CHAPTER 13

Routing Protocols (RIP, OSPF, and BGP)

An internet is a combination of networks connected by routers. When a datagram goes from a source to a destination, it will probably pass through many routers until it reaches the router attached to the destination network.

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?

A **metric** is a cost assigned for passing through a network. The total metric of a particular route is equal to the sum of the metrics of networks that comprise the route. A router chooses the route with the shortest (smallest) metric.

The metric assigned to each network depends on the type of protocol. Some simple protocols, like the Routing Information Protocol (RIP), treat each network as equals. The cost of passing through each 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 throughput 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 line. OSPF allows each router to have several routing tables based on the required type of service.

Other protocols define the metric totally differently. 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.

Whatever the metric, a router should have a routing table to consult when a packet is ready to be forwarded. The routing table specifies the optimum path for the packet. However, the table can be either static or dynamic. A *static table* is one that is not changed frequently. 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 route is down, and they need to be updated whenever a better route has been created.

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.

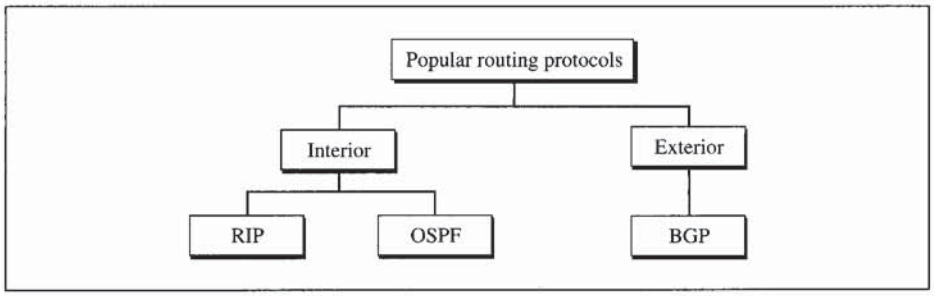
In this chapter we discuss unicast routing protocols. Multicast routing protocols will be discussed in the next chapter.

13.1 INTERIOR AND EXTERIOR ROUTING

Today, an internet can be so large that one routing protocol cannot handle the task of updating 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 *interior routing*. Routing between autonomous systems is referred to as *exterior routing*. Each autonomous system can choose an interior routing protocol to handle routing inside the autonomous system. However, only one exterior routing protocol is usually chosen to handle routing between autonomous systems.

Several interior and exterior routing protocols are in use. In this chapter, we cover only the most popular ones. We discuss two interior routing protocols, RIP and OSPF, and one exterior routing protocol, BGP (see Figure 13.1).

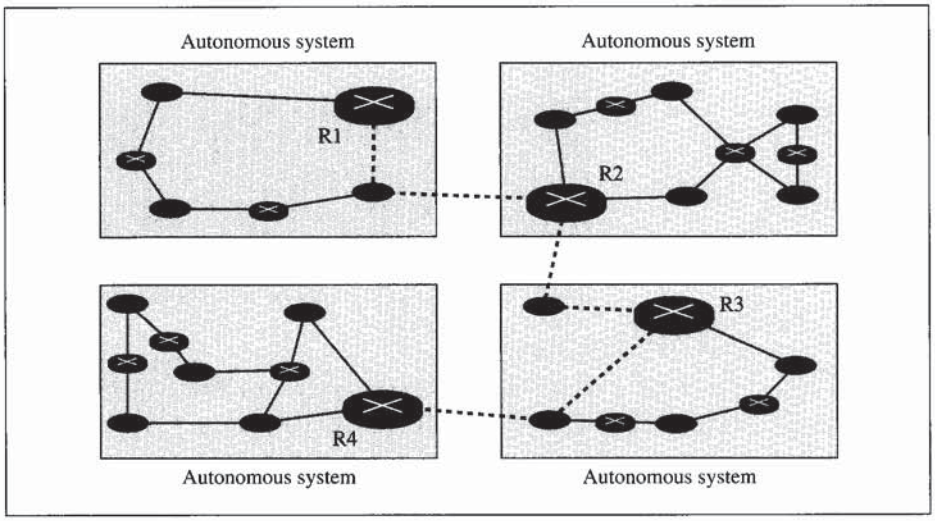
Figure 13.1 *Popular routing protocols*



RIP and OSPF can be used to update routing tables inside an autonomous system. BGP can be used to update routing tables for routers that join the autonomous systems together.

In Figure 13.2, routers R1, R2, R3, and R4 use an interior and an exterior routing protocol. The other routers use only interior routing protocols. The solid thin lines show the communication between routers that use interior routing protocols. The broken thick lines show the communication between the routers that use an exterior routing protocol.

Figure 13.2 *Autonomous systems*



13.2 RIP

The routing information protocol is an interior routing protocol used inside an autonomous system. It is a very simple protocol based on *distance vector routing,* which uses the Bellman-Ford algorithm for calculating the routing tables. In this section, we first study the principle of distance vector routing, as it is applied to RIP, and then discuss the RIP protocol itself.

Distance Vector Routing

In **distance vector routing,** each router periodically shares its knowledge about the entire internet with its neighbors. The three keys to understanding how this algorithm works are as follows:

1. **Sharing knowledge about the entire autonomous system.** Each router shares its knowledge about the entire autonomous system with its neighbors. At the outset, a router's knowledge may be sparse. How much it knows, however, is unimportant; it sends whatever it has.
2. **Sharing only with neighbors.** Each router sends its knowledge only to neighbors. It sends whatever knowledge it has through all of its interfaces.
3. **Sharing at regular intervals.** Each router sends its knowledge to its neighbors at fixed intervals, for example, every 30 s.

Routing Table

Every router keeps a routing table that has one entry for each destination network of which the router is aware. The entry consists of the destination network address, the shortest distance to reach the destination in hop count, and the next hop (next router) to which the packet should be delivered to reach its final destination. The hop count is the number of networks a packet encounters to reach its final destination.

The table may contain other information such as the subnet mask (or prefix), or the time this entry was last updated. Table 13.1 shows an example of a routing table.

Table 13.1 *A distance vector routing table*

Destination	*Hop Count*	*Next Hop*	*Other Information*
163.5.0.0	7	172.6.23.4	
197.5.13.0	5	176.3.6.17	
189.45.0.0	4	200.5.1.6	
115.0.0.0	6	131.4.7.19	

RIP Updating Algorithm

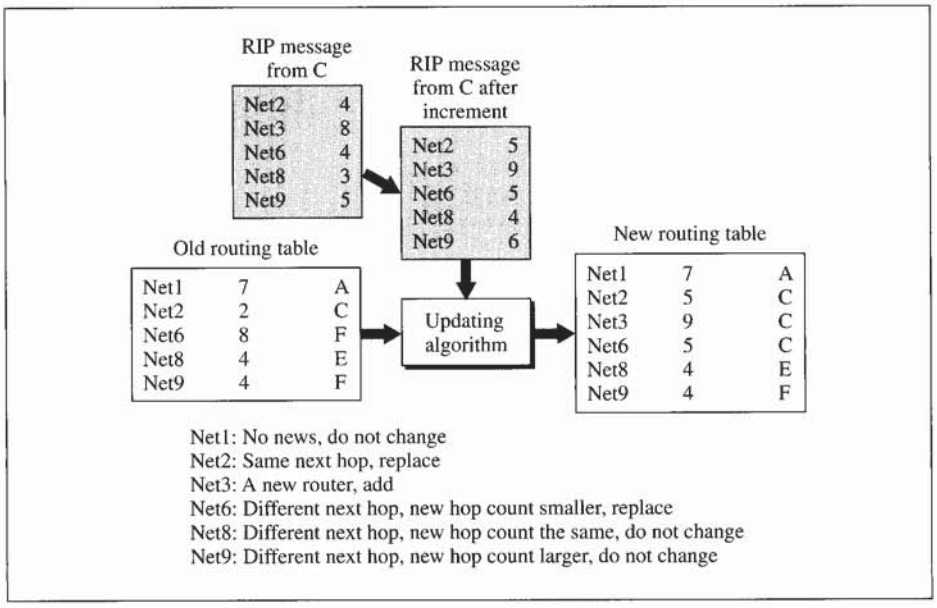
The routing table is updated upon receipt of a RIP response message. The following shows the updating algorithm used by RIP.

RIP Updating Algorithm

```
Receive: a response RIP message
1. Add one hop to the hop count for each advertised destination.
2. Repeat the following steps for each advertised destination:
    1. If (destination not in the routing table)
        1. Add the advertised information to the table.
    2. Else
        1. If (next-hop field is the same)
            1. Replace entry in the table with the advertised one.
        2. Else
            1. If (advertised hop count smaller than one in the table)
                1. Replace entry in the routing table.
3. Return.
```

In Figure 13.3 a router receives a RIP message from router C. The message lists destination networks and their corresponding hop counts. The first step according to the updating algorithm is to increase the hop count by one. Next, this updated RIP packet and the old routing table are compared. The result is a routing table with an up-to-date hop count for each destination. For Net1 there is no new information, so the Net1 entry remains the same.

Figure 13.3 *Example of updating a routing table*



For Net2, information in the table and in the message identify the same next hop (router C). Although the value of the hop count in the table (2) is less than the one in the message (5), the algorithm selects the one received in the message because the original value has come from router C. This value is now invalid because router C is advertising a new value.

Net3 is added as a new destination. For Net6, the RIP packet contains a lower hop count and this shows up on the new routing table. Both Net8 and Net9 retain their original values since the corresponding hop counts in the message are not an improvement.

Initializing the Routing Table

When a router is added to a network, it initializes a routing table for itself using its configuration file. The table contains only the directly attached networks and the hop counts, which are initialized to 1. The next-hop field, which identifies the next router, is empty. Figure 13.4 shows the initial routing tables in a small autonomous system.

Updating the Routing Table

Each routing table is updated upon receipt of RIP messages using the RIP updating algorithm shown above. Figure 13.5 shows our previous autonomous system with final routing tables.

Figure 13.4 *Initial routing tables in a small autonomous system*

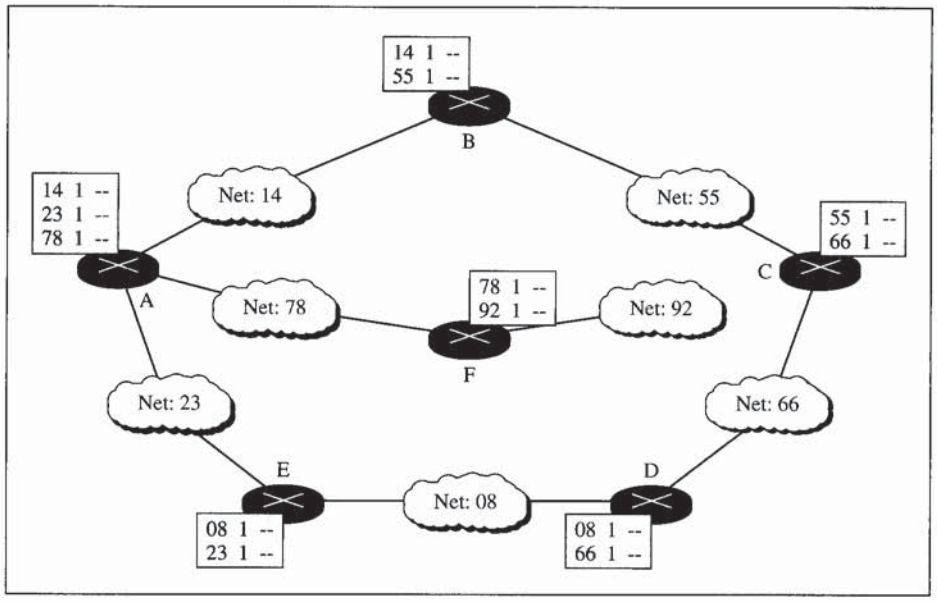
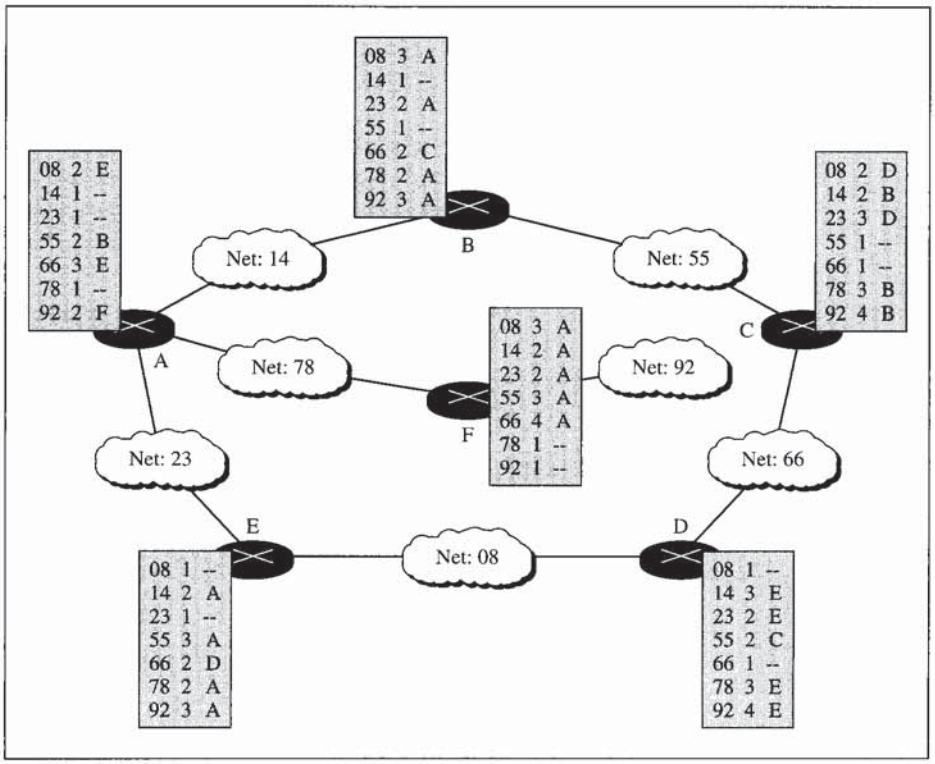


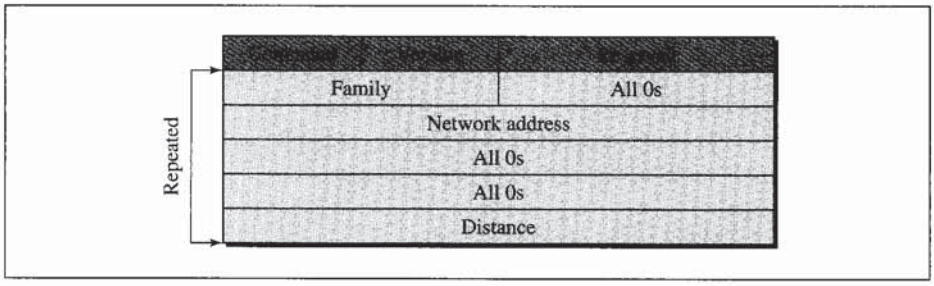
Figure 13.5 *Final routing tables for the previous figure*



RIP Message Format

The format of the RIP message is shown in Figure 13.6.

Figure 13.6 *RIP message format*



- **Command.** This 8-bit field specifies the type of message: request (1) or response (2).
- **Version.** This 8-bit field defines the version. In this book we use version 1, but at the end of this section, we give some new features of version 2.
- **Family.** This 16-bit field defines the family of the protocol used. For TCP/IP the value is 2.
- **Address.** The address field defines the address of the destination network. RIP has allocated 14 bytes for this field to be applicable to any protocol. However, IP currently uses only 4 bytes. The rest of the address is filled with 0s.
- **Distance.** This 32-bit field defines the hop count from the advertising router to the destination network.

Note that part of the message is repeated for each destination network. We refer to this as an *entry*.

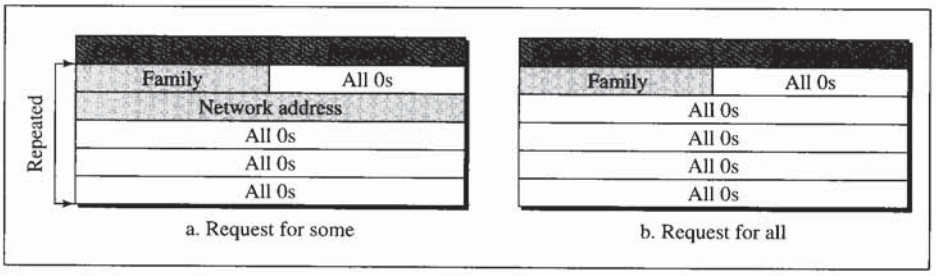
Requests and Responses

RIP uses two types of messages: request and response.

Request

A request message is sent by a router that has just come up or by a router that has some time-out entries. A request can ask about specific entries or all entries (see Figure 13.7).

Figure 13.7 *Request messages*



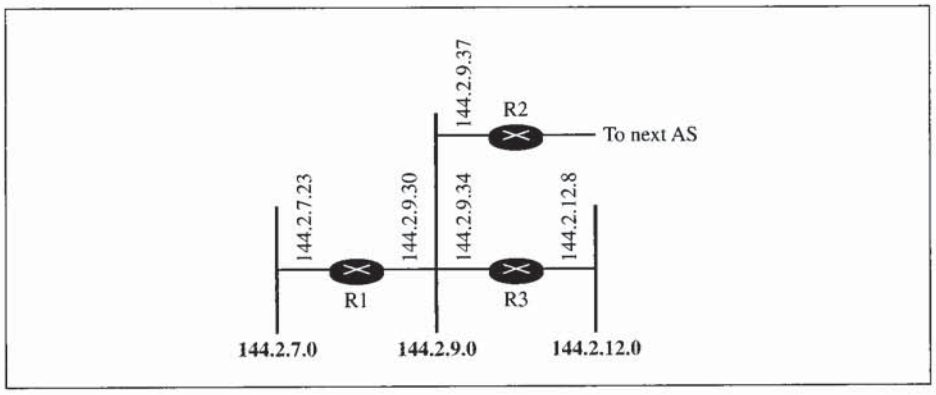
Response

A response can be either solicited or unsolicited. A *solicited response* is sent only in answer to a request. It contains information about the destination specified in the corresponding request. An *unsolicited response,* on the other hand, is sent periodically, every 30 s, and contains information about the entire routing table. This periodic response is sometimes called update packet. Figure 13.6 shows the response message format.

Example 1

What is the periodic response sent by router R1 in Figure 13.8? Assume R1 knows about the whole autonomous system.

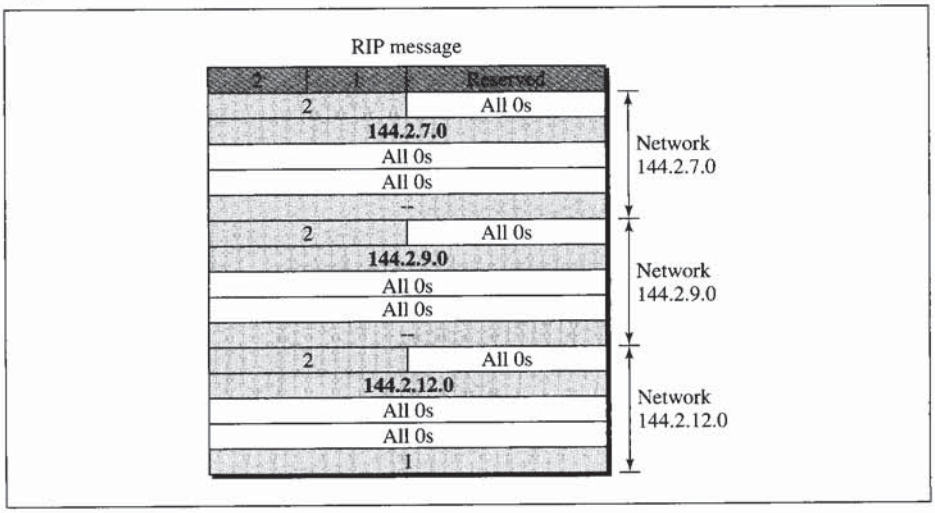
Figure 13.8 *Example 1*



Solution

R1 can advertise three networks 144.2.7.0, 144.2.9.0, and 144.2.12.0. The periodic response (update packet) is shown in Figure 13.9.

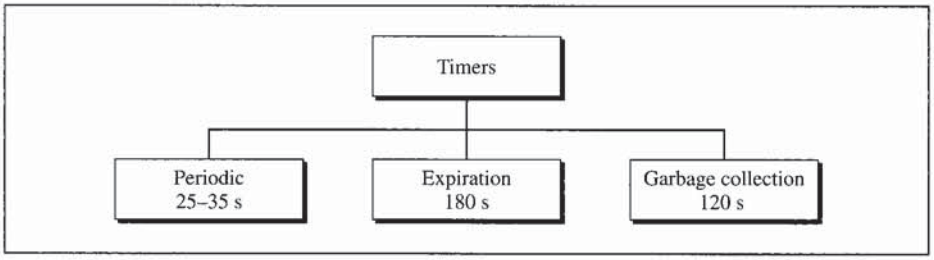
Figure 13.9 *Solution to Example 1*



Timers in RIP

RIP uses three timers to support its operation (see Figure 13.10). The periodic timer controls the sending of messages, the expiration timer governs the validity of a route, and the garbage collection timer advertises the failure of a route.

Figure 13.10 *RIP timers*



Periodic Timer

The **periodic timer** controls the advertising of regular update messages. Although the protocol specifies that this timer must be set to 30 s, the working model uses a random number between 25 and 35 s. This is to prevent any possible synchronization and therefore overload on an internet if routers update simultaneously.

Each router has one periodic timer that is set randomly to a number between 25 and 35. It counts down; when zero is reached, the update message is sent, and the timer is randomly set once again.

If RIP uses an additional timing method to send out updates (see triggered update, to follow), the periodic timer is not affected. The periodic update messages go out on their own schedule without regard to other update messages from other timing systems.

Expiration Timer

The **expiration timer** governs the validity of a route. When a router receives update information for a route, the expiration timer is set to 180 s for that particular route. Every time a new update for the route is received, the timer is reset. In normal situations this occurs every 30 s. However, if there is a problem on an internet and no update is received within the allotted 180 s, the route is considered expired and the hop count of the route is set to 16, which means the destination is unreachable. Every route has its own expiration timer.

Garbage Collection Timer

When the information about a route becomes invalid, the router does not immediately purge that route from its table. Instead, it continues to advertise the route with a metric value of 16. At the same time, a timer called the **garbage collection timer** is set to 120 s for that route. When the count reaches zero, the route is purged from the table. This timer allows neighbors to become aware of the invalidity of a route prior to purging.

Example 2

A routing table has 20 entries. It does not receive information about five routes for 200 s. How many timers are running at this time?

Solution

The timers are listed below:

Periodic timer: 1

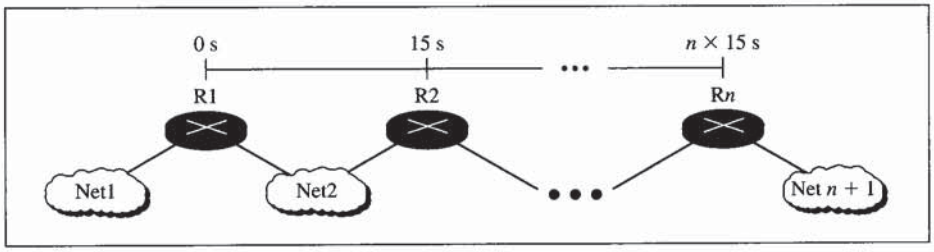
Expiration timer: 20 – 5 = 15

Garbage collection timer: 5

Slow Convergence

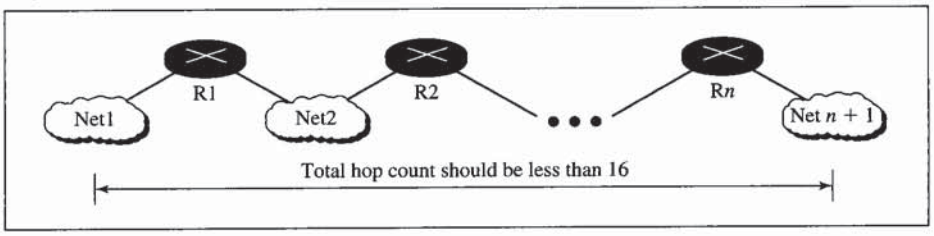
One of the problems with RIP is slow convergence, which means that a change somewhere in the internet propagates very slowly through the rest of the internet. For example, suppose there is a change in network 1 in Figure 13.11. Router R1 updates itself immediately. However, since each router sends its periodic update every 30 s, this means an average of 15 s (range of 0 to 30 s) before a change reaches R2. It also takes another average 15 s before R3 receives the change, and so on. When the information finally reaches router Rn, 15 × n s have passed. If n is 20, then this is 300 s. In this 300 s an ATM network can send more than one billion bits. If this change affects these bits, one billion bits are lost.

Figure 13.11 *Slow convergence*



One method to deal with RIP shortcomings is to limit the hop count to 15. This prevents data packets from wandering around forever, clogging the internet. An autonomous system using RIP is limited to a diameter of 15; the number 16, therefore, is considered infinity and designates an unreachable network (see Figure 13.12).

Figure 13.12 *Hop count*

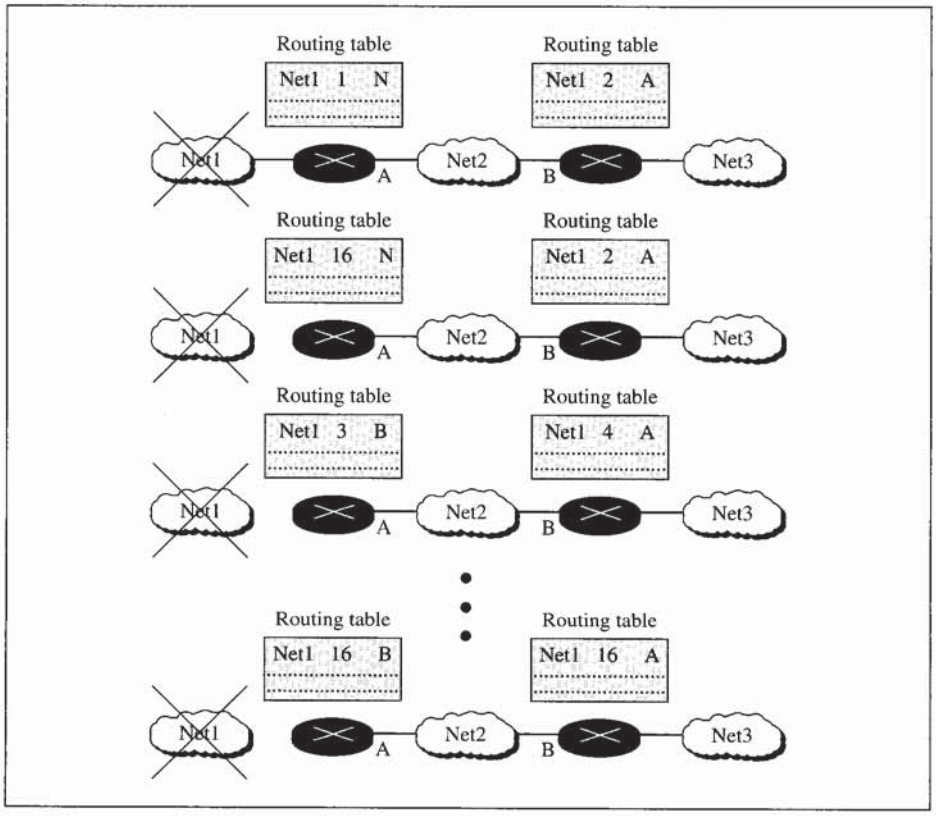


Instability

A much more important problem with RIP is instability, which means that an internet running RIP can become unstable. When this happens a packet could go from one router to another in a loop. Limiting the number of hops to 15 will improve stability but will not eliminate all of the problems.

To understand the problem, assume that the connection to Net1 in Figure 13.13 is nonfunctioning. Router A shows a cost of 1 for this network in its routing table. Router B, which can access Net1 only through router A, shows a cost of 2. When access to Net1 fails, router A immediately responds and changes the Net1 cost column to 16 (infinity). However, it may have to wait up to 30 s before it can send its update with this new information. In the meantime, it could happen that router B sends its own update message to A. Router A now has two entries for Net1: from its own table the cost is 16 and from router B the cost is 2. A is fooled into thinking that there is a backdoor access to Net1 through B. Router A then changes the cost column for Net1 to 3 (2 + 1) and this update gets sent to B. Router B's 2 cost values for Net1 are now 3 (from A) and 2 (from itself). Router B knows that Net1 is accessible only through router A so it disregards its

Figure 13.13 *Instability*



own lower cost and changes its cost to 4 (3 + 1). This back-and-forth updating continues until both routers reach a cost of 16. At this point, the routers realize there is no access to the network Net1.

Some Remedies for Instability

Some remedies have been proposed to improve the stability. However, none of them are 100 percent effective.

Triggered Update

If there are no changes on the network, updates are sent at the usual 30-s intervals. If there is a change, however, the router springs into action immediately by sending out its new table, a process called **triggered update.**

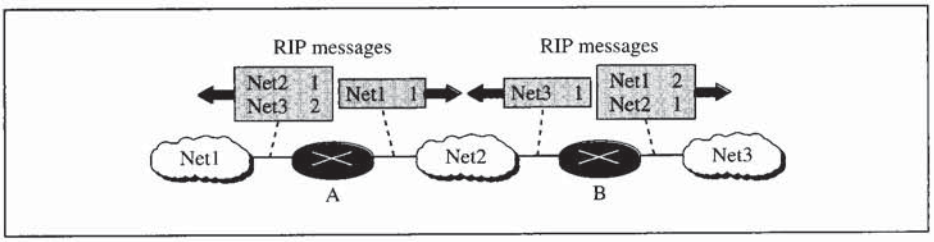
Triggered update can improve stability. Each router that receives an update with a change sends out new information at once, in considerably less time than the 15-s average. For example, in Figure 13.13, when router A realizes that Net1 is unavailable, it changes the cost to 16 in its routing table and then immediately sends this to B. Router B then changes its table, and now both tables show a cost of 16 for Net1. The sending of update messages with incremental changes in cost has been avoided as have any looping problems.

Although triggered update can vastly improve routing, it cannot solve all routing problems. For example, router failure cannot be handled by this method.

Split Horizons

Split horizons, a second method for improving stability, utilizes selectivity in the sending of routing messages; a router must distinguish between different interfaces. If a router has received route updating information from an interface, then this same updated information must not be sent back through this interface. If an interface has passed information to help update a router, this updated information must not be sent back; it is already known and thus is not needed. Figure 13.14 illustrates this concept. In this figure, router B has received information about Net1 and Net2 through its left interface; this information is updated and passed on through the right interface but not to the left. Similarly, information received by router B about Net3 is updated and passed on only through the left interface of B.

Figure 13.14 *Split horizon*



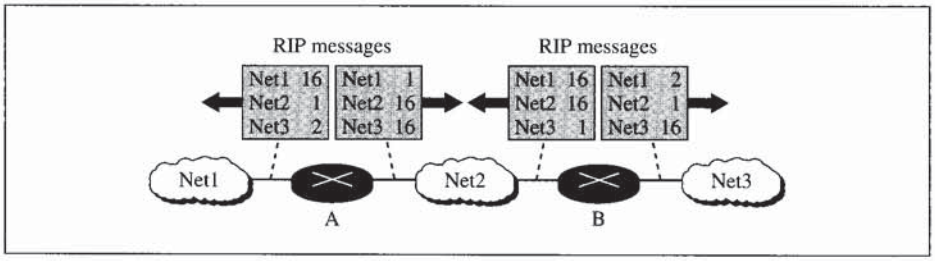
Split horizons can definitely improve stability. Assume that Net1 is inaccessible to router A in Figure 13.13. Router B receives its information about Net1 from A; it does not send information about Net1 to router A. Router A, therefore, has but one entry for the Net1 cost (16), and is not fooled into thinking that there is some back-door access to Net1. Router A sends its routing table to B and both will then end up with a cost of 16 for Net1.

Poison Reverse

Poison reverse is a variation of split horizons. In this method, information received by the router is used to update the routing table and then passed out to all interfaces. However, a table entry that has come through one interface is set to a metric of 16 as it goes out through the same interface.

Figure 13.15 illustrates this concept: Router B has received information about Net1 and Net2 through its left interface, so it sends information out about these networks with a metric of 16. Likewise, information about Net3 comes from the right interface, and the cost of Net3 in the update message going right is 16. Stability is improved using poison reverse. Assume that Net1 is inaccessible to router A in Figure 13.13. Router B receives its information about Net1 from A. In each update, B sends its routing table to A with a cost of 16 for Net1. This has no effect on A if Net1 is accessible because router A will not select B's entry for Net1. However, if Net1 does go down, both cost values are 16 and instability is thereby avoided.

Figure 13.15 *Poison reverse*



RIP Version 2

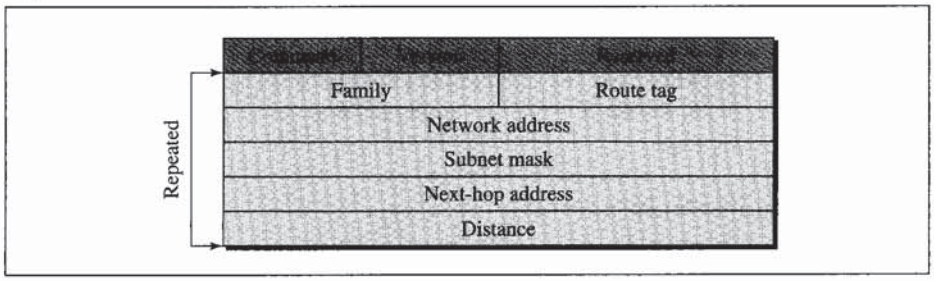
RIP version 2 was designed to overcome some of the shortcomings of version 1. The designers of version 2 have not augmented the length of the message for each entry. They have only replaced those fields in version 1 that were filled with 0s for the TCP/IP protocol with some new fields.

Message Format

Figure 13.16 shows the format of a RIP version 2 message. The new fields of this message are as follows:

- **Route Tag.** This field carries information such as the autonomous system number. It can be used to enable RIP to receive information from an exterior routing protocol.
- **Subnet mask.** This is a 4-byte field that carries the subnet mask (or prefix). This means that RIP2 supports classless addressing and CIDR.

Figure 13.16 *RIP version 2 format*



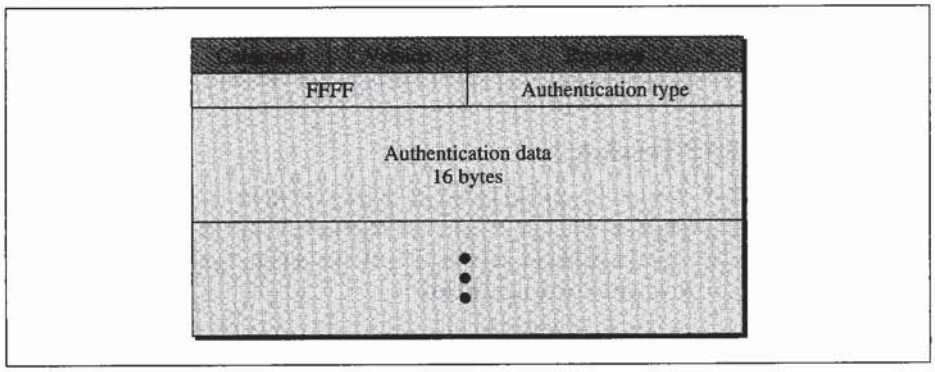
- **Next-hop address.** This field shows the address of the next hop. This is particularly useful if two autonomous systems share a network (a backbone) for example. Then the message can define the router, in the same autonomous system or other autonomous systems, to which the packet should go next.

RIP version 2 supports CIDR.

Authentication

Authentication is added to protect the message against unauthorized advertisement. No new fields are added to the packet; instead, the first entry of the message is set aside for authentication information. To indicate that the entry is authentication information and not routing information, the value of $FFFF_{16}$ is entered in the family field (see Figure 13.17). The second field, the authentication type, defines the method used for authentication, and the third field contains the actual authentication data.

Figure 13.17 *Authentication*



Multicasting

Version 1 of RIP uses broadcasting to send RIP messages to every neighbor. In this way, all the routers on the network receive the packets, as well as the hosts. RIP version 2, on the other hand, uses the multicast address 224.0.0.9 to multicast RIP messages only to RIP routers in the network.

Encapsulation

RIP messages are encapsulated in UDP user datagrams. A RIP message does not include a field that indicates the length of the message. This can be determined from the UDP packet. The well-known port assigned to RIP in UDP is port 520.

RIP uses the services of UDP on well-known port 520.

13.3 OSPF

The open shortest path first (OSPF) protocol is another interior routing protocol that is gaining in popularity. Its domain is also an autonomous system. Special routers called *autonomous system boundary routers* are responsible for dissipating information about other autonomous systems into the current system. To handle routing efficiently and in a timely manner, OSPF divides an autonomous system into areas.

Areas

An **area** is a collection of networks, hosts, and routers all contained within an autonomous system. An autonomous system, in turn, 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 of 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 the secondary areas. This does not mean that the routers within areas cannot be connected with 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.

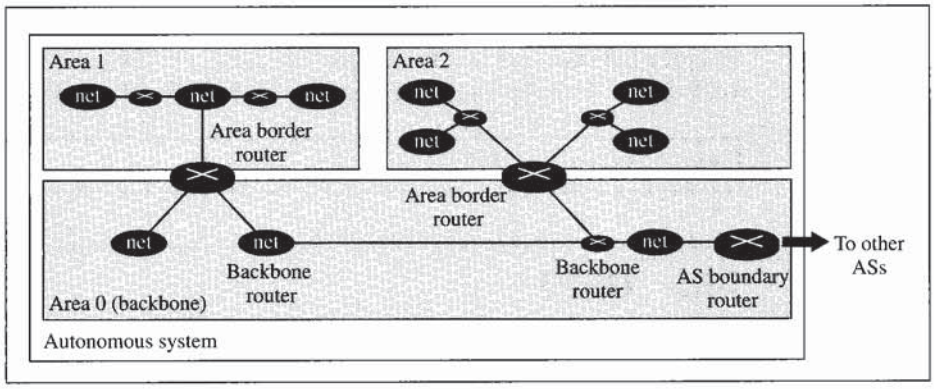
If, due to some problem, the connectivity between a backbone and an area is broken, a *virtual link* between routers must be created by the administration to allow continuity of the functions of the backbone as the primary area.

Each area has an area identification. The area identification of the backbone is zero. Figure 13.18 shows an autonomous system and its areas.

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.

Figure 13.18 *Areas in an autonomous system*



Link State Routing

OSPF uses link state routing to update the routing tables in an area. Before discussing the details of the OSPF protocol, let us discuss **link state routing,** a process by which each router shares its knowledge about its neighborhood with every router in the area. The three keys to understanding how this method works are as follows:

1. **Sharing knowledge about the neighborhood.** Each router sends the *state of its neighborhood* to every other router in the area.
2. **Sharing with every other router.** Each router sends the state of its neighborhood to *every other router in the area.* It does so by **flooding,** a process whereby a router sends its information to all of its neighbors (through all of its output ports). Each neighbor sends the packet to all of its neighbors, and so on. Every router that receives the packet sends copies to each of its neighbors. Eventually, every router (without exception) has received a copy of the same information.
3. **Sharing when there is a change.** Each router shares the state of its neighborhood only when there is a change. This rule contrasts with distance vector routing, where information is sent out at regular intervals regardless of change. This characteristic results in lower internet traffic than that required by distance vector routing.

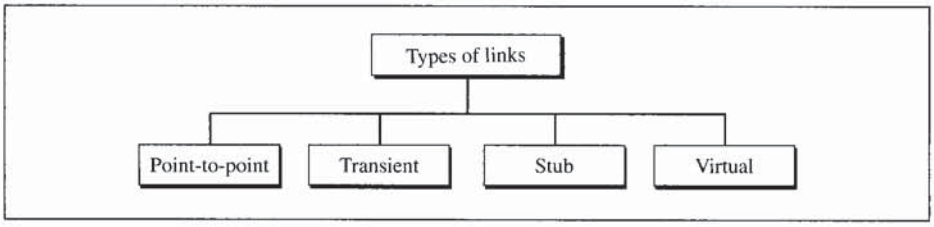
The idea behind link state routing is that each router should have the exact topology of the internet at every moment. In other words, every router should have the whole "picture" of the internet. From this topology, a router can calculate the shortest path between itself and each network. The topology here means a graph consisting of nodes and edges. To represent an internet by a graph, however, we need more definitions.

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 13.19).

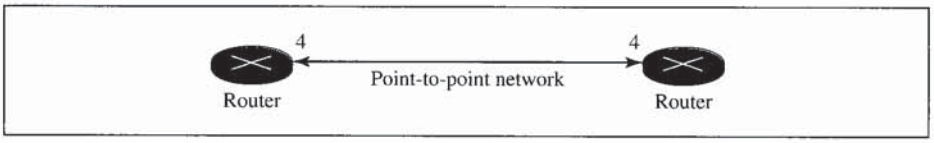
Point-to-Point Link A point-to-point link connects two routers without any other host or router in between. In other words, the purpose of the link (network) is just to connect the two routers. An example of this type of link is two routers connected by a

Figure 13.19 *Types of links*



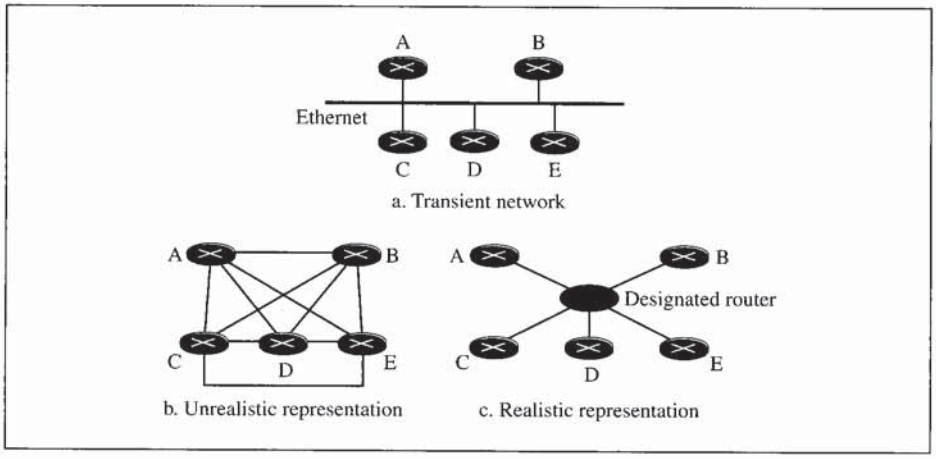
telephone line or a T-line. There is no need to assign a network address to this type of link. Graphically, the routers are represented by nodes, and the link is represented by a bidirectional edge connecting the nodes. The metrics, which are usually the same, are shown at the two ends, one for each direction. In other words, each router has only one neighbor at the other side of the link (see Figure 13.20).

Figure 13.20 *Point-to-point link*



Transient 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 13.21a. 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 13.21b.

Figure 13.21 *Transient link*



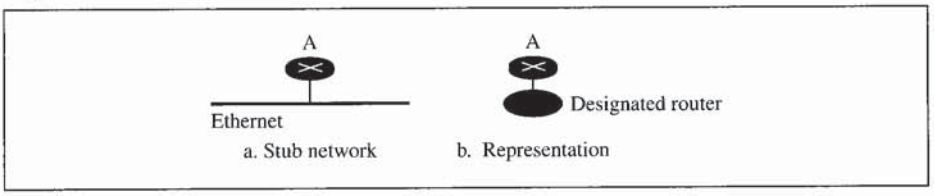
This is neither efficient nor realistic. It is not efficient because each router needs to advertise the neighborhood of four other routers, for a total of 20 advertisements. It is not realistic, because there is no single network (link) between each pair of routers; there is only one network that serves 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 13.21c 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.

Stub Link 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 13.22).

Figure 13.22 *Stub link*



Virtual Link 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 a small internet using link state routing and see how we can represent it graphically. Figure 13.23 shows a small internet 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 a number to a point-to-point network.

To show the above internet graphically, we use square nodes for the routers and ovals for the networks (represented by designated routers); see Figure 13.24. Note that we have three stub networks.

Figure 13.23 *Example of an internet*

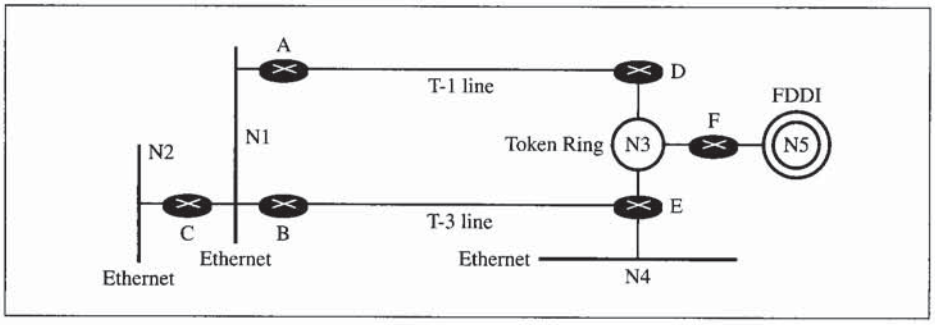
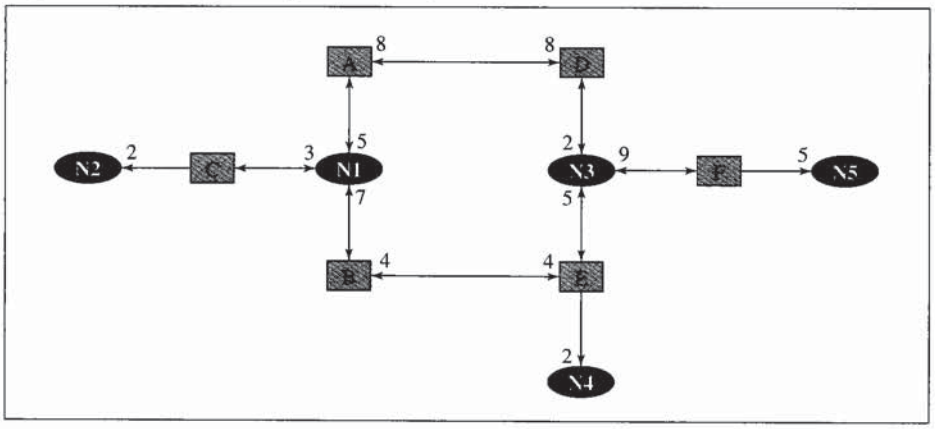


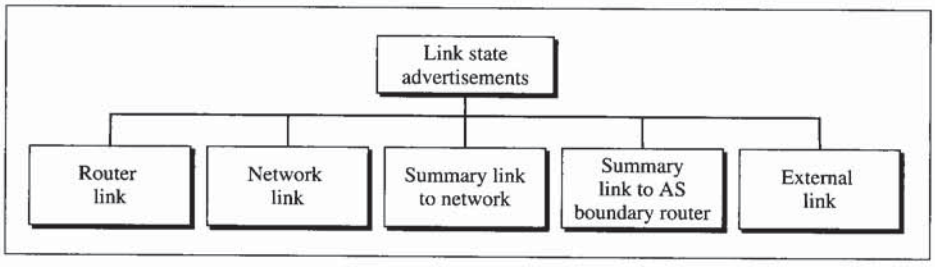
Figure 13.24 *Graphical representation of an internet*



Link State Advertisements

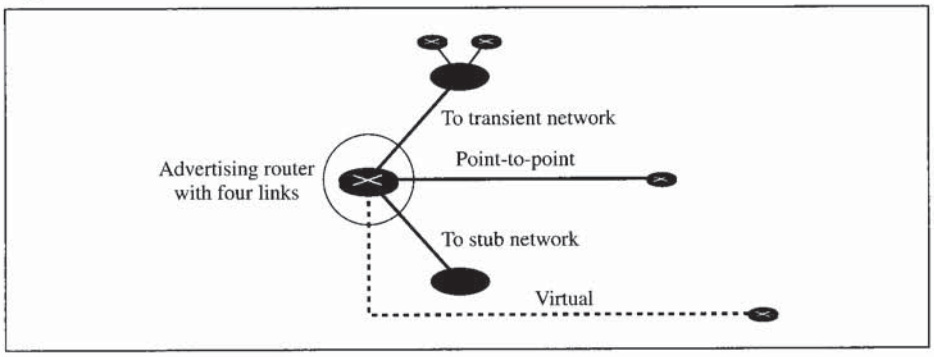
To share information about their neighbors, each entity distributes Link State Advertisements (LSAs). An LSA announces the states of entity links. Depending on the type of entity, we can define five different LSAs (see Figure 13.25).

Figure 13.25 *Types of LSAs*



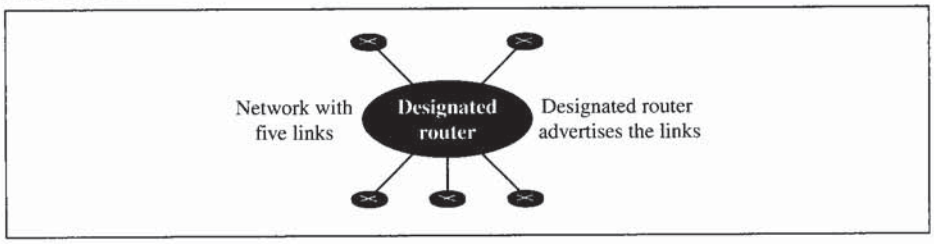
Router Link A router link defines the links of a true router. A true router uses this advertisement to announce information about all of its links and what is at the other side of the link (neighbors). See Figure 13.26 for a depiction of a router link.

Figure 13.26 *Router link*



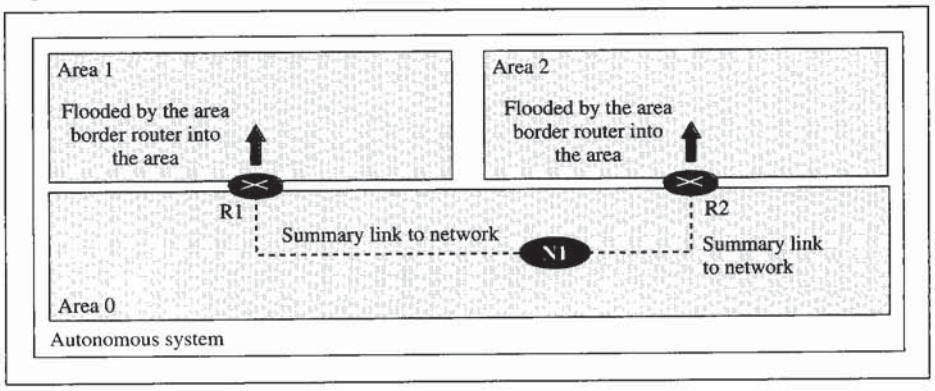
Network Link A network link defines the links of a network. A designated router, on behalf of the transient network, distributes this type of LSA packet. The packet announces the existence of all of the routers connected to the network (see Figure 13.27).

Figure 13.27 *Network link*



Summary Link to Network Router link and network link advertisements flood the area with information about the router links and network links inside an area. But a router must also know about the networks outside its area, and the area border routers can provide this information. An area border router is active in more than one area. It receives router link and network link advertisements, and, as we will see, creates a routing table for each area. For example, in Figure 13.28, router R1 is an area border router.

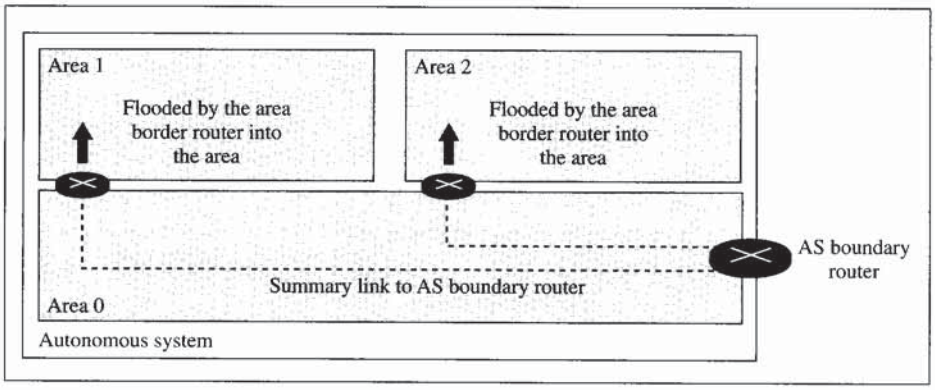
Figure 13.28 *Summary link to network*



It has two routing tables, one for area 1 and one for area 0. R1 floods area 1 with information about how to reach a network located in area 0. In the same way, router R2 floods area 2 with information about how to reach the same network in area 0.

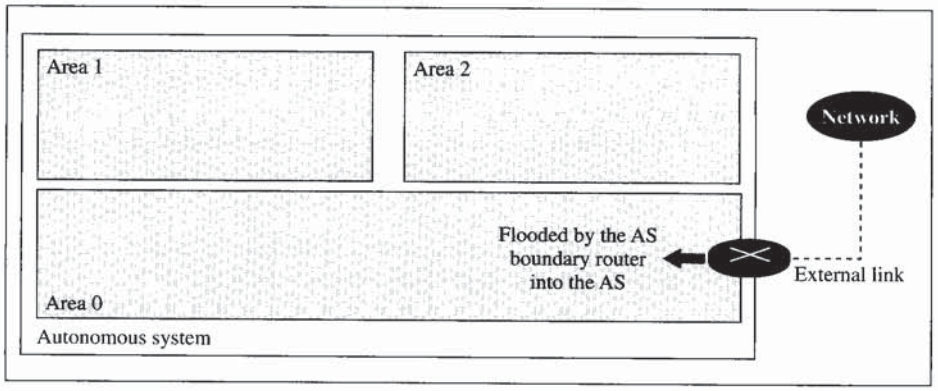
Summary Link to AS Boundary Router The previous advertisement lets every router know the cost to reach all of the networks inside the autonomous system. But what about a network outside the autonomous system? If a router inside an area wants to send a packet outside the autonomous system, it should first know the route to an autonomous boundary router; the summary link to AS boundary router provides this information. The area border routers flood their areas with this information (see Figure 13.29).

Figure 13.29 *Summary link to AS boundary router*



External Link Although the previous advertisement lets each router know the route to an AS boundary router, this information is not enough. A router inside an autonomous system wants to know which networks are available outside the autonomous system; the external link advertisement provides this information. The AS boundary router floods the autonomous system with the cost of each network outside the autonomous system using a routing table created by an exterior routing protocol. Each advertisement announces one single network. If there is more than one network, separate announcements are made. Figure 13.30 depicts an external link.

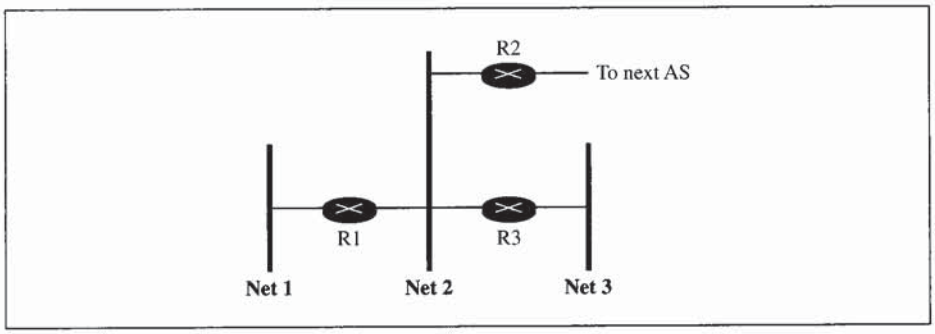
Figure 13.30 *External link*



Example 3

In Figure 13.31, which router(s) sends out router link LSAs?

Figure 13.31 *Example 3 and Example 4*



Solution

All routers advertise router link LSAs.

a. R1 has two links, Net1 and Net2.
b. R2 has one link, Net1 in this AS.
c. R3 has two links, Net2 and Net3.

Example 4

In Figure 13.31, which router(s) sends out the network link LSAs?

Solution

All three network must advertise network links:

a. Advertisement for Net1 is done by R1 because it is the only router and therefore the designated router.
b. Advertisement for Net2 can be done by either R1, R2, or R3, depending on which one is chosen as the designated router.
c. Advertisement for Net3 is done by R3 because it is the only router and therefore the designated router.

Link State Database

Every router in an area receives the router link and network link LSAs from every other router and forms a link state database. Note that every router in the same area has the same link state database.

A link state database is a tabular representation of the topology of the internet inside an area. It shows the relationship between each router and its neighbors including the metrics.

In OSPF, all routers have the same link state database.

Dijkstra Algorithm

To calculate its routing table, each router applies the Dijkstra algorithm to its link state database. The **Dijkstra algorithm** calculates the shortest path between two points on a network using a graph made up of nodes and edge. The algorithm divides the nodes into two sets: tentative and permanent. It chooses nodes, makes them tentative, examines them, and if they pass the criteria, makes them permanent. We can informally define the algorithm using the following steps:

Dijkstra Algorithm

1. Start with the local node (router): the root of the tree.
2. Assign a cost of 0 to this node and make it the first permanent node.
3. Examine each neighbor node of the node that was the last permanent node.
4. Assign a cumulative cost to each node and make it tentative.
5. Among the list of tentative nodes
    1. Find the node with the smallest cumulative cost and make it permanent.
    2. If a node can be reached from more than one direction
        1. Select the direction with the shortest cumulative cost.
6. Repeat steps 3 to 5 until every node becomes permanent.

Figure 13.32 shows the steps of the Dijkstra algorithm applied to node A of our sample internet in Figure 13.24. The number next to each node represents the cumulative cost from the root node. Note that in step h, network N3 is reached through two directions with cumulative costs of 14 and 10. The direction with the cumulative cost of 10 is kept and the other one is deleted.

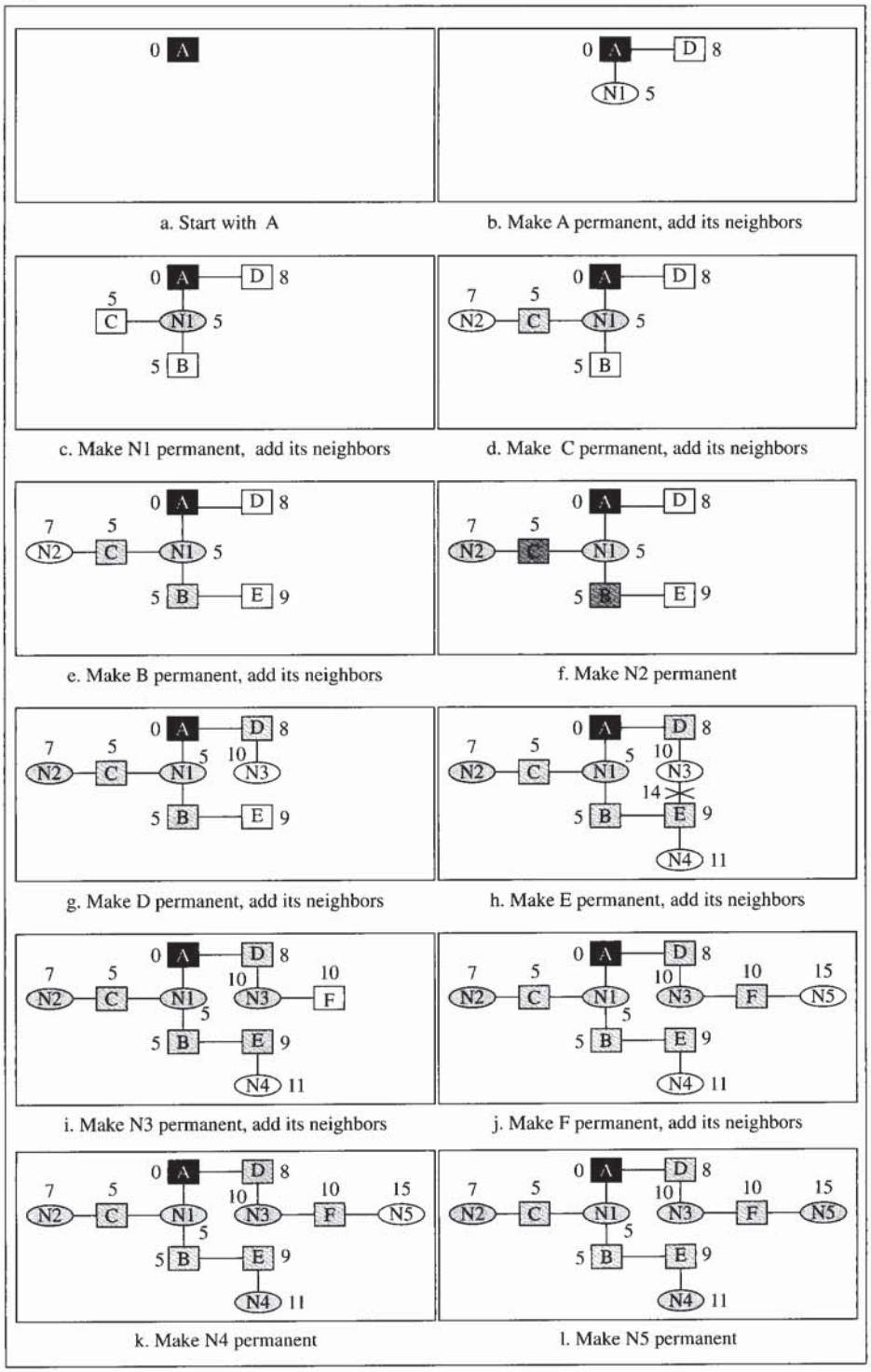
Routing Table

Each router uses the shortest path tree method to construct its routing table. The routing table shows the cost of reaching each network in the area. To find the cost of reaching networks outside of the area, the routers use the summary link to network, the summary link to boundary router, and the external link advertisements. Table 13.2 shows the routing table for router A.

Table 13.2 *Link state routing table for router A*

Network	*Cost*	*Next Router*	*Other Information*
N1	5		
N2	7	C	
N3	10	D	
N4	11	B	
N5	15	D	

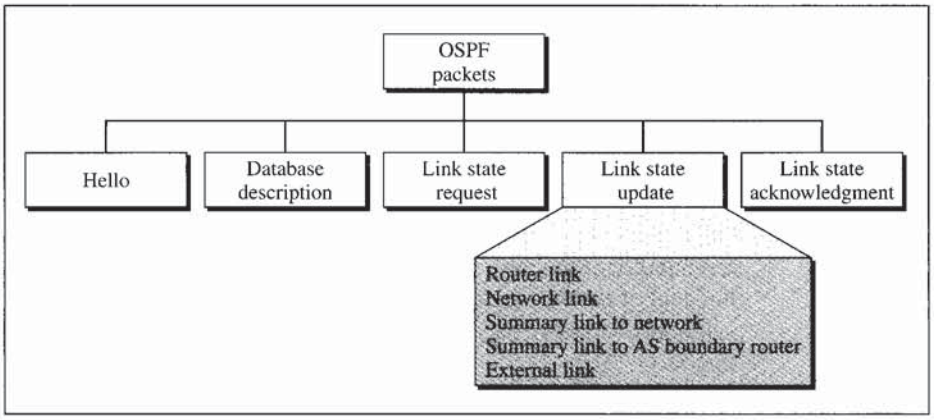
Figure 13.32 *Shortest path calculation*



Types of Packets

OSPF uses five different types of packets: the hello packet, database description packet, link state request packet, link state update packet, and link state acknowledgment packet (see Figure 13.33).

Figure 13.33 *Types of OSPF packets*

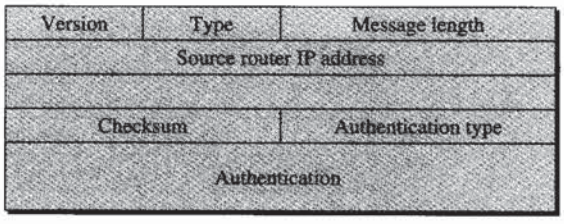


Packet Format

All OSPF packets share the same common header (see Figure 13.34). Before studying the different types of packets, let us talk about this common header.

- **Version.** This 8-bit field defines the version of the OSPF protocol. It is currently version 2.
- **Type.** This 8-bit field defines the type of the packet. As we said before, we have five types, with values 1 to 5 defining the types.
- **Message length.** This 16-bit field defines the length of the total message including the header.
- **Source router IP address.** This 32-bit field defines the IP address of the router that sends the packet.

Figure 13.34 *OSPF packet header*



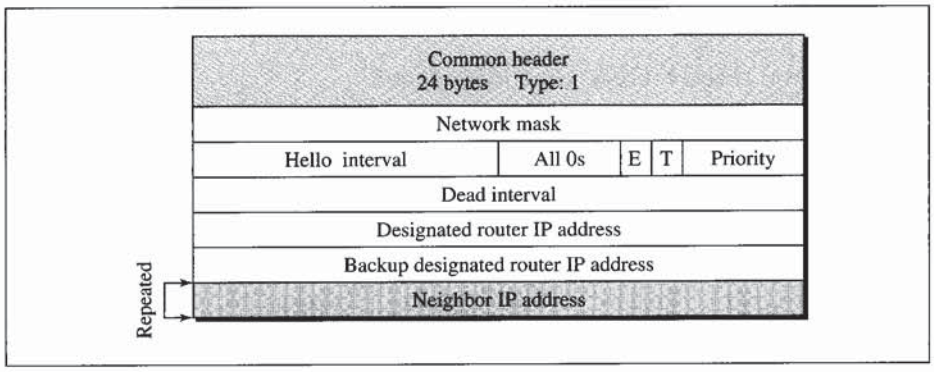
- **Area identification.** This 32-bit field defines the area within which the routing takes place.
- **Checksum.** This field is used for error detection on the entire packet excluding the authentication type and authentication data field.
- **Authentication type.** This 16-bit field defines the authentication method used in this area. At this time, two types of authentication are defined: 0 for none and 1 for password.
- **Authentication.** This 64-bit field is the actual value of the authentication data. In the future, when more authentication types are defined, this field will contain the result of the authentication calculation. For now, if the authentication type is 0, this field is filled with 0s. If the type is 1, this field carries an eight-character password.

Hello Message

OSPF uses the hello message to create neighborhood relationships and to test the reachability of neighbors. This is the first step in link state routing. Before a router can flood all of the other routers with information about its neighbors, it must first greet its neighbors. It must know if they are alive, and it must know if they are reachable (see Figure 13.35).

- **Network mask.** This 32-bit field defines the network mask of the network over which the hello message is sent.
- **Hello interval.** This 16-bit field defines the number of seconds between hello messages.
- **E flag.** This is a 1-bit flag. When it is set, it means that the area is a stub area.
- **T flag.** This is a 1-bit flag. When it is set, it means that the router supports multiple metrics.
- **Priority.** This field defines the priority of the router. The priority is used for the selection of the designated router. After all neighbors declare their priorities, the router with the highest priority is chosen as the designated router. The one with the second highest priority is chosen as the backup designated router. If the value

Figure 13.35 *Hello packet*



of this field is 0, it means that the router never wants to be a designated or backup designated router.

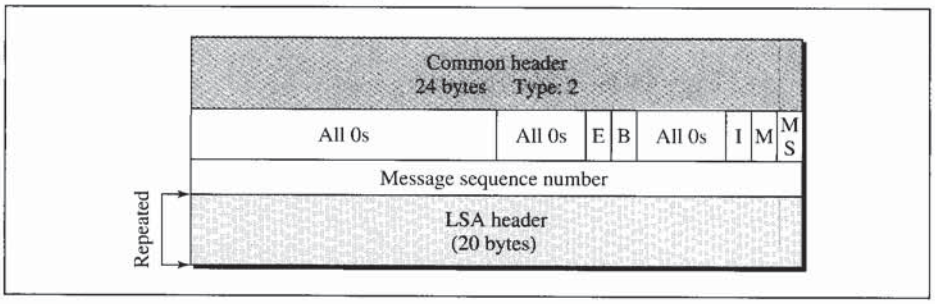
- **Dead interval.** This 32-bit field defines the number of seconds before a router assumes that a neighbor is dead.
- **Designated router IP address.** This 32-bit field is the IP address of the designated router for the network over which the message is sent.
- **Backup designated router IP address.** This 32-bit field is the IP address of the backup designated router for the network over which the message is sent.
- **Neighbor IP address.** This is a repeated 32-bit field that defines the routers that have agreed to be the neighbors of the sending router. In other words, it is a current list of all the neighbors from which the sending router has received the hello message.

Database Description Message

When a router is connected to the system for the first time or after a failure, it needs the complete link state database immediately. It cannot wait for all link state update packets to come from every other router before making its own database and calculating its routing table. Therefore, after a router is connected to the system, it sends hello packets to greet its neighbors. If this is the first time that the neighbors hear from the router, they send a database description packet. The database description packet does not contain complete database information; it only gives an outline, the title of each line in the database. The newly connected router examines the outline and finds out which lines of information it does not have. It then sends one or more link state request packets to get full information about that particular link. When two routers want to exchange database description packets, one of them takes the role of master and the other the role of slave. Because the message can be very long, the contents of the database can be divided into several messages. The format of the database description packet is shown in Figure 13.36. The fields are as follows:

- **E flag.** This 1-bit flag is set to 1 if the advertising router is an autonomous boundary router (*E* stands for external).
- **B flag.** This 1-bit flag is set to 1 if the advertising router is an area border router.

Figure 13.36 *Database description packet*

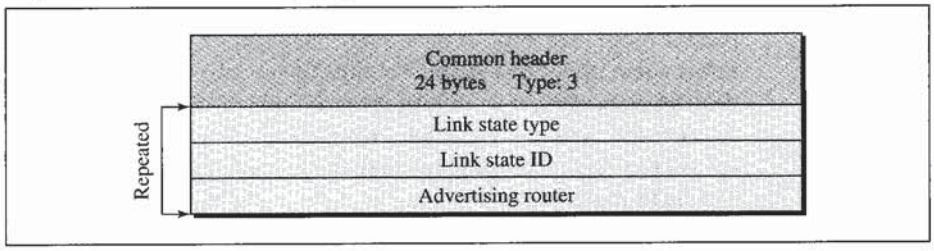


- **I flag.** This 1-bit field, the *initialization* flag, is set to 1 if the message is the first message.
- **M flag.** This 1-bit field, the *more* flag, is set to 1 if this is not the last message.
- **M/S flag.** This 1-bit field, the *master/slave* bit, indicates the origin of the packet: master (M/S = 1) or slave (M/S = 0).
- **Message sequence number.** This 32-bit field contains the sequence number of the message. It is used to match a request with the response.
- **LSA header.** This 20-byte field is used in each LSA. The format of this header is discussed in the link state update message section. This header gives the outline of each link, without details. It is repeated for each link in the link state database.

Link State Request Packet

The format of the link state request packet is shown in Figure 13.37. This is a packet that is sent by a router that needs information about a specific route or routes. It is answered with a link state update packet. It can be used by a newly connected router to request more information about some routes after receiving the database description packet. The three fields here are part of the LSA header which we will see shortly. Each set of the three fields is a request for one single LSA. The set is repeated if more than one advertisement is desired.

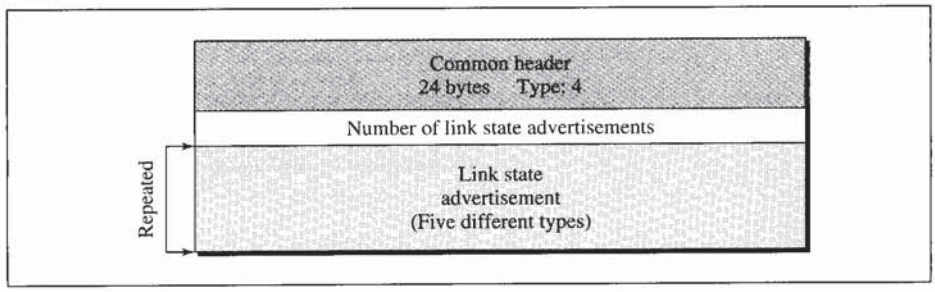
Figure 13.37 *Link state request packet*



Link State Update Packet

The link state update packet is the heart of the OSPF operation. It is used by a router to advertise the states of its links. The general format of the link state update packet is shown in Figure 13.38. Each update packet may contain several different LSAs. For

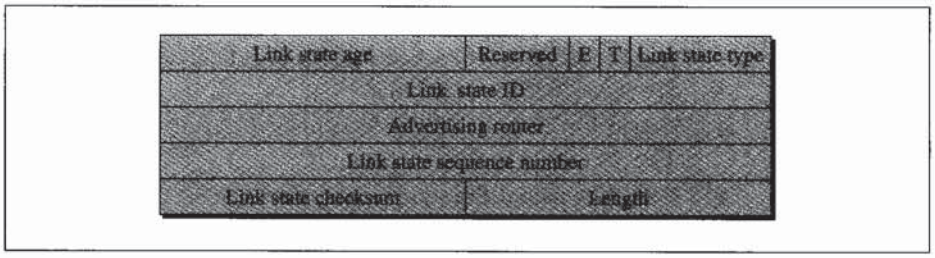
Figure 13.38 *Link state update packet*



example, a link state update packet can contain 14 LSAs, four of which are router link advertisements, three of which are network link advertisements, two of which are summary link to network advertisements, two of which are summary link to AS boundary router advertisements, and three of which are external link advertisements. The fields are as follows:

- **Number of advertisements.** This 32-bit field defines the number of advertisements. One packet can advertise the states of several links.
- **Link state advertisements.** There are five different LSAs, as we discussed before, all having the same header format, but different bodies. We first discuss the general header common to all of them. The format of the LSA header is shown in Figure 13.39.

Figure 13.39 *LSA header*



- **Link state age.** This field indicates the number of seconds elapsed since this message was first generated. Recall that this type of message goes from router to router (flooding). When a router creates the message, the value of this field is 0. When each successive router forwards this message, it estimates the transit time and adds it to the cumulative value of this field.
- **E flag.** If this 1-bit flag is set to 1, it means that the area is a stub area. A stub area is an area that is connected to the backbone area by only one path.
- **T flag.** If this 1-bit flag is set to 1, it means that the router can handle multiple types of service.
- **Link state type.** This field defines the LSA type. As we discussed before, there are five different advertisement types: router link (1), network link (2), summary link to network (3), summary link to AS boundary router (4), and external link (5).
- **Link state ID.** The value of this field depends on the type of link. For type 1 (router link), it is the IP address of the router. For type 2 (network link), it is the IP address of the designated router. For type 3 (summary link to network), it is the address of the network. For type 4 (summary link to AS boundary router), it is the IP address of the AS boundary router. For type 5 (external link), it is the address of the external network.
- **Advertising router.** This is the IP address of the router advertising this message.
- **Link state sequence number.** This is a sequence number assigned to each link state update message.

- **Link state checksum.** This is not the usual checksum field. It uses a special checksum calculation called *Fletcher's checksum,* which is based on the whole packet except for the age field.
- **Length.** This defines the length of the whole packet in bytes.

Router Link LSA The router link LSA advertises all of the links of a router (true router). The format of the router link packet is shown in Figure 13.40. The fields of the router link LSA are as follows:

- **Link ID.** The value of this field depends on the type of link. Table 13.3 shows the different link identifications based on link type.
- **Link data.** This field gives additional information about the link. Again, the value depends on the type of the link (see Table 13.3).
- **Link type.** Four different types of links are defined based on the type of network to which the router is connected (see Table 13.3).
- **Number of types of service (TOS).** This field defines the number of types of services announced for each link.
- **Metric for TOS 0.** This field defines the metric for the default type of service (TOS 0).
- **TOS.** This field defines the type of service.
- **Metric.** This field defines the metric for the corresponding TOS.

Figure 13.40 *Router link LSA*

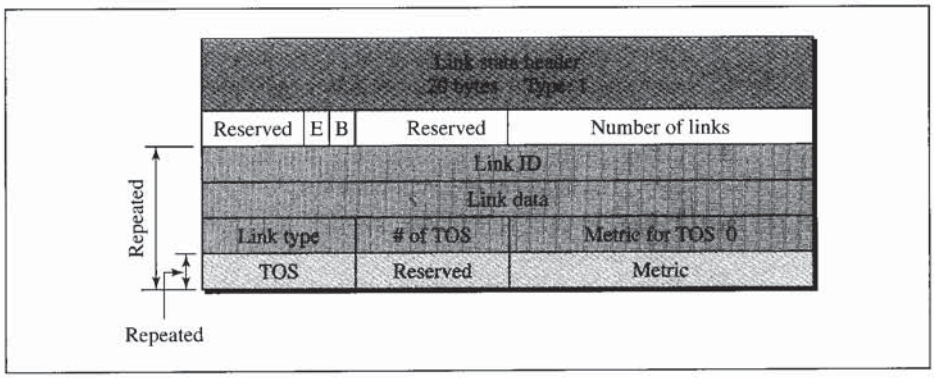


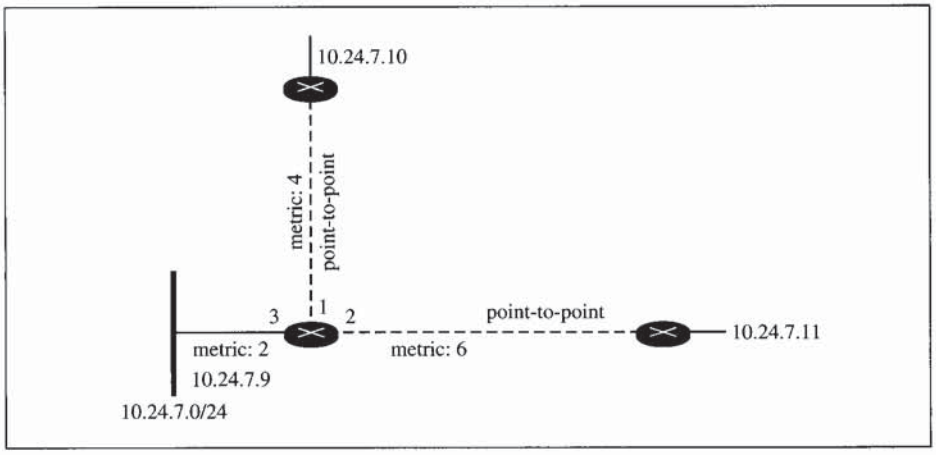
Table 13.3 *Link types, link identification, and link data*

Link Type	*Link Identification*	*Link Data*
Type 1: Point-to-point connection to another router	Address of neighbor router	Interface number
Type 2: Connection to any-to-any network	Address of designated router	Router address
Type 3: Connection to stub network	Network address	Network mask
Type 4: Virtual link	Address of neighbor router	Router address

Example 5

Give the router link LSA sent by router 10.24.7.9 in Figure 13.41.

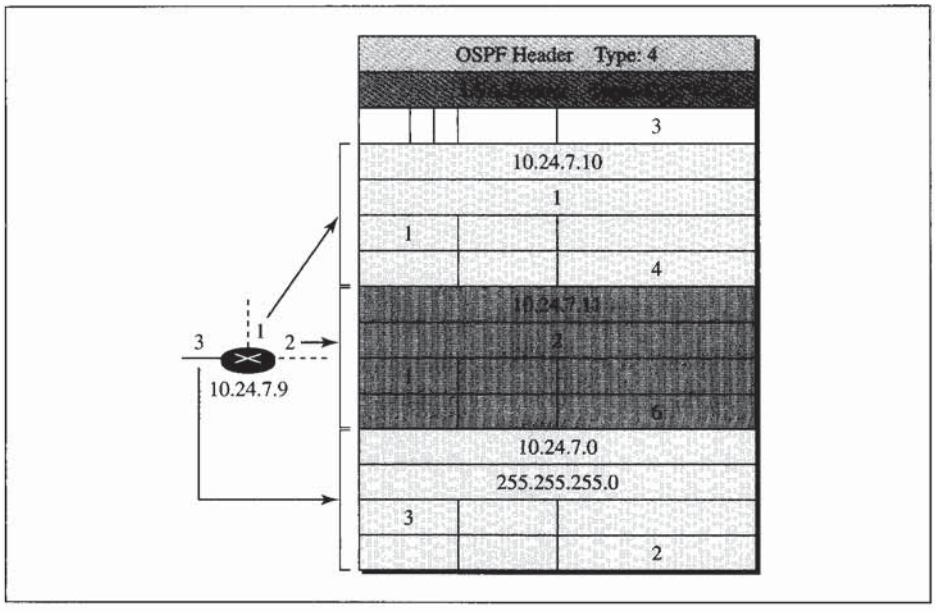
Figure 13.41 *Example 5*



Solution

This router has three links: two of type 1 (point-to-point) and one of type 3 (stub network). Figure 13.42 shows the router link LSA.

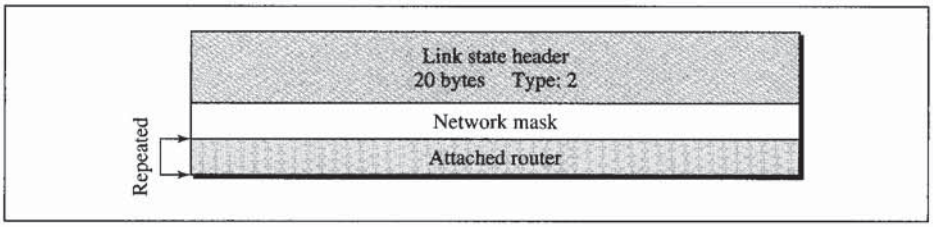
Figure 13.42 *Solution to Example 5*



Network Link LSA The network link advertisement announces the links connected to a network. The format of the network link advertisement is shown in Figure 13.43. The fields of the network link LSA are as follows:

- **Network mask.** This field defines the network mask.
- **Attached router.** This repeated field defines the IP addresses of all attached routers.

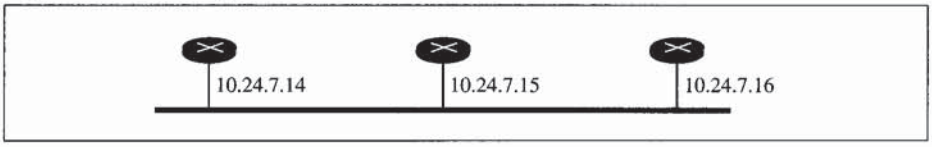
Figure 13.43 *Network link advertisement format*



Example 6

Give the network link LSA in Figure 13.44.

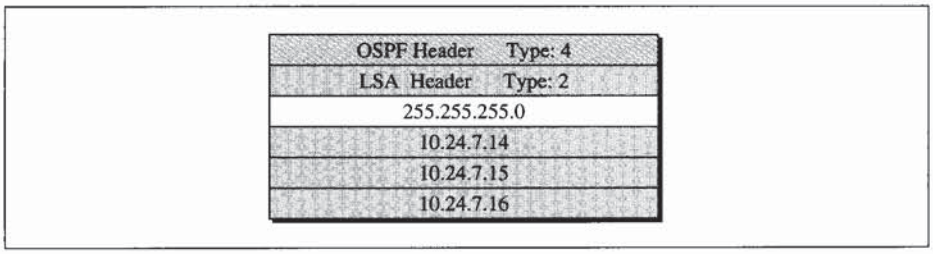
Figure 13.44 *Example 6*



Solution

The network, for which the network link advertises, has three routers attached. The LSA shows the mask and the router addresses. Figure 13.45 shows the network link LSA.

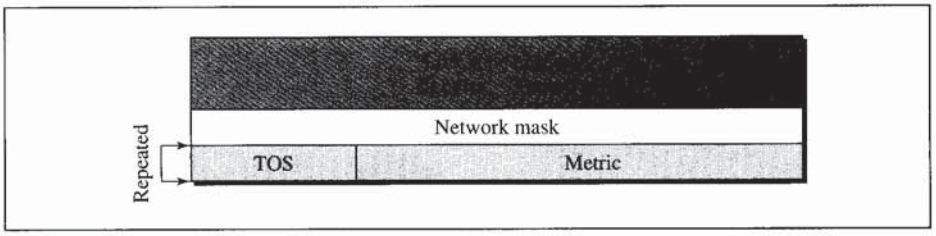
Figure 13.45 *Solution to Example 6*



Summary Link to Network LSA This is used by the area border router to announce the existence of other networks outside the area. The summary link to network advertisement is very simple. It consists of the network mask and the metric for each type of service. Note that each advertisement announces only one single network. If there is

more than one network, a separate advertisement must be issued for each. The reader may ask why only the mask of the network is advertised. What about the network address itself? The IP address of the advertising router is announced in the header of the link state advertisement. From this information and the mask, one can deduce the network address. The format of this advertisement is shown in Figure 13.46. The fields of the summary link to network LSA are as follows:

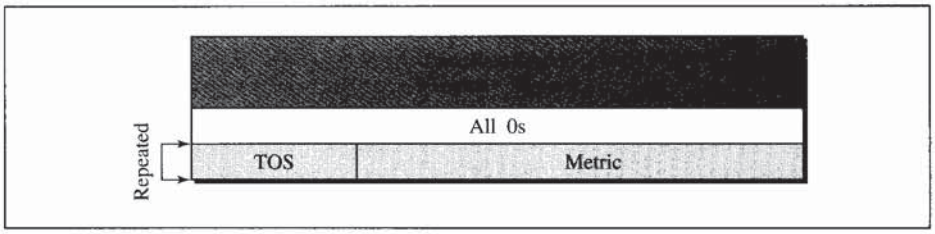
Figure 13.46 *Summary link to network LSA*



- **Network mask.** This field defines the network mask.
- **TOS.** This field defines the type of service.
- **Metric.** This field defines the metric for the type of service defined in the TOS field.

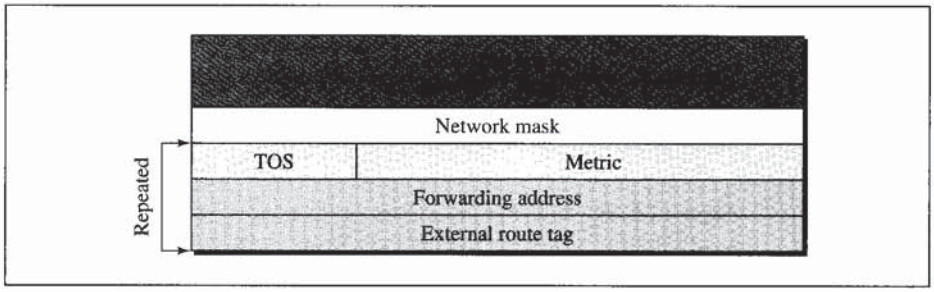
Summary Link to AS Boundary Router LSA This packet is used to announce the route to an AS boundary router. Its format is the same as the previous summary link. The packet just defines the network to which the AS boundary router is attached. If a message can reach the network, it can be picked up by the AS boundary router. The format of the packet is shown in Figure 13.47. The fields are the same as the fields in the summary link to network advertisement message.

Figure 13.47 *Summary link to AS boundary router LSA*



External Link LSA This is used to announce all the networks outside the AS. The format of the LSA is similar to the summary link to AS boundary router LSA, with the addition of two fields. The AS boundary router may define a forwarding router that can provide a better route to the destination. The packet also can include an external route tag, used by other protocols, but not by OSPF. The format of the packet is shown in Figure 13.48.

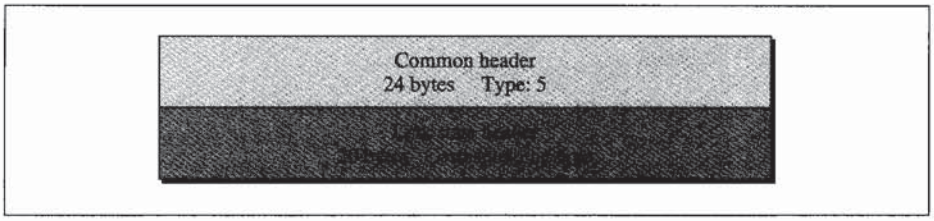
Figure 13.48 *External link LSA*



Link State Acknowledgment Packet

OSPF makes routing more reliable by forcing every router to acknowledge the receipt of every link state update packet. The format of the link state acknowledgment packet is shown in Figure 13.49. It has the common OSPF header and the generic link state update header. These two sections are sufficient to acknowledge a packet.

Figure 13.49 *Link state acknowledgment packet*



Encapsulation

OSPF packets are encapsulated in IP datagrams. They contain the acknowledgment mechanism for flow and error control. They do not need a transport layer protocol to provide these services.

OSPF packets are encapsulated in IP datagrams.

13.4 BGP

Border Gateway Protocol (BGP) is an inter-autonomous system routing protocol. It first appeared in 1989 and has gone through four versions. BGP is based on a routing method called *path vector routing*. However, before describing the principle behind path vector routing, let us see why the two previously discussed methods—namely, distance vector routing and link state routing—are not good candidates for inter-autonomous system routing.

Distance vector is not a good candidate because there are occasions in which the route with the smallest hop count is not the preferred route. For example, we may not

want a packet to pass through an autonomous system that is not secure even though it is the shortest route. Also, distance vector routing is unstable due to the fact that the routers announce only the number of hop counts to the destination without actually defining the path that leads to that destination. A router that receives a distance vector advertisement packet may be fooled if the shortest path is actually calculated through the receiving router itself.

Link state routing is also not a good candidate for inter-autonomous system routing because an internet is usually too big for this routing method. To use link state routing for the whole internet would require each router to have a huge link state database. It would also take a long time for each router to calculate its routing table using the Dijkstra algorithm.

Path Vector Routing

Path vector routing is different from both distance vector routing and link state routing. Each entry in the routing table contains the destination network, the next router, and the path to reach the destination. The path is usually defined as an ordered list of autonomous systems that a packet should travel through to reach the destination. Table 13.4 shows an example of a path vector routing table.

Table 13.4 *Path vector routing table*

Network	*Next Router*	*Path*
N01	R01	AS14, AS23, AS67
N02	R05	AS22, AS67, AS05, AS89
N03	R06	AS67, AS89, AS09, AS34
N04	R12	AS62, AS02, AS09

Path Vector Messages

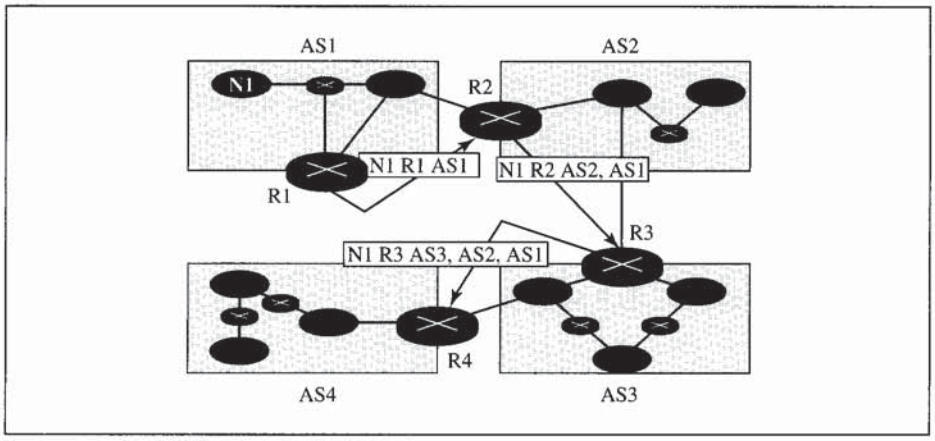
The autonomous boundary routers that participate in path vector routing advertise the reachability of the networks in their own autonomous systems to neighbor autonomous boundary routers. The concept of neighborhood here is the same as the one described in the RIP or OSPF protocol. Two autonomous boundary routers connected to the same network are neighbors.

We should mention here that an autonomous boundary router receives its information from an interior routing algorithm such as RIP or OSPF.

Each router that receives a path vector message verifies that the advertised path is in agreement with its policy (a set of rules imposed by the administrator controlling the routes). If it is, the router updates its routing table and modifies the message before sending it to the next neighbor. The modification consists of adding its AS number to the path and replacing the next router entry with its own identification.

For example, Figure 13.50 shows an internet with four autonomous systems. The router R1 sends a path vector message advertising the reachability of N1. Router R2 receives the message, updates its routing table, and after adding its autonomous system to the path and inserting itself as the next router, sends the message to router R3. Router R3

Figure 13.50 *Path vector packets*



receives the message, updates its routing table, and sends the message, after changes, to router R4.

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.

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. This means that the routing tables in path vector routing are not based on the smallest hop count or the minimum metric; they are based on the policy imposed on the router by the administrator.

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 better 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 should recognize. An *optional attribute* is one that need 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

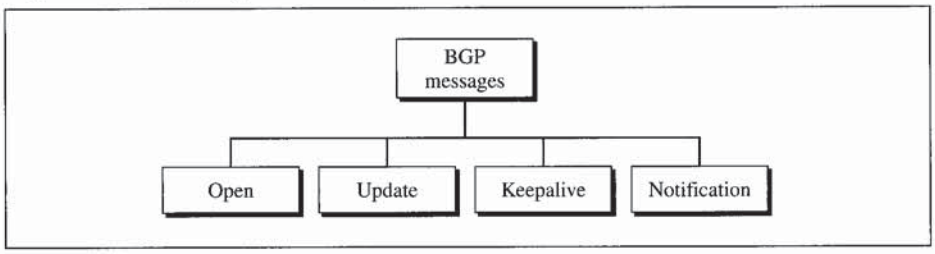
each router, but is not required to be included in every update message. One well-known mandatory attribute is ORIGIN. This defines the source of the routing information (RIP, OSPF, and so on). Another well-known mandatory attribute is AS_PATH. This defines the list of autonomous systems through which 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 should be discarded if the receiving router has not implemented it.

Types of Packets

BGP uses four different types of messages: open, update, keepalive, and notification (see Figure 13.51).

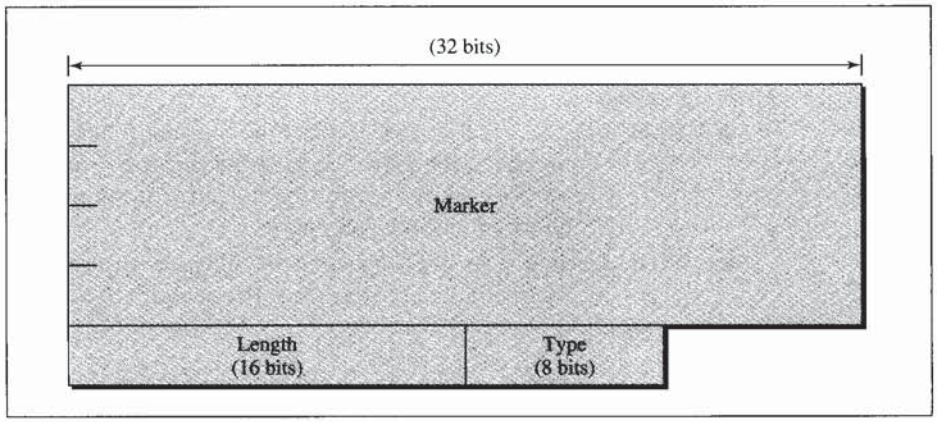
Figure 13.51 *Types of BGP messages*



Packet Format

All BGP packets share the same common header. Before studying the different types of packets, let us talk about this common header (see Figure 13.52). The fields of this

Figure 13.52 *BGP packet header*



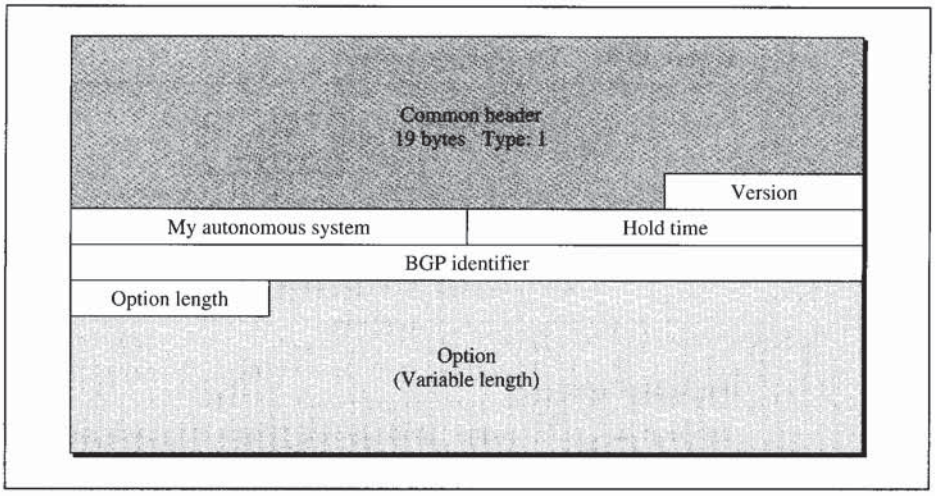
header are as follows:

- **Marker.** The 16-byte marker field is reserved for authentication.
- **Length.** This 2-byte field defines the length of the total message including the header.
- **Type.** This 1-byte field defines the type of the packet. As we said before, we have four types, and the values of 1 to 4 define those types.

Open Message

To create a neighborhood relationship, a router running BGP opens a TCP connection with a neighbor and sends an open message. If the neighbor accepts the neighborhood relationship, it responds with a keepalive message, which means that a relationship has been established between the two routers. See Figure 13.53 for a depiction of the open message format.

Figure 13.53 *Open message*



The fields of the open message are as follows:

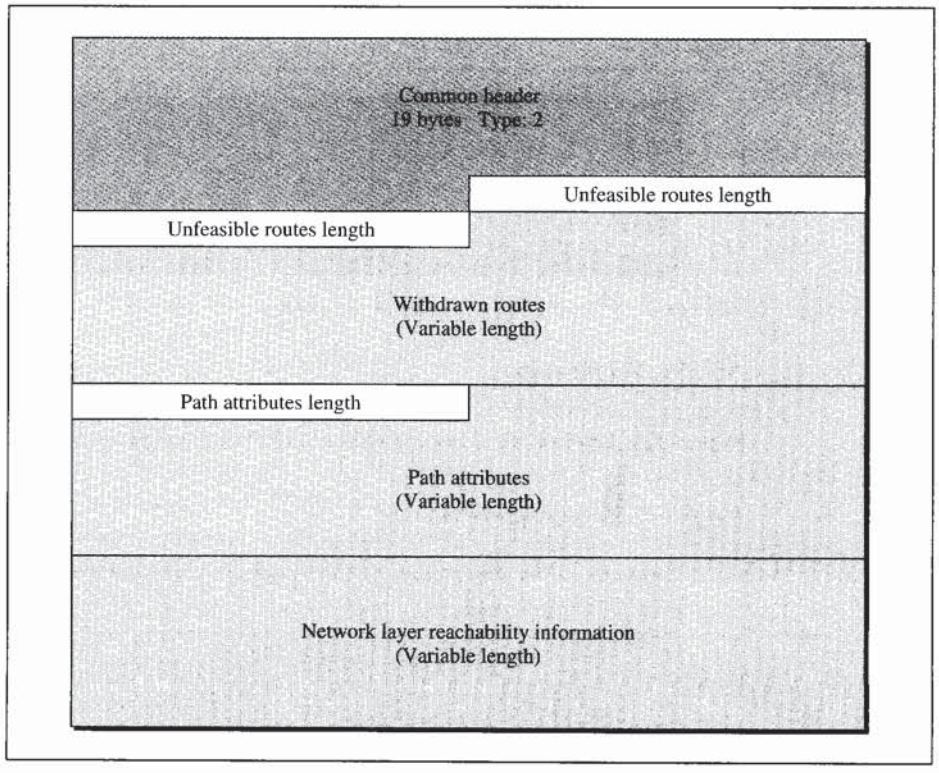
- **Version.** This 1-byte field defines the version of BGP. The current version is 4.
- **My autonomous system.** This 2-byte field defines the autonomous system number.
- **Hold time.** This 2-byte field defines the maximum number of seconds that can elapse before one of the parties receives a keepalive or update message from the other. If a router does not receive one of these messages during the hold time period, it considers the other party dead.
- **BGP identifier.** This is a 4-byte field defining the router that sends the open message. The router usually uses one of its IP addresses (because it is unique) for this purpose.

- **Option parameter length.** The open message may also contain some option parameters. If so, this 1-byte field defines the length of the total option parameters. If there are no option parameters, the value of this field is zero.
- **Option parameters.** If the value of the option parameter length is not zero, it means that there are some option parameters. Each option parameter itself has two subfields: the length of the parameter and the parameter value. The only option parameter defined so far is authentication.

Update Message

The update message is the heart of the BGP protocol. It is used by a router to withdraw destinations that have been advertised previously, announce a route to a new destination, or both. Note that BGP can withdraw several destinations that were advertised before, but it can only advertise one new destination in a single update message. The format of the update message is shown in Figure 13.54.

Figure 13.54 *Update message*



The update message fields are listed below:

- **Unfeasible routes length.** This 2-byte field defines the length of the next field.
- **Withdrawn routes.** This field lists all the routes that should be deleted from the previously advertised list.
- **Path attributes length.** This 2-byte field defines the length of the next field.

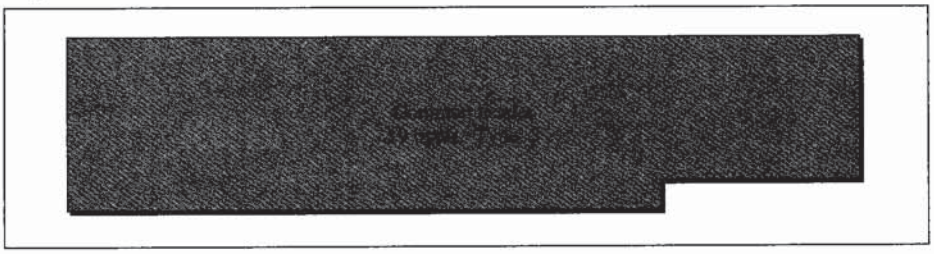
- **Path attributes.** This field defines the attributes of the path (route) to the network whose reachability is being announced in this message.
- **Network layer reachability information (NLRI).** This field defines the network that is actually advertised by this message. It has a length field and an IP address prefix. The length defines the number of bits in the prefix. The prefix defines the common part of the network address. For example, if the network is 153.18.7.0/24. The length of the prefix is 24 and the prefix is 153.18.7. This means that BGP4 supports classless addressing and CIDR.

BGP supports classless addressing and CIDR.

Keepalive Message

The routers (called *peers* in BGP parlance), running the BGP protocols, exchange keepalive messages regularly (before their hold time expires) to tell each other that they are alive. The keepalive message consists of only the common header shown in Figure 13.55.

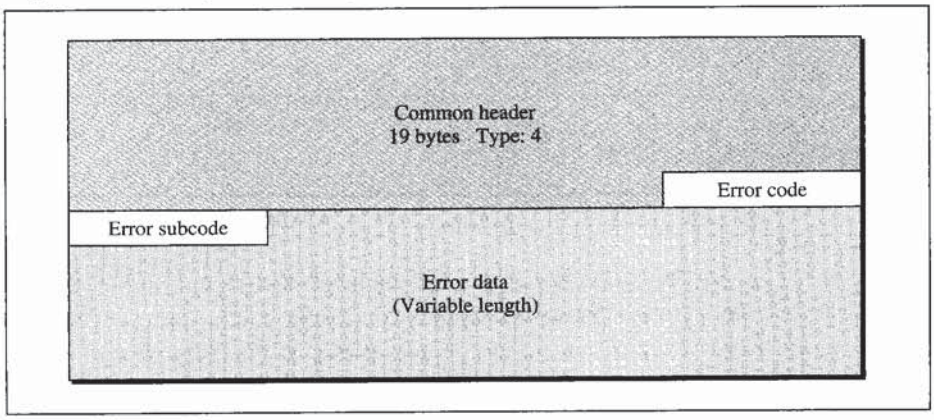
Figure 13.55 *Keepalive message*



Notification Message

A notification message is sent by a router whenever an error condition is detected or a router wants to close the connection. The format of the message is shown in Figure 13.56.

Figure 13.56 *Notification message*



The fields comprising the notification message follow:

- **Error code.** This 1-byte field defines the category of the error. See Table 13.5.
- **Error subcode.** This 1-byte field further defines the type of error in each category.
- **Error data.** This field can be used to give more diagnostic information about the error.

Table 13.5 *Error codes*

Error Code	*Error Code Description*	*Error Subcode Description*
1	Message header error	Three different subcodes are defined for this type of error: synchronization problem (1), bad message length (2), and bad message type (3).
2	Open message error	Six different subcodes are defined for this type of error: unsupported version number (1), bad peer AS (2), bad BGP identifier (3), unsupported optional parameter (4), authentication failure (5), and unacceptable hold time (6).
3	Update message error	Eleven different subcodes are defined for this type of error: malformed attribute list (1), unrecognized well-known attribute (2), missing well-known attribute (3), attribute flag error (4), attribute length error (5), invalid origin attribute (6), AS\ routing loop (7), invalid next hop attribute (8), optional attribute error (9), invalid network field (10), malformed AS_PATH (11).
4	Hold timer expired	No subcode defined.
5	Finite state machine error	This defines the procedural error. No subcode defined.
6	Cease	No subcode defined.

Encapsulation

BGP messages are encapsulated in TCP segments using the well-known port 179. This means that there is no need for error control and flow control. When a TCP connection is opened, the exchange of update, keepalive, and notification messages is continued until a notification message of type cease is sent.

BGP uses the services of TCP on port 179.

13.5 KEY TERMS

area
area border router
area identification
authentication
autonomous system (AS)
autonomous system boundary router
backbone router
Bellman-Ford algorithm
border gateway protocol (BGP)
database description message
dead interval field
Dijkstra algorithm
distance vector multicast routing protocol (DVMRP)
distance vector routing
encapsulation
expiration timer
exterior routing
external link
external link LSA
flooding
garbage collection timer
hello interval
hello message
hop count
inter-autonomous routing protocol
interior routing
keepalive message
link state acknowledgment packet
link state advertisement (LSA)
link state database
link state request packet
link state routing
link state update packet
MBONE
metric
multicasting
network link LSA
network mask
next-hop address
notification message
open message
open shortest path first (OSPF)
optional attribute
packet
path vector routing
periodic timer
point-to-point transmission
poison reverse
policy routing
RIPv2
router
router link LSA
routing information protocol (RIP)
routing table
slow convergence
solicited response
split horizon
static routing table

stub link

subnet mask

summary link to AS boundary router LSA

summary link to network LSA

transient link

triggered update process

unsolicited response

update message

virtual link

well-known attribute

13.6 SUMMARY

- A metric is the cost assigned for passage of a packet through a network.
- A router consults its routing table to determine the best path for a packet.
- An autonomous system (AS) is a group of networks and routers under the authority of a single administration.
- RIP and OSPF are popular interior routing protocols used to update routing tables in an AS.
- RIP is based on distance vector routing, in which each router shares, at regular intervals, its knowledge about the entire AS with its neighbors.
- A RIP routing table entry consists of a destination network address, the hop count to that destination, and the IP address of the next router.
- RIP uses three timers: the periodic timer controls the advertising of the update message, the expiration timer governs the validity of a route, and the garbage collection timer advertises the failure of a route.
- 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.
- The RIP version 2 packet format contains fields carrying AS information and authentication information.
- OSPF divides an AS into areas, defined as collections of networks, hosts, and routers.
- 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 defines four types of links (networks): point-to-point, transient, stub, and virtual.
- Five types of link state advertisements (LSAs) disperse information in OSPF: router link, network link, summary link to network, summary link to AS boundary router, and external link.
- A router compiles all the information from the LSAs it receives into a link state database. This database is common to all routers in an area.
- OSPF routing tables are calculated using Dijkstra's algorithm.

- There are five types of OSPF packets: hello, database description, link state request, link state update, and link state acknowledgment.
- An LSA is a multifield entry in a link state update packet.
- BGP is an inter-autonomous system routing protocol used to update routing tables.
- BGP is based on a routing method called path vector routing. In this method, the ASs through which a packet must pass are explicitly listed.
- Path vector routing does not have the instability nor looping problems of distance vector routing.
- There are four types of BGP messages: open, update, keepalive, and notification.

13.7 PRACTICE SET

Multiple-Choice Questions

1. RIP is based on _____.
   a. link state routing
   b. distance vector routing
   c. Dijkstra's algorithm
   d. path vector routing
2. In distance vector routing each router receives information directly from _____.
   a. every router on the network
   b. every router less than two units away
   c. a table stored by the network hosts
   d. its neighbors only
3. In distance vector routing a router sends out information _____.
   a. at regularly scheduled intervals
   b. only when there is a change in its table
   c. only when a new host is added
   d. only when a new network is added
4. A routing table contains _____.
   a. the destination network ID
   b. the hop count to reach the network
   c. the router ID of the next hop
   d. all of the above
5. Router B receives an update from router A that indicates Net1 is two hops away. The next update from A says Net1 is five hops away. What value is entered in B's routing table for Net1? Assume the basic RIP is being used.
   a. 2
   b. 3
   c. 6
   d. 7

6. If the routing table contains four new entries, how many update messages must the router send to its one neighbor router?
   a. 1
   b. 2
   c. 3
   d. 4
7. The hop count field of a router's first table always has a value of _____.
   a. 0
   b. 1
   c. infinity
   d. some positive integer
8. Which field in the RIP message contains the message type?
   a. command
   b. version
   c. network address
   d. distance
9. Which field in the RIP message corresponds to the cost field of the routing table?
   a. command
   b. version
   c. network address
   d. distance
10. Which field in the RIP message corresponds to the network ID field of the routing table?
    a. command
    b. version
    c. network address
    d. distance
11. Which timer schedules the sending out of regular update messages?
    a. periodic
    b. expiration
    c. garbage collection
    d. b and c
12. Which timer can set the distance field to 16?
    a. periodic
    b. expiration
    c. garbage collection
    d. b and c

13. A periodic update message goes out at time = 37 s. A triggered update follows at time = 57 s. Assuming a period of 30 s, when does the next regular periodic update message go out?
    a. at time = 67 s
    b. at time = 87 s
    c. at time = 58 s
    d. at some random time after 57 s
14. Which of the following attempts to alleviate the slow convergence problem?
    a. hop count limit
    b. triggered update
    c. looping
    d. a and b
15. Which of the following features the immediate sending of an update when a change occurs?
    a. hop count limit
    b. triggered update
    c. split horizons
    d. poison reverse
16. Which of the following sets an outgoing distance field to 16 for networks which previously sent incoming information through the same interface?
    a. hop count limit
    b. triggered update
    c. split horizons
    d. poison reverse
17. Which of the following does not allow the sending of information about the same network through the same interface?
    a. hop count limit
    b. triggered update
    c. split horizons
    d. poison reverse
18. Dijkstra's algorithm is used to _____.
    a. create LSAs
    b. flood an internet with information
    c. calculate the routing tables
    d. create a link state database
19. An area is _____.
    a. part of an AS
    b. composed of at least two ASs
    c. another term for an internet
    d. a collection of stub areas

20. In an autonomous system with n areas, how many areas are connected to the backbone?
    a. 1
    b. $n - 1$
    c. n
    d. $n + 1$
21. An area border router can be connected to _____.
    a. only another router
    b. another router or another network
    c. only another network
    d. only another area border router
22. Which of the following usually has the least number of connections to other areas?
    a. an area
    b. an autonomous system
    c. a transient link
    d. a stub link
23. Which type of network using the OSPF protocol always consists of just two connected routers?
    a. point-to-point
    b. transient
    c. stub
    d. virtual
24. Which type of network using the OSPF protocol is the result of a break in a link between two routers?
    a. point-to-point
    b. transient
    c. stub
    d. virtual
25. Which type of network using the OSPF protocol can have five routers attached to it?
    a. point-to-point
    b. transient
    c. stub
    d. all of the above
26. A WAN using the OSPF protocol that connects two routers is an example of a _____ type of OSPF network.
    a. point-to-point
    b. transient
    c. stub
    d. virtual

27. An Ethernet LAN using the OSPF protocol with five attached routers can be called a _____ network.
    a. point-to-point
    b. transient
    c. stub
    d. virtual
28. Which layer produces the OSPF message?
    a. data link
    b. network
    c. transport
    d. application
29. Which OSPF packet floods the Internet with information to update the database?
    a. link state request message
    b. link state update message
    c. link state acknowledgment message
    d. database description message
30. Which type of OSPF message must be sent prior to the others?
    a. hello message
    b. link state acknowledgment message
    c. link state request message
    d. database description message
31. Which IP address is needed in the hello message?
    a. designated router
    b. backup designated router
    c. neighbor router
    d. all of the above
32. Which of the following is an exterior routing protocol?
    a. RIP
    b. OSPF
    c. BGP
    d. a and b
33. Which of the following is an interior routing protocol?
    a. RIP
    b. OSPF
    c. BGP
    d. a and b

34. The Dijkstra algorithm is related to _____.
    a. distance vector routing
    b. link state routing
    c. path vector routing
    d. a and b
35. OSPF is based on _____.
    a. distance vector routing
    b. link state routing
    c. path vector routing
    d. a and b
36. BGP is based on _____.
    a. distance vector routing
    b. link state routing
    c. path vector routing
    d. a and b
37. Which timer is reset when a new update message for a route is received?
    a. garbage collection timer
    b. expiration timer
    c. periodic timer
    d. convergence timer
38. Which timer controls the advertising of regular update messages?
    a. garbage collection timer
    b. expiration timer
    c. periodic timer
    d. convergence timer
39. Which timer is involved in purging an invalid route from a table?
    a. garbage collection timer
    b. expiration timer
    c. periodic timer
    d. convergence timer
40. Which type of BGP message creates a relationship between two routers?
    a. open
    b. update
    c. keepalive
    d. notification
41. Which type of BGP message announces a route to a new destination?
    a. open
    b. update
    c. keepalive
    d. notification

42. Which type of BGP message is sent by a system to notify another router of the sender's existence?
    a. open
    b. update
    c. keepalive
    d. notification
43. Which type of BGP message is sent by a router to close a connection?
    a. open
    b. update
    c. keepalive
    d. notification

Exercises

44. What is the purpose of RIP?
45. What are the functions of a RIP message?
46. Why is the expiration timer value six times that of the periodic timer value?
47. How does the hop count limit alleviate RIP's problems?
48. List RIP shortcomings and their corresponding fixes.
49. Compare split horizons and poison reverse. When would one be used in preference to the other?
50. What is the basis of classification for the four types of links defined by OSPF?
51. What is the purpose of the authentication type and authentication data fields?
52. Contrast and compare distance vector routing with link state routing.
53. Draw a flowchart of the steps involved when a router receives a distance vector message from a neighbor.
54. Why do OSPF messages propagate faster than RIP messages?
55. What is the size of a RIP message that advertises only one network? What is the size of a RIP message that advertises N packets? Devise a formula that shows the relationship between the number of networks advertised and the size of a RIP message.
56. A router running RIP has a routing table with 20 entries. How many periodic timers are needed to handle this table?
57. A router running RIP has a routing table with 20 entries. How many expiration timers are needed to handle this table?
58. A router running RIP has a routing table with 20 entries. How many garbage collection timers are needed to handle this table if five routes are invalid?
59. A router has the following RIP routing table:

Net1	4	B
Net2	2	C
Net3	1	F
Net4	5	G

What would be the contents of the table if the router receives the following RIP message from router C:

Net1	2
Net2	1
Net3	3
Net4	7

60. How many bytes are empty in a RIP message that advertises *N* networks?
61. A router has the following RIP routing table:

Net1	4	B
Net2	2	C
Net3	1	F
Net4	5	G

Show the response message sent by this router.

62. Using Figure 13.24, show the link state update/router link advertisement for router A.
63. Using Figure 13.24, show the link state update/router link advertisement for router D.
64. Using Figure 13.24, show the link state update/router link advertisement for router E.
65. Show the link state update/network link advertisement for network N2 in Figure 13.24.
66. Show the link state update/network link advertisement for network N4 in Figure 13.24.
67. Show the link state update/network link advertisement for network N5 in Figure 13.24.
68. In Figure 13.24 assume that the designated router for network N1 is router A. Show the link state update/network link advertisement for this network.
69. In Figure 13.24 assume that the designated router for network N3 is router D. Show the link state update/network link advertisement for this network.
70. Assign IP addresses to networks and routers in Figure 13.24.
71. Using the result of exercise 70, show the OSPF hello message sent by router C.
72. Using the result of exercise 70, show the OSPF database description message sent by router C.
73. Using the result of exercise 70, show the OSPF link state request message sent by router C.
74. Show the autonomous system with the following specifications:
    a. There are eight networks (N1 to N8)
    b. There are eight routers (R1 to R8)
    c. N1, N2, N3, N4, and N5 are Ethernet networks
    d. N6 is a Token Ring
    e. N7 and N8 are point-to-point networks
    f. R1 connects N1 and N2
    g. R2 connects N1 and N7
    h. R3 connects N2 and N8
    i. R4 connects N7 and N6
    j. R5 connects N6 and N3

k. R6 connects N6 and N4
l. R7 connects N6 and N5
m. R8 connects N8 and N5

75. Draw the graphical representation of the autonomous system of exercise 74 as seen by OSPF.
76. Which of the networks in exercise 74 is a transient network? Which is a stub network?
77. Show the BGP open message for router R1 in Figure 13.50.
78. Show the BGP update message for router R1 in Figure 13.50.
79. Show the BGP keepalive message for router R1 in Figure 13.50.
80. Show the BGP notification message for router R1 in Figure 13.50.

Programming Exercises

81. Write declarations for all RIP messages in C.
82. Write declarations for all OSPF messages in C.
83. Write declarations for all BGP messages in C.
84. Write C code to implement the routing algorithm for RIP.
85. Modify the code in exercise 84 to include triggered update.
86. Modify the code in exercise 84 to include split horizon.
87. Modify the code in exercise 84 to include poison reverse.
88. Write C code to implement Dijkstra's algorithm.