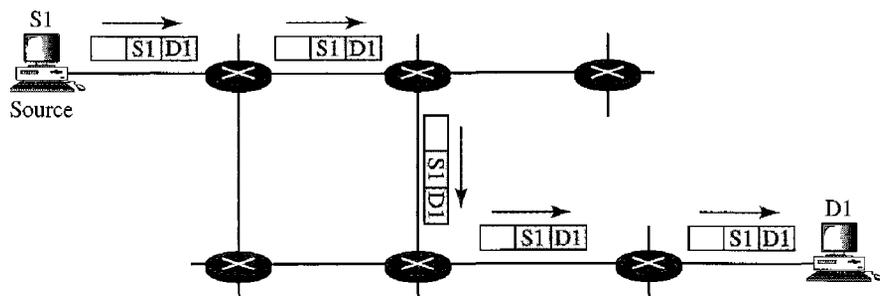
packet starts from the source S1 and passes through routers to reach the destination D1. We have shown the networks as a link between the routers to simplify the figure.
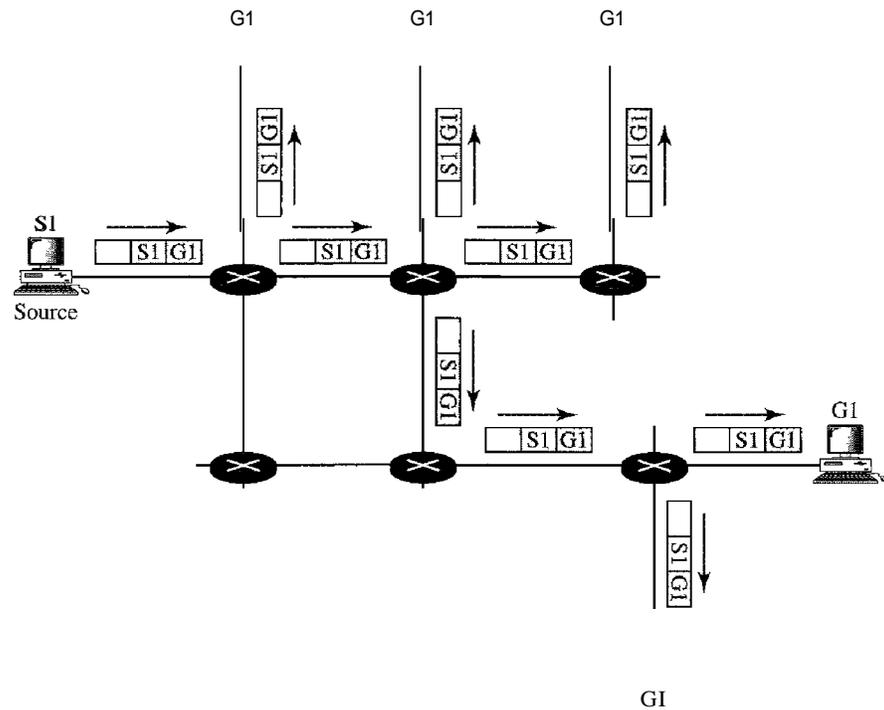
Note that in unicasting, when a router receives a packet, it forwards the packet through only one of its interfaces (the one belonging to the optimum path) as defined in the routing table. The router may discard the packet if it cannot find the destination address in its routing table.

In unicasting, the router forwards the received packet through
only one of its interfaces.

Multicasting

In multicast communication, there is one source and a group of destinations. The relationship is one-to-many. In this type of communication, the source address is a unicast address, but the destination address is a group address, which defines one or more destinations. The group address identifies the members of the group. Figure 22.34 shows the idea behind multicasting.

Figure 22.34   *Multicasting*

A multicast packet starts from the source S1 and goes to all destinations that belong to group G1. In multicasting, when a router receives a packet, it may forward it through several of its interfaces.

In multicasting, the router may forward the received packet
through several of its interfaces.

Broadcasting

In broadcast communication, the relationship between the source and the destination is one-to-all. There is only one source, but all the other hosts are the destinations. The Internet does not explicitly support broadcasting because of the huge amount of traffic it would create and because of the bandwidth it would need. Imagine the traffic generated in the Internet if one person wanted to send a message to everyone else connected to the Internet.
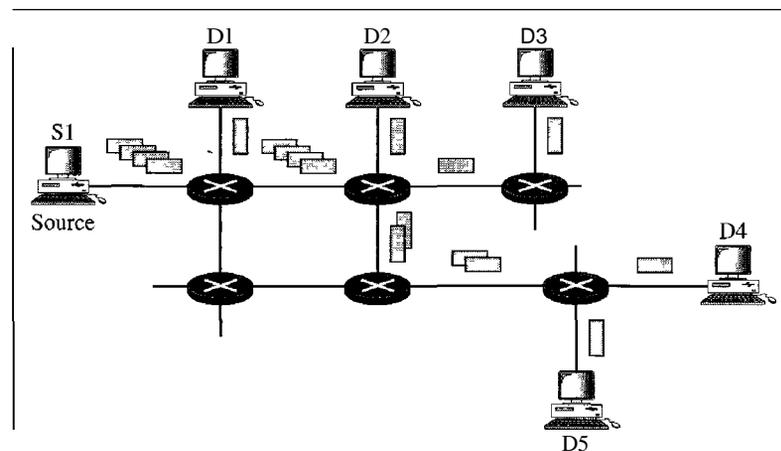
Multicasting Versus Multiple Unicasting

Before we finish this section, we need to distinguish between multicasting and multiple unicasting. Figure 22.35 illustrates both concepts.

Figure 22.35   *Multicasting versus multiple unicasting*
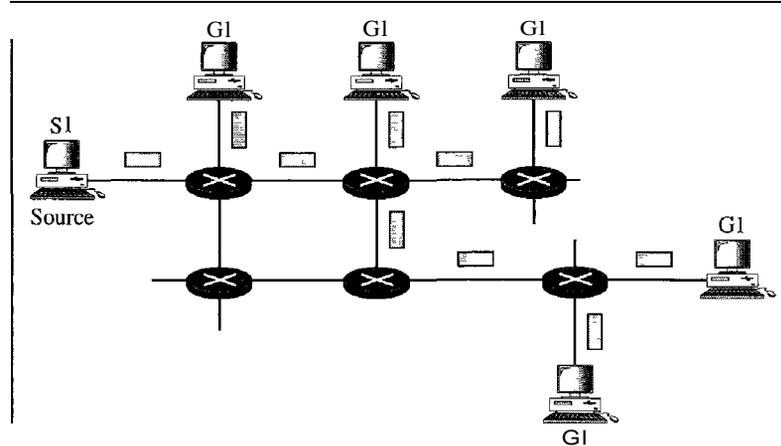
a. Multicasting

b. Multiple nnicasting

Multicasting starts with one single packet from the source that is duplicated by the routers. The destination address in each packet is the same for all duplicates. Note that only one single copy of the packet travels between any two routers.

In multiple unicasting, several packets start from the source. If there are five destinations, for example, the source sends five packets, each with a different unicast destination address. Note that there may be multiple copies traveling between two routers. For example, when a person sends an e-mail message to a group of people, this is multiple unicasting. The e-mail software creates replicas of the message, each with a different destination address and sends them one by one. This is not multicasting; it is multiple unicasting.

Emulation of Multicasting with Unicasting

You might wonder why we have a separate mechanism for multicasting, when it can be emulated with unicasting. There are two obvious reasons for this.

1. Multicasting is more efficient than multiple unicasting. In Figure 22.35, we can see how multicasting requires less bandwidth than does multiple unicasting. In multiple unicasting, some of the links must handle several copies.

2. In multiple unicasting, the packets are created by the source with a relative delay between packets. If there are 1000 destinations, the delay between the first and the last packet may be unacceptable. In multicasting, there is no delay because only one packet is created by the source.

Emulation of multicasting through multiple unicasting is not efficient
and may create long delays, particularly with a large group.

Applications

Multicasting has many applications today such as access to distributed databases, information dissemination, teleconferencing, and distance learning.

Access to Distributed Databases

Most of the large databases today are distributed. That is, the information is stored in more than one location, usually at the time of production. The user who needs to access the database does not know the location of the information. A user's request is multicast to all the database locations, and the location that has the information responds.

Information Dissemination

Businesses often need to send information to their customers. If the nature of the information is the same for each customer, it can be multicast. In this way a business can send one message that can reach many customers. For example, a software update can be sent to all purchasers of a particular software package.

Dissemination of News

In a similar manner news can be easily disseminated through multicasting. One single message can be sent to those interested in a particular topic. For example, the statistics of the championship high school basketball tournament can be sent to the sports editors of many newspapers.

Teleconferencing

Teleconferencing involves multicasting. The individuals attending a teleconference all need to receive the same information at the same time. Temporary or permanent groups can be formed for this purpose. For example, an engineering group that holds meetings every Monday morning could have a permanent group while the group that plans the holiday party could form a temporary group.

Distance Learning

One growing area in the use of multicasting is distance learning. Lessons taught by one single professor can be received by a specific group of students. This is especially convenient for those students who find it difficult to attend classes on campus.

Multicast Routing

In this section, we first discuss the idea of optimal routing, common in all multicast protocols. We then give an overview of multicast routing protocols.

Optimal Routing: Shortest Path Trees

The process of optimal interdomain routing eventually results in the finding of the *shortest path tree.* The root of the tree is the source, and the leaves are the potential destinations. The path from the root to each destination is the shortest path. However, the number of trees and the formation of the trees in unicast and multicast routing are different. Let us discuss each separately.

Unicast Routing   In unicast routing, when a router receives a packet to forward, it needs to find the shortest path to the destination of the packet. The router consults its routing table for that particular destination. The next-hop entry corresponding to the destination is the start of the shortest path. The router knows the shortest path for each destination, which means that the router has a shortest path tree to optimally reach all destinations. In other words, each line of the routing table is a shortest path; the whole routing table is a shortest path tree. In unicast routing, each router needs only one shortest path tree to forward a packet; however, each router has its own shortest path tree. Figure 22.36 shows the situation.

   The figure shows the details of the routing table and the shortest path tree for router R1. Each line in the routing table corresponds to one path from the root to the corresponding network. The whole table represents the shortest path tree.

In unicast routing, each router in the domain has a table that defines
a shortest path tree to possible destinations.

Multicast Routing   When a router receives a multicast packet, the situation is different from when it receives a unicast packet. A multicast packet may have destinations in more than one network. Forwarding of a single packet to members of a group requires a shortest path tree. If we have n groups, we may need n shortest path trees. We can imagine the complexity of multicast routing. Two approaches have been used to solve the problem: source-based trees and group-shared trees.

Figure 22.36   *Shortest path tree in unicast routing*
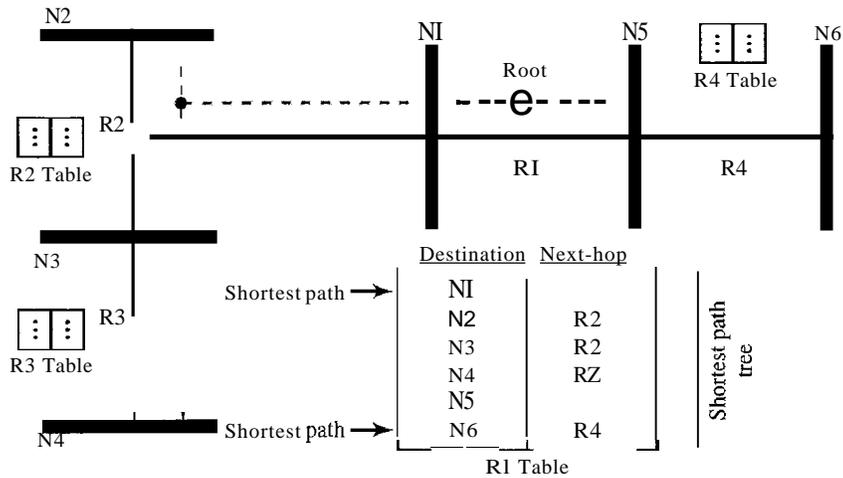
In multicast routing, each involved router needs to construct
a shortest path tree for each group.

O   Source-Based Tree. In the source-based tree approach, each router needs to have
one shortest path tree for each group. The shortest path tree for a group defines the
next hop for each network that has loyal member(s) for that group. In Figure 22.37,
we assume that we have only five groups in the domain: GI, G2, G3, G4, and G5.
At the moment GI has loyal members in four networks, G2 in three, G3 in two, G4
in two, and G5 in two. We have shown the names of the groups with loyal mem-
bers on each network. Figure 22.37 also shows the multicast routing table for
router RI. There is one shortest path tree for each group; therefore there are five
shortest path trees for five groups. If router R1 receives a packet with destination

Figure 22.37   *Source-based tree approach*
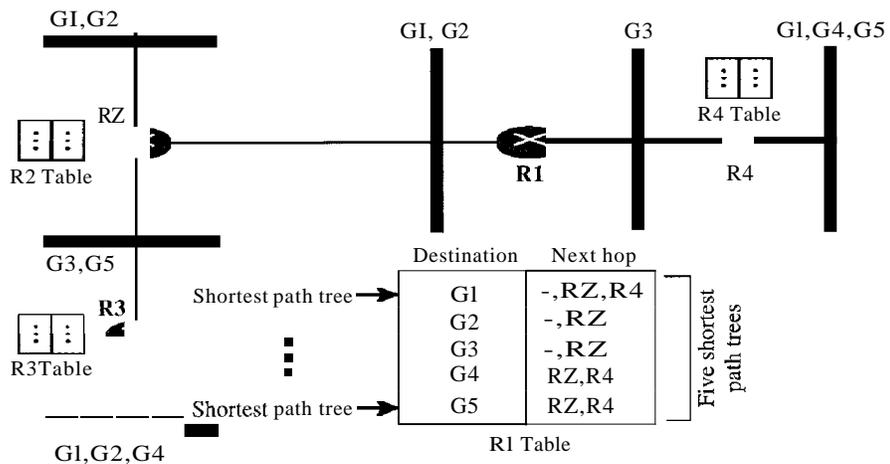
address G1, it needs to send a copy of the packet to the attached network, a copy to router R2, and a copy to router R4 so that all members of G1 can receive a copy. In this approach, if the number of groups is *m,* each router needs to have *m* shortest path trees, one for each group. We can imagine the complexity of the routing table if we have hundreds or thousands of groups. However, we will show how different protocols manage to alleviate the situation.

In the source-based tree approach, each router needs
to have one shortest path tree for each group.

O Group-Shared Tree. In the group-shared tree approach, instead of each router having *m* shortest path trees, only one designated router, called the center core, or rendezvous router, takes the responsibility of distributing multicast traffic. The core has *m* shortest path trees in its routing table. The rest of the routers in the domain have none. If a router receives a multicast packet, it encapsulates the packet in a unicast packet and sends it to the core router. The core router removes the multicast packet from its capsule, and consults its routing table to route the packet. Figure 22.38 shows the idea.

Figure 22.38 *Group-shared tree approach*
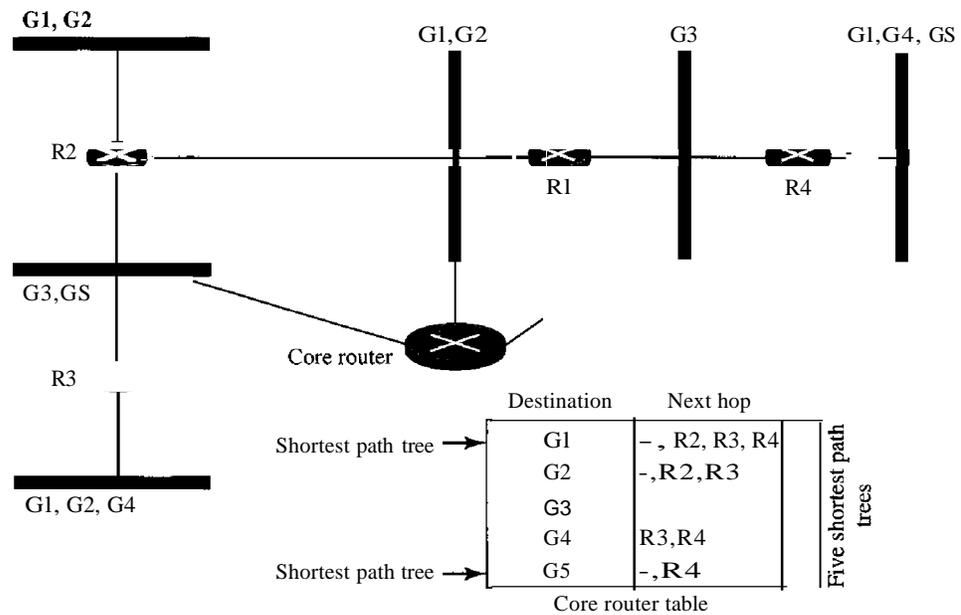
In the group-shared tree approach, only the core router,
which has a shortest path tree for each group, is involved in multicasting.

Routing Protocols

During the last few decades, several multicast routing protocols have emerged. Some of these protocols are extensions of unicast routing protocols; others are totally new.

We discuss these protocols in the remainder of this chapter. Figure 22.39 shows the taxonomy of these protocols.

Figure 22.39 *Taxonomy of common multicast protocols*
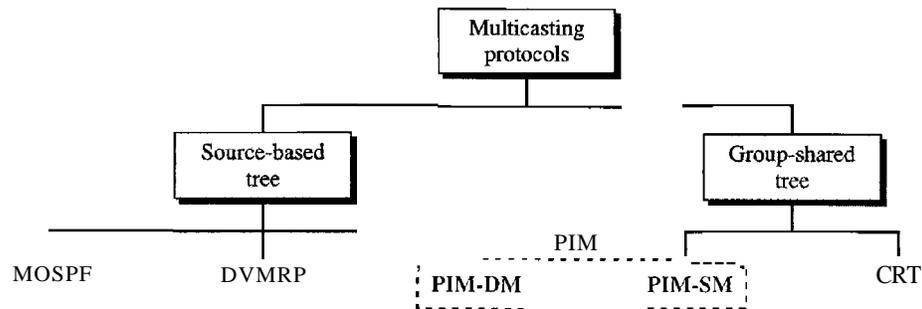
Multicast Link State Routing: MOSPF

In this section, we briefly discuss multicast link state routing and its implementation in the Internet, MOSPF.

Multicast Link State Routing   We discussed unicast link state routing in Section 22.3. We said that each router creates a shortest path tree by using Dijkstra's algorithm. The routing table is a translation of the shortest path tree. Multicast link state routing is a direct extension of unicast routing and uses a source-based tree approach. Although unicast routing is quite involved, the extension to multicast routing is very simple and straightforward.

Multicast link state routing uses the source-based tree approach.

Recall that in unicast routing, each node needs to advertise the state of its links. For multicast routing, a node needs to revise the interpretation of *state*. A node advertises every group which has any loyal member on the link. Here the meaning of state is "what groups are active on this link." The information about the group comes from IGMP (see Chapter 21). Each router running IGMP solicits the hosts on the link to find out the membership status.

When a router receives all these LSPs, it creates *n (n* is the number of groups) topologies, from which *n* shortest path trees are made by using Dijkstra's algorithm. So each router has a routing table that represents as many shortest path trees as there are groups.

The only problem with this protocol is the time and space needed to create and save the many shortest path trees. The solution is to create the trees only when needed. When a router receives a packet with a multicast destination address, it runs the Dijkstra algorithm to calculate the shortest path tree for that group. The result can be cached in case there are additional packets for that destination.

MOSPF   Multicast Open Shortest Path First (MOSPF) protocol is an extension of the OSPF protocol that uses multicast link state routing to create source-based trees.

The protocol requires a new link state update packet to associate the unicast address of a host with the group address or addresses the host is sponsoring. This packet is called the group-membership LSA. In this way, we can include in the tree only the hosts (using their unicast addresses) that belong to a particular group. In other words, we make a tree that contains all the hosts belonging to a group, but we use the unicast address of the host in the calculation. For efficiency, the router calculates the shortest path trees on demand (when it receives the first multicast packet). In addition, the tree can be saved in cache memory for future use by the same source/group pair. MOSPF is a data-driven protocol; the first time an MOSPF router sees a datagram with a given source and group address, the router constructs the Dijkstra shortest path tree.

Multicast Distance Vector: DVMRP

In this section, we briefly discuss multicast distance vector routing and its implementation in the Internet, DVMRP.

Multicast Distance Vector Routing   Unicast distance vector routing is very simple; extending it to support multicast routing is complicated. Multicast routing does not allow a router to send its routing table to its neighbors. The idea is to create a table from scratch by using the information from the unicast distance vector tables.

Multicast distance vector routing uses source-based trees, but the router never actually makes a routing table. When a router receives a multicast packet, it forwards the packet as though it is consulting a routing table. We can say that the shortest path tree is evanescent. After its use (after a packet is forwarded) the table is destroyed.

To accomplish this, the multicast distance vector algorithm uses a process based on four decision-making strategies. Each strategy is built on its predecessor. We explain them one by one and see how each strategy can improve the shortcomings of the previous one.

D   Flooding. Flooding is the first strategy that comes to mind. A router receives a packet and, without even looking at the destination group address, sends it out from every interlace except the one from which it was received. Flooding accomplishes the first goal of multicasting: every network with active members receives the packet. However, so will networks without active members. This is a broadcast, not a multicast. There is another problem: it creates loops. A packet that has left the router may come back again from another interlace or the same interlace and be forwarded again. Some flooding protocols keep a copy of the packet for a while and discard any duplicates to avoid loops. The next strategy, reverse path forwarding, corrects this defect.

Flooding broadcasts packets, but creates loops in the systems.

O   Reverse Path Forwarding (RPF). RPF is a modified flooding strategy. To prevent loops, only one copy is forwarded; the other copies are dropped. In RPF, a router forwards only the copy that has traveled the shortest path from the source to the router. To find this copy, RPF uses the unicast routing table. The router receives a packet and extracts the source address (a unicast address). It consults its unicast routing table as though it wants to send a packet to the source address. The routing table tells the

router the next hop. If the multicast packet has just come from the hop defined in the table, the packet has traveled the shortest path from the source to the router because the shortest path is reciprocal in unicast distance vector routing protocols. If the path from A to B is the shortest, then it is also the shortest from B to A. The router forwards the packet if it has traveled from the shortest path; it discards it otherwise.

This strategy prevents loops because there is always one shortest path from the source to the router. If a packet leaves the router and comes back again, it has not traveled the shortest path. To make the point clear, let us look at Figure 22.40.

Figure 22.40 shows part of a domain and a source. The shortest path tree as calculated by routers RI, R2, and R3 is shown by a thick line. When RI receives a packet from the source through the interface rnl, it consults its routing table and finds that the shortest path from RI to the source is through interface mI. The packet is forwarded. However, if a copy of the packet has arrived through interface m2, it is discarded because m2 does not define the shortest path from RI to the source. The story is the same with R2 and R3. You may wonder what happens if a copy of a packet that arrives at the ml interface of R3, travels through R6, R5, R2, and then enters R3 through interface ml. This interface is the correct interface for R3. Is the copy of the packet forwarded? The answer is that this scenario never happens because when the packet goes from R5 to R2, it will be discarded by R2 and never reaches R3. The upstream routers toward the source always discard a packet that has not gone through the shortest path, thus preventing confusion for the downstream routers.

RPF eliminates the loop in the flooding process.

Figure 22.40   *Reverse path forwarding (RPF)*
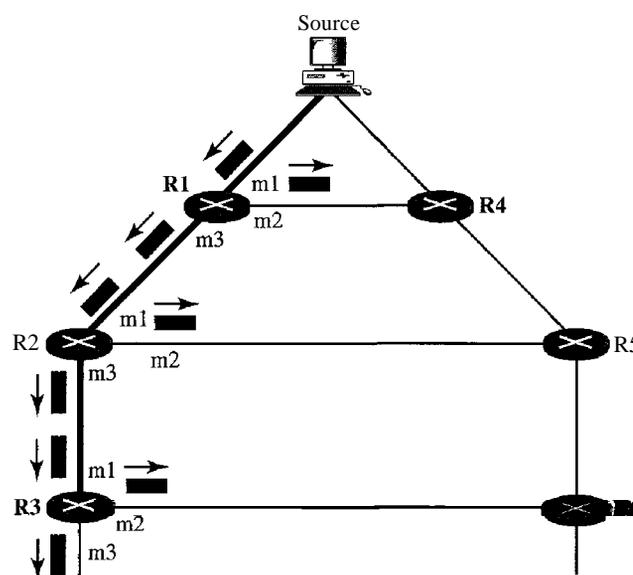
O   Reverse Path Broadcasting (RPB).   RPF guarantees that each network receives a copy of the multicast packet without formation of loops. However, RPF does not

guarantee that each network receives only one copy; a network may receive two or more copies. The reason is that RPF is not based on the destination address (a group address); forwarding is based on the source address. To visualize the problem, let us look at Figure 22.41.

Figure 22.41   *Problem with RPF*
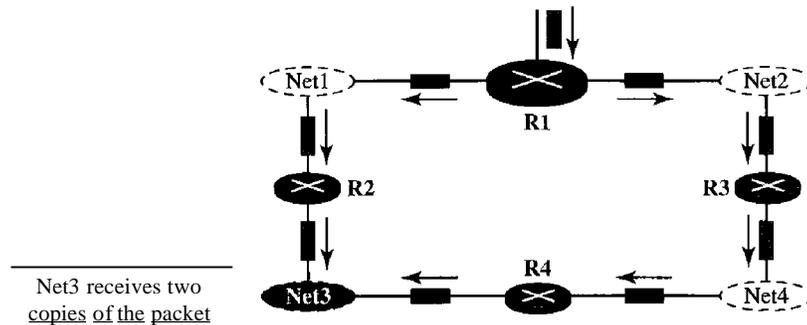
Net3 in this figure receives two copies of the packet even though each router just sends out one copy from each interface. There is duplication because a tree has not been made; instead of a tree we have a graph. Net3 has two parents: routers R2 andR4.

To eliminate duplication, we must define only one parent router for each network. We must have this restriction: A network can receive a multicast packet from a particular source only through a designated parent router.

Now the policy is clear. For each source, the router sends the packet only out of those interfaces for which it is the designated parent. This policy is called reverse path broadcasting (RPB). RPB guarantees that the packet reaches every network and that every network receives only one copy. Figure 22.42 shows the difference between RPF and RPB.

Figure 22.42   *RPF Versus RPB*
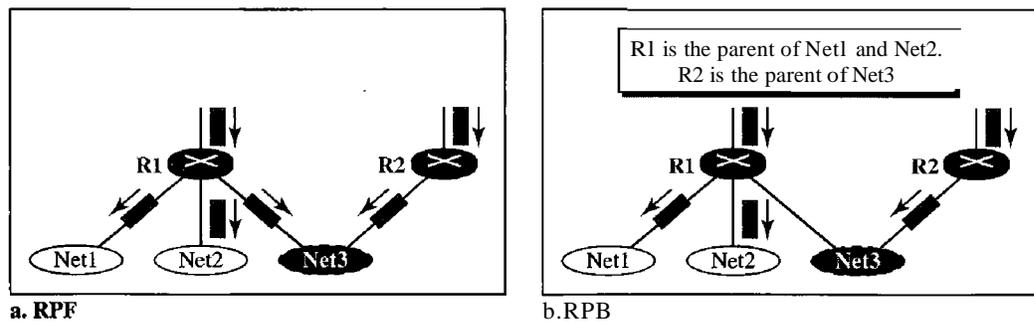
The reader may ask how the designated parent is determined. The designated parent router can be the router with the shortest path to the source. Because routers periodically

send updating packets to each other (in RIP), they can easily determine which router in the neighborhood has the shortest path to the source (when interpreting the source as the destination), If more than one router qualifies, the router with the smallest IP address is selected.

RPB creates a shortest path broadcast tree from the source to each destination.
It guarantees that each destination receives one and only one copy of the packet.

O Reverse Path Multicasting (RPM). As you may have noticed, RPB does not multicast the packet, it broadcasts it. This is not efficient. To increase efficiency, the multicast packet must reach only those networks that have active members for that particular group. This is called reverse path multicasting (RPM). To convert broadcasting to multicasting, the protocol uses two procedures, pruning and grafting. Figure 22.43 shows the idea of pruning and grafting.

Figure 22.43 *RPF, RPB, and RPM*

a.RPF

b.RPB

c. RPM (after pruning)

d. RPM (after grafting)
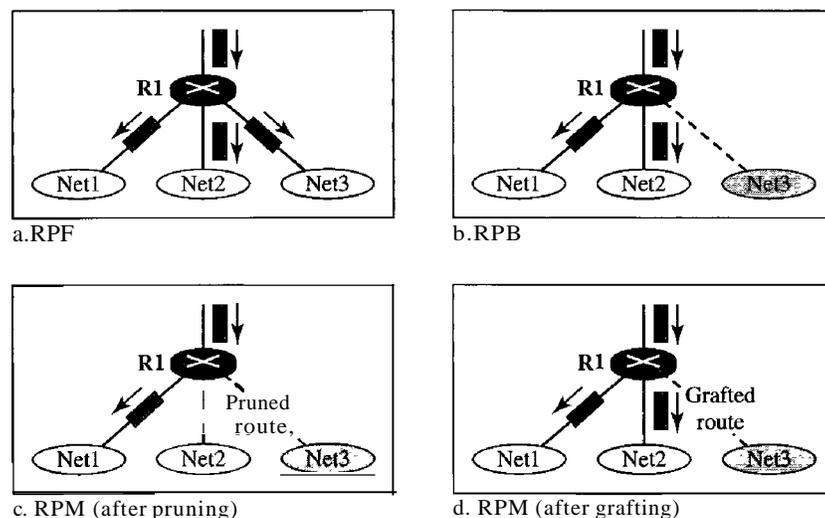
The designated parent router of each network is responsible for holding the membership information. This is done through the IGMP protocol described in Chapter 21. The process starts when a router connected to a network finds that there is no interest in a multicast packet. The router sends a prune message to the upstream router so that it can exclude the corresponding interface. That is, the upstream router can stop sending multicast messages for this group through that interface. Now if this router receives prune messages from all downstream routers, it, in turn, sends a prune message to its upstream router.

What if a leaf router (a router at the bottom of the tree) has sent a prune message but suddenly realizes, through IGMP, that one of its networks is again interested in receiving the multicast packet? It can send a graft message. The graft message forces the upstream router to resume sending the multicast messages.

RPM adds pruning and grafting to RPB to create a multicast shortest
path tree that supports dynamic membership changes.

DVMRP   The Distance Vector Multicast Routing Protocol (DVMRP) is an imple-
mentation of multicast distance vector routing. It is a source-based routing protocol,
based on RIP.

CRT

The Core-Based Tree (CBT) protocol is a group-shared protocol that uses a core as
the root of the tree. The autonomous system is divided into regions, and a core (center
router or rendezvous router) is chosen for each region.

Formation of the Tree   After the rendezvous point is selected, every router is infonned
of the unicast address of the selected router. Each router then sends a unicast join message
(similar to a grafting message) to show that it wants to join the group. This message passes
through all routers that are located between the sender and the rendezvous router. Each
intermediate router extracts the necessary infonnation from the message, such as the
unicast address of the sender and the interface through which the packet has arrived, and
forwards the message to the next router in the path. When the rendezvous router has
received all join messages from every member of the group, the tree is formed. Now
every router knows its upstream router (the router that leads to the root) and the down-
stream router (the router that leads to the leaf).

   If a router wants to leave the group, it sends a leave message to its upstream router.
The upstream router removes the link to that router from the tree and forwards the mes-
sage to its upstream router, and so on. Figure 22.44 shows a group-shared tree with its
rendezvous router.

Figure 22.44   *Group-shared tree with rendezvous router*
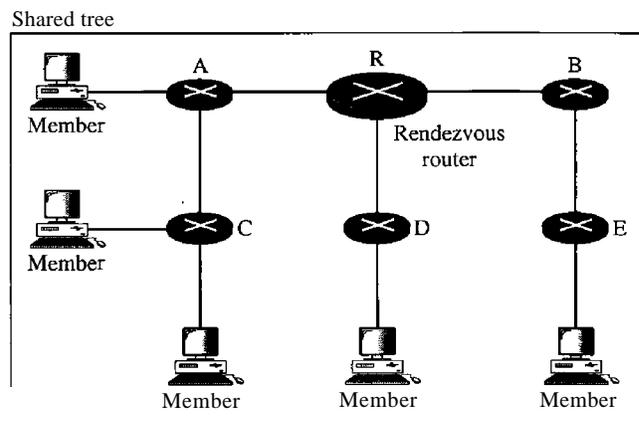
   The reader may have noticed two differences between DVMRP and MOSPF, on
one hand, and CBT, on the other. First, the tree for the first two is made from the root
up; the tree for CBT is fonned from the leaves down. Second, in DVMRP, the tree is

first made (broadcasting) and then pruned; in CBT, there is no tree at the beginning; the joining (grafting) gradually makes the tree.

Sending Multicast Packets After formation of the tree, any source (belonging to the group or not) can send a multicast packet to all members of the group. It simply sends the packet to the rendezvous router, using the unicast address of the rendezvous router; the rendezvous router distributes the packet to all members of the group. Figure 22.45 shows how a host can send a multicast packet to all members of the group. Note that the source host can be any of the hosts inside the shared tree or any host outside the shared tree. In Figure 22.45 we show one located outside the shared tree.

Figure 22.45 *Sending a multicast packet to the rendezvous router*
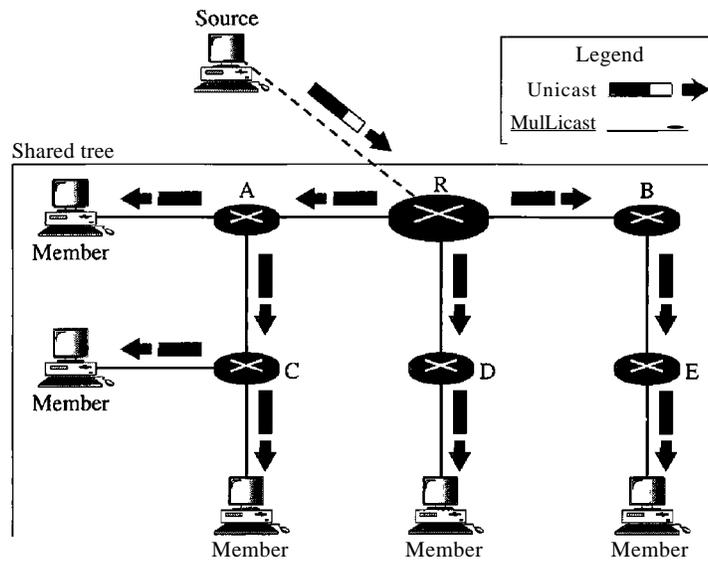
Selecting the Rendezvous Router This approach is simple except for one point. How do we select a rendezvous router to optimize the process and multicasting as well? Several methods have been implemented. However, this topic is beyond the scope of this book, and we leave it to more advanced books.

In summary, the Core-Based Tree (CBT) is a group-shared tree, center-based protocol using one tree per group. One of the routers in the tree is called the core. A packet is sent from the source to members of the group following this procedure:

1. The source, which mayor may not be part of the tree, encapsulates the multicast packet inside a unicast packet with the unicast destination address of the core and sends it to the core. This part of delivery is done using a unicast address; the only recipient is the core router.

2. The core decapsulates the unicast packet and forwards it to all interested intetfaces.

3. Each router that receives the multicast packet, in tum, forwards it to all interested interfaces.

In CRT, the source sends the multicast packet (encapsulated in a unicast packet) to the core router. The core router decapsulates the packet and forwards it to all interested interfaces.

PIM

Protocol Independent Multicast (PIM) is the name given to two independent multicast routing protocols: Protocol Independent Multicast, Dense Mode (PIM-DM) and Protocol Independent Multicast, Sparse Mode (PIM-SM). Both protocols are unicast-protocol-dependent, but the similarity ends here. We discuss each separately.

PIM-DM   PIM-DM is used when there is a possibility that each router is involved in multicasting (dense mode). In this environment, the use of a protocol that broadcasts the packet is justified because almost all routers are involved in the process.

PIM-DM is used in a dense multicast environment, such as a LAN.

PIM-DM is a source-based tree routing protocol that uses RPF and pruning and grafting strategies for multicasting. Its operation is like that of DVMRP; however, unlike DVMRP, it does not depend on a specific unicasting protocol. It assumes that the autonomous system is using a unicast protocol and each router has a table that can find the outgoing interface that has an optimal path to a destination. This unicast protocol can be a distance vector protocol (RIP) or link state protocol (OSPF).

PIM-DM uses RPF and pruning and grafting strategies to handle multicasting.
However, it is independent of the underlying unicast protocol.

PIM-SM   PIM-SM is used when there is a slight possibility that each router is involved in multicasting (sparse mode). In this environment, the use of a protocol that broadcasts the packet is not justified; a protocol such as CBT that uses a group-shared tree is more appropriate.

PIM-SM is used in a sparse multicast environment such as a WAN.

PIM-SM is a group-shared tree routing protocol that has a rendezvous point (RP) as the source of the tree. Its operation is like CBT; however, it is simpler because it does not require acknowledgment from a join message. In addition, it creates a backup set of RPs for each region to cover RP failures.

One of the characteristics of PIM-SM is that it can switch from a group-shared tree strategy to a source-based tree strategy when necessary. This can happen if there is a dense area of activity far from the RP. That area can be more efficiently handled with a source-based tree strategy instead of a group-shared tree strategy.


PIM-SM is similar to CRT but uses a simpler procedure.

MBONE

Multimedia and real-time communication have increased the need for multicasting in the Internet. However, only a small fraction of Internet routers are multicast routers. In

other words, a multicast router may not find another multicast router in the neighbor-hood to forward the multicast packet. Although this problem may be solved in the next few years by adding more and more multicast routers, there is another solution to this problem. The solution is tunneling. The multicast routers are seen as a group of routers on top of unicast routers. The multicast routers may not be connected directly, but they are connected logically. Figure 22.46 shows the idea. In Figure 22.46, only the routers enclosed in the shaded circles are capable of multicasting. Without tunneling, these routers are isolated islands. To enable multicasting, we make a multicast backbone (MBONE) out of these isolated routers by using the concept of tunneling.

Figure 22.46   *Logical tunneling*
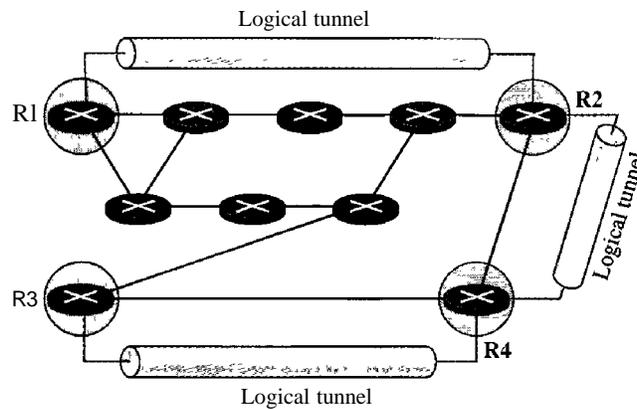
A logical tunnel is established by encapsulating the multicast packet inside a uni-cast packet. The multicast packet becomes the payload (data) of the unicast packet. The intermediate (nonmulticast) routers forward the packet as unicast routers and deliver the packet from one island to another. It's as if the unicast routers do not exist and the two multicast routers are neighbors. Figure 22.47 shows the concept. So far the only protocol that supports MBONE and tunneling is DVMRP.

Figure 22.47   *MBONE*
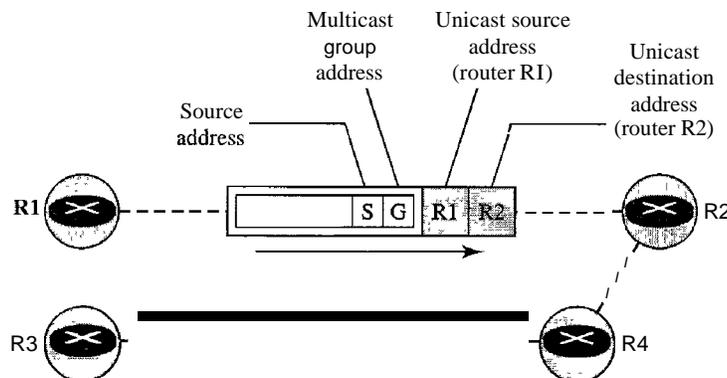
22.5   RECOMMENDED READING

For more details about subjects discussed in this chapter, we recommend the following books and sites. The items in brackets [...J refer to the reference list at the end of the text.

Books

Delivery and forwarding are discussed in Chapter 6 of [For06]. Unicast routing protocols are discussed in Chapter 14 of [For06]. Multicasting and multicast routing are discussed in Chapter 15 of [For06]. For a complete discussion of multicasting see [WZOl]. For routing protocols see [HuiOO]. OSPF is discussed in [Moy98].

Sites

O   www.ietf.org/rfc.html   Information about RFCs

RFCs

A discussion of RIP can be found in the following RFCs:

  1131,1245,1246,1247,1370,1583,1584,1585, 1586,1587, 1722,1723,2082,2453

A discussion of OSPF can be found in the following RFCs:

  1131,1245,1246,1247,1370,1583,1584,1585, 1586, 1587,2178,2328,2329,2370

A discussion of BGP can be found in the following RFCs:

  1092,1105,1163,1265,1266,1267,1364,1392,1403, 1565,1654,1655,1665,1771,1772, 1745,1774,2283

22.6   KEY TERMS

address aggregation	Dijkstra's algorithm
area	direct delivery
area border routers	distance learning
area identification	Distance Vector Multicast Routing Protocol (DVMRP)
autonomous system (AS)	distance vector routing
backbone router	distributed database
Border Gateway Protocol (BGP)	dynamic routing method
broadcasting	dynamic routing table
Core-Based Tree (CBT) protocol	flooding
data-driven	forwarding
default method	graft message
delivery	group-shared tree
designated parent router	

hierarchical routing

hop count

host-specific method

ifconfig

immediate neighbors

indirect delivery

interdomain routing

intradomain routing

least-cost tree

link state routing

logical tunnel

longest mask matching

metric

multicast backbone (MBONE)

Multicast Open Shortest Path First (MOSPF)

multicast router

multicast routing

multicasting

multiple unicasting

netstat

network-specific method

next-hop address

next-hop method

Open Shortest Path First (OSPF)

optional attribute

OSPF protocol

path vector routing

point-to-point link

poison reverse

policy routing

Protocol Independent Multicast (PIM)

Protocol Independent Multicast, Dense Mode (PIM-DM)

Protocol Independent Multicast, Sparse Mode (PIM-SM)

prune message

rendezvous router

rendezvous-point tree

reverse path broadcasting (RPB)

reverse path forwarding (RPF)

reverse path multicasting (RPM)

route method

routing

Routing Information Protocol (RIP)

routing protocols

shortest path tree

slow convergence

source-based tree

speaker node

split horizon

static routing table

stub link

switching fabric

teleconferencing

transient link

triggered update

tunneling

unicasting

update message

virtual link

well-known attribute

22.7   SUMMARY

O   The delivery of a packet is called direct if the deliverer (host or router) and the destination are on the same network; the delivery of a packet is called indirect if the deliverer (host or router) and the destination are on different networks.

O   In the next-hop method, instead of a complete list of the stops the packet must make, only the address of the next hop is listed in the routing table; in the network-specific method, all hosts on a network share one entry in the routing table.

❑ In the host-specific method, the full IP address of a host is given in the routing table.

❑ In the default method, a router is assigned to receive all packets with no match in the routing table.

❑ The routing table for classless addressing needs at least four columns.

❑ Address aggregation simplifies the forwarding process in classless addressing.

O Longest mask matching is required in classless addressing.

O Classless addressing requires hierarchical and geographical routing to prevent immense routing tables.

❑ A static routing table's entries are updated manually by an administrator; a dynamic routing table's entries are updated automatically by a routing protocol.

❑ A metric is the cost assigned for passage of a packet through a network.

❑ An autonomous system (AS) is a group of networks and routers under the authority of a single administration.

❑ RIP is based on distance vector routing, in which each router shares, at regular intervals, its knowledge about the entire AS with its neighbors.

❑ Two shortcomings associated with the RIP protocol are slow convergence and instability. Procedures to remedy RIP instability include triggered update, split horizons, and poison reverse.

❑ OSPF divides an AS into areas, defined as collections of networks, hosts, and routers.

O OSPF is based on link state routing, in which each router sends the state of its neighborhood to every other router in the area. A packet is sent only if there is a change in the neighborhood.

❑ OSPF routing tables are calculated by using Dijkstra's algorithm.

❑ BGP is an interautonomous system routing protocol used to update routing tables.

❑ BGP is based on a routing protocol called path vector routing. In this protocol, the ASs through which a packet must pass are explicitly listed.

O In a source-based tree approach to multicast routing, the source/group combination determines the tree; in a group-shared tree approach to multicast routing, the group determines the tree.

❑ MOSPF is a multicast routing protocol that uses multicast link state routing to create a source-based least-cost tree.

❑ In reverse path forwarding (RPF), the router forwards only the packets that have traveled the shortest path from the source to the router.

❑ Reverse path broadcasting (RPB) creates a shortest path broadcast tree from the source to each destination. It guarantees that each destination receives one and only one copy of the packet.

❑ Reverse path multicasting (RPM) adds pruning and grafting to RPB to create a multicast shortest path tree that supports dynamic membership changes.

❑ DVMRP is a multicast routing protocol that uses the distance routing protocol to create a source-based tree.

❑ The Core-Based Tree (CBT) protocol is a multicast routing protocol that uses a router as the root of the tree.

D PIM-DM is a source-based tree routing protocol that uses RPF and pruning and grafting strategies to handle multicasting.

O PIM-SM is a group-shared tree routing protocol that is similar to CBT and uses a rendezvous router as the source of the tree.

D For multicasting between two noncontiguous multicast routers, we make a multicast backbone (MBONE) to enable tunneling.

22.8 PRACTICE SET

Review Questions

1. What is the difference between a direct and an indirect delivery?
2. List three forwarding techniques discussed in this chapter and give a brief description of each.
3. Contrast two different routing tables discussed in this chapter.
4. What is the purpose of RIP?
5. What are the functions of a RIP message?
6. Why is the expiration timer value 6 times that of the periodic timer value?
7. How does the hop count limit alleviate RIP's problems?
8. List RIP shortcomings and their corresponding fixes.
9. What is the basis of classification for the four types of links defined by OSPF?
10. Why do OSPF messages propagate faster than RIP messages?
11. What is the purpose of BGP?
12. Give a brief description of two groups of multicast routing protocols discussed in this chapter.

Exercises

13. Show a routing table for a host that is totally isolated.
14. Show a routing table for a host that is connected to a LAN without being connected to the Internet.
15. Find the topology of the network if Table 22.3 is the routing table for router R1.

Table 22.3   *Routing table for Exercise 15*

Mask	Network Address	Next-Hop Address	Interface
/27	202.14.17.224	-	m1
/18	145.23.192.0	-	m0
Default	Default	130.56.12.4	m2

16. Can router R1 in Figure 22.8 receive a packet with destination address 140.24.7.194? Explain your answer.

17. Can router R1 in Figure 22.8 receive a packet with destination address 140.24.7.42? Explain your answer.

18. Show the routing table for the regional ISP in Figure 22.9.

19. Show the routing table for local ISP 1 in Figure 22.9.

20. Show the routing table for local ISP 2 in Figure 22.9.

21. Show the routing table for local ISP 3 in Figure 22.9.

22. Show the routing table for small ISP 1 in Figure 22.9.

23. Contrast and compare distance vector routing with link state routing.

24. A router has the following RIP routing table:

Net!	4	B
Net2	2	C
Net3	!	F
Net4	5	G

What would be the contents of the table if the router received the following RIP message from router C?

Net!	2
Net2	!
Net3	3
Net4	7

25. How many bytes are empty in a RIP message that advertises *N* networks?

26. A router has the following RIP routing table:

Net!	4	B
Net2	2	C
Net3	1	F
Net4	5	G

Show the response message sent by this router.

27. Show the autonomous system with the following specifications:

a. There are eight networks (Nl to N8).

b. There are eight routers (Rl to R8).

c. Nl, N2, N3, N4, NS, and N6 are Ethernet LANs.

d. N7 and N8 are point-to-point WANs.

e. Rl connects N1 and N2.

f. R2 connects N1 and N7.

g. R3 connects N2 and N8.

h. R4 connects N7 and N6.

i. RS connects N6 and N3.

j. R6 connects N6 and N4.

k. R7 connects N6 and NS.

l. R8 connects N8 and N5.

28. Draw the graphical representation of the autonomous system of Exercise 27 as seen by OSPF.

29. Which of the networks in Exercise 27 is a transient network? Which is a stub network?

30. A router using DVMRP receives a packet with source address 10.14.17.2 from interface 2. If the router forwards the packet, what are the contents of the entry related to this address in the unicast routing table?

31. Does RPF actually create a shortest path tree? Explain.

32. Does RPB actually create a shortest path tree? Explain. What are the leaves of the tree?

33. Does RPM actually create a shortest path tree? Explain. What are the leaves of the tree?

Research Activities

34. If you have access to UNIX (or LINUX), use *netstat* and *ifconfig* to find the routing table for the server to which you are connected.

35. Find out how your ISP uses address aggregation and longest mask match principles.

36. Find out whether your IP address is part of the geographical address allocation.

37. If you are using a router, find the number and names of the columns in the routing table.