CHAPTER 6

Delivery and Routing of IP Packets

This chapter describes the delivery and routing of IP packets to their final destinations. By **delivery,** we mean the physical forwarding of the packets. Concepts such as connectionless and connection-oriented services, and direct and indirect delivery are discussed. By **routing,** we mean finding the route (next hop) for a datagram. We discuss routing methods, types of routing, the routing table, and the routing module.

6.1 CONNECTION-ORIENTED VERSUS CONNECTIONLESS SERVICES

Delivery of a packet in the network layer is accomplished using either a connection-oriented or a connectionless network service.

In a connection-oriented situation, the network layer protocol first makes a connection with the network layer protocol at the remote site before sending a packet. When the connection is established, a sequence of packets from the same source to the same destination can be sent one after another. In this case, there is a relationship between packets. They are sent on the same path in sequential order. A packet is logically connected to the packet traveling before it and to the packet traveling after it. When all packets of a message have been delivered, the connection is terminated.

In a connection-oriented protocol, the decision about the route of a sequence of packets with the same source and destination addresses can be made only once, when the connection is established. Routers do not recalculate the route for each individual packet.

In a connectionless situation, the network layer protocol treats each packet independently, with each packet having no relationship to any other packet. The packets in a message may or may not travel the same path to their destination.

The IP protocol is a connectionless protocol. It is designed this way because IP, as an internetwork protocol, may have to deliver the packets through several heterogeneous networks. If IP were to be connection-oriented, all of the networks in the internet should also be connection-oriented, which is not the case.

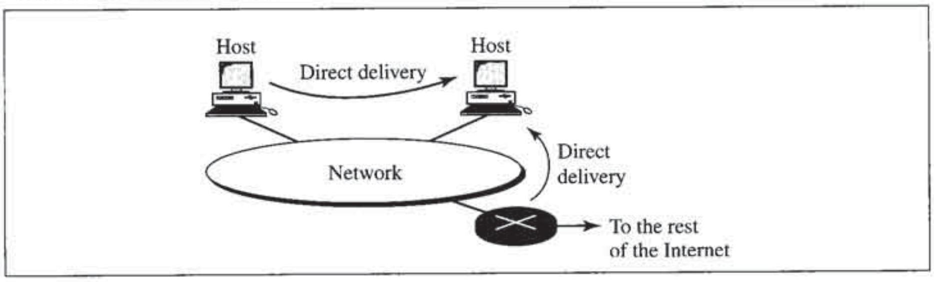
6.2 DIRECT VERSUS INDIRECT DELIVERY

The delivery of a packet to its final destination is accomplished using two different methods of delivery: direct and indirect.

Direct Delivery

In a **direct delivery,** the final destination of the packet is a host connected to the same physical network as the deliverer. Direct delivery occurs when the source and destination of the packet are located on the same physical network or if the delivery is between the last router and the destination host (see Figure 6.1).

Figure 6.1 *Direct delivery*



The sender can easily determine if the delivery is direct. It can extract the network address of the destination packet (setting the hostid part to all 0s) and compare this address with the addresses of the networks to which it is connected. If a match is found, the delivery is direct.

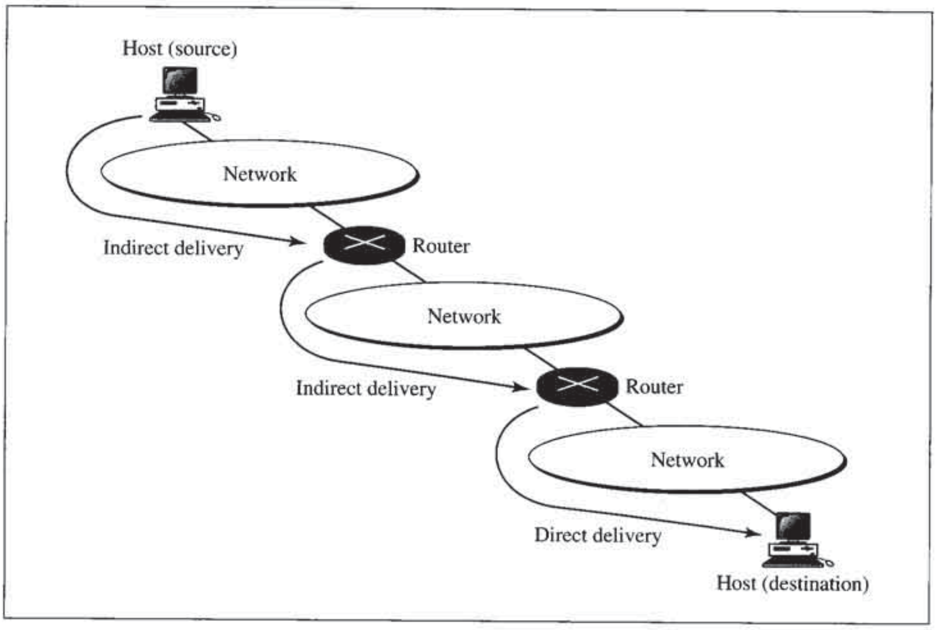
In direct delivery, the sender uses the destination IP address to find the destination physical address. The IP software then delivers the destination IP address with the destination physical address to the data link layer for actual delivery. This process is called *mapping the IP address to the physical address.* Although this mapping can be done by finding a match in a table, we will see in Chapter 8 that a protocol called address resolution protocol (ARP) dynamically maps an IP address to the corresponding physical address.

Indirect Delivery

If the destination host is not on the same network as the deliverer, the packet is delivered indirectly. In an **indirect delivery,** the packet goes from router to router until it reaches the one connected to the same physical network as its final destination (see Figure 6.2).

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

Figure 6.2 *Indirect delivery*



In an indirect delivery, the sender uses the destination IP address and a routing table to find the IP address of the next router to which the packet should be delivered. The sender then uses the ARP protocol to find the physical address of the next router. Note that in direct delivery, the address mapping is between the IP address of the final destination and the physical address of the final destination. In an indirect delivery, the address mapping is between the IP address of the next router and the physical address of the next router.

6.3 ROUTING METHODS

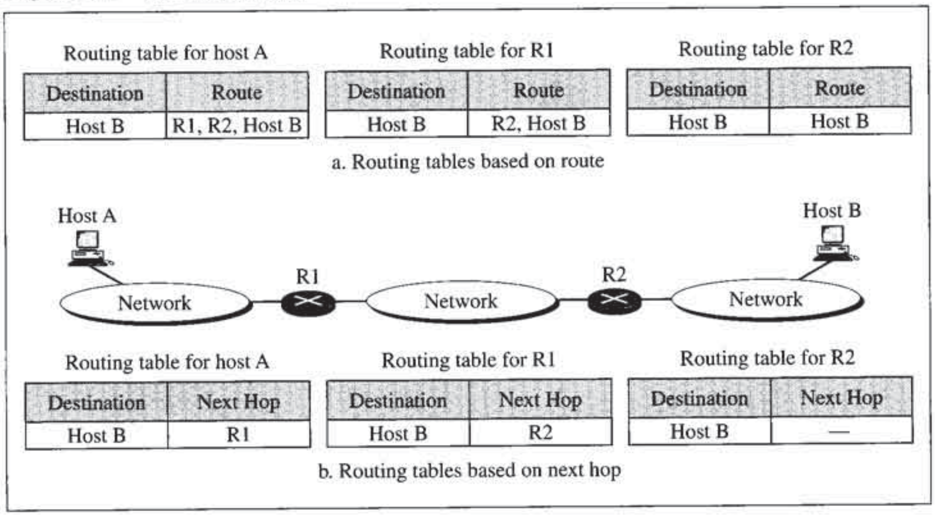
Routing requires a host or a router to have a routing table. When a host has a packet to send or when a router has received a packet to be forwarded, it looks at this table to find the route to the final destination. However, this simple solution is impossible today in an internetwork such as the Internet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

Several techniques can make the size of the routing table manageable and handle issues such as security. We will discuss these methods here.

Next-Hop Routing

One technique to reduce the contents of a routing table is called **next-hop routing.** In this technique, the routing table holds only the address of the next hop instead of holding information about the complete route. The entries of a routing table must be consistent with each other. Figure 6.3 shows how routing tables can be simplified using this technique.

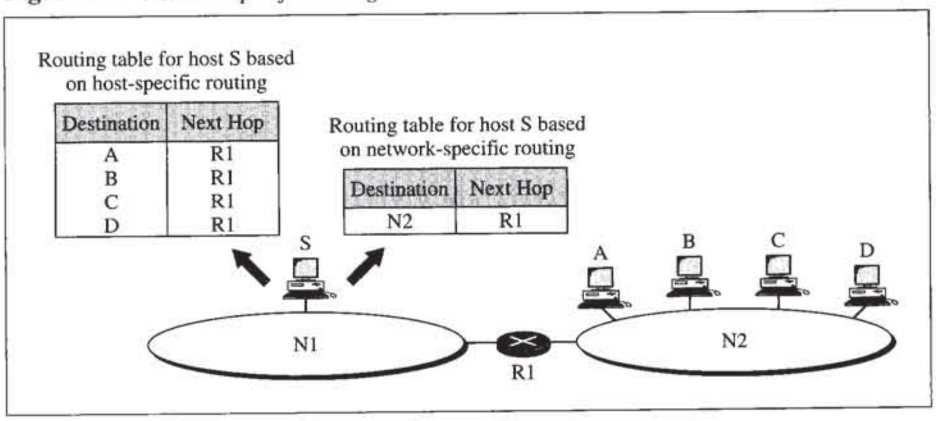
Figure 6.3 *Next-hop routing*



Network-Specific Routing

A second technique to reduce the routing table and simplify the searching process is called **network-specific routing.** Here, instead of having an entry for every host connected to the same physical network, we have only one entry to define the address of the network itself. In other words, we treat all hosts connected to the same network as one single entity. For example, if 1000 hosts are attached to the same network, only one entry exists in the routing table instead of 1000. Figure 6.4 shows the concept.

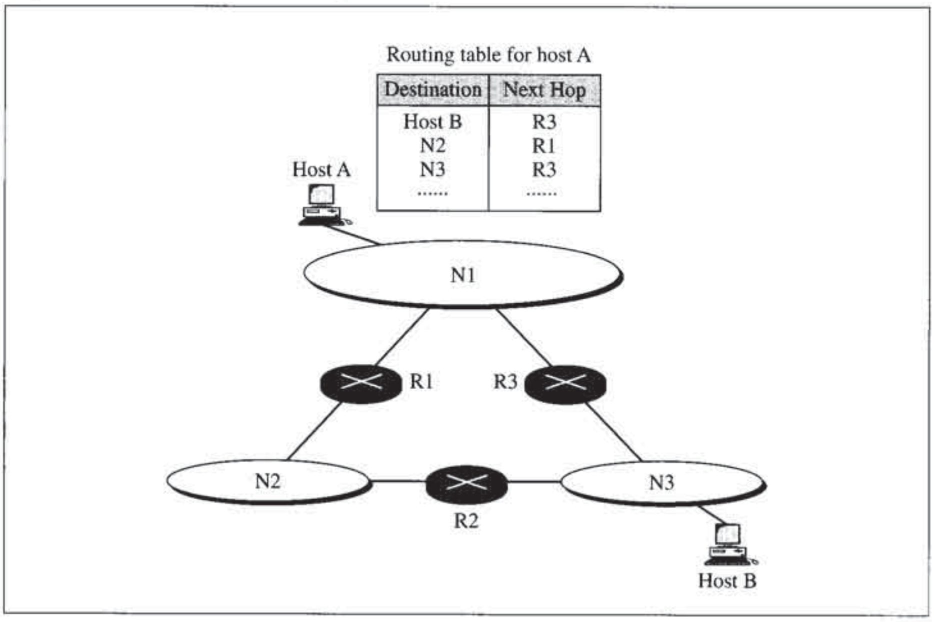
Figure 6.4 *Network-specific routing*



Host-Specific Routing

In **host-specific routing,** the destination host address is given in the routing table. The idea of host-specific routing is the inverse of network-specific routing. Here efficiency is sacrificed for other advantages: Although it is not efficient to put the host address in the routing table, there are occasions in which the administrator wants to have more control over routing. For example, in Figure 6.5 if the administrator wants all packets arriving for host B delivered to router R3 instead of R1, one single entry in the routing table of host A can explicitly define the route.

Figure 6.5 *Host-specific routing*



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

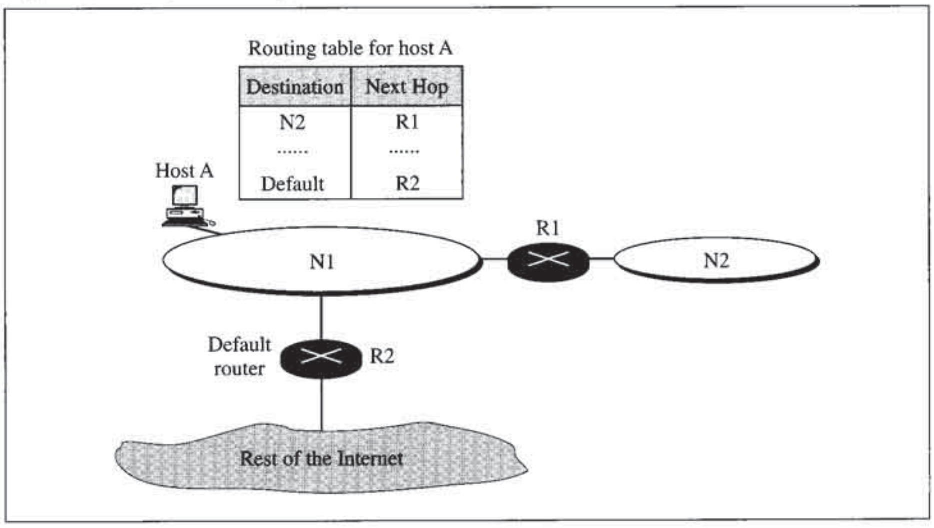
Default Routing

Another technique to simplify routing is **default routing.** In Figure 6.6 host A is connected to a network with two routers. Router R1 is used to route the packets to hosts connected to network N2. However, for the rest of the Internet, router R2 is used. So instead of listing all networks in the entire Internet, host A can just have one entry called the *default* (network address 0.0.0.0).

6.4 STATIC VERSUS DYNAMIC ROUTING

A host or a router keeps a routing table, with an entry for each destination, to route IP packets. The routing table can be either static or dynamic.

Figure 6.6 *Default routing*



Static Routing Table

A **static routing table** contains information entered manually. The administrator enters the route for each destination into the table. When a table is created, it cannot update automatically when there is a change in the Internet. The table must be manually altered by the administrator.

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

Dynamic Routing Table

A **dynamic routing table** is updated periodically using one of the dynamic routing protocols such as RIP, OSPF, or BGP (see Chapter 13). Whenever there is a change in the Internet, such as a shutdown of a router or breaking of a link, the dynamic routing protocols update all of the tables in the routers (and eventually in the host).

The routers in a big internet such as the Internet need to be updated dynamically for efficient delivery of the IP packets. We will discuss in detail the three dynamic routing protocols in Chapter 13.

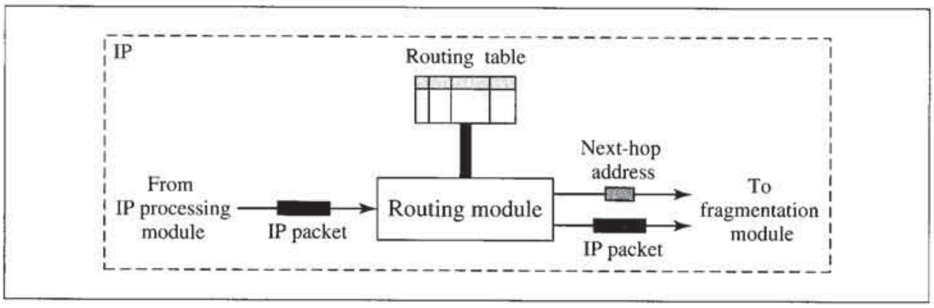
6.5 ROUTING TABLE AND ROUTING MODULE

In this section, we discuss a simplified routing module. The module, presented in pseudocode, basically shows how the router can extract the outgoing interface for the packet and the next-hop address if the delivery is indirect. The next-hop address is needed, as we will see in Chapter 7, to find the physical address of the next router to which the packet should be delivered.

When looking for the route, the router must first check for direct delivery, then host-specific delivery, then network-specific delivery, and finally default delivery. This hierarchical strategy can be implemented in the routing module or in the routing table. To make our routing module as simple as possible, we have used a routing table that is organized according to the above hierarchical scheme.

The module receives an IP packet from the IP processing module (see Chapter 8). The routing module consults the routing table to find the best route for the packet. After the route is found, the packet is sent along with the next-hop address to the fragmentation module (see Chapter 8), which makes a decision on fragmentation. See Figure 6.7.

Figure 6.7 *Routing module and routing table*



Routing Table

As mentioned previously, our routing table is organized in a hierarchical scheme with direct-delivery entries first, host-specific delivery entries next, network-specific entries third, and the default delivery entry last.

The routing table usually has these seven fields: mask, destination address, next-hop address, flags, reference count, use, and interface (see Figure 6.8).

Figure 6.8 *Fields in a routing table*

Mask	Destination address	Next-hop address	Flags	Reference count	Use	Interface
255.0.0.0	124.0.0.0	145.6.7.23	UG	4	20	m2
....	....		....	....	....	....
....	....	....	....	....	....	....

- **Mask.** This field defines the mask applied to the destination IP address of the packet to find the network or subnetwork address of the destination. In host-specific routing, the mask is 255.255.255.255. In default routing, the mask is 0.0.0.0. In an unsubnetted network, the mask is the default mask (255.0.0.0, 255.255.0.0, or 255.255.255.0 for class A, B, or C, respectively).

- **Destination address.** This field defines either the destination host address (host-specific address) or the destination network address (network-specific) address. A host-specific destination address gives the full destination address, netid and hostid. A network-specific address gives only the address of the network to which the destination entity is connected. The netid is specific, but the hostid is all 0s.
- **Next-hop address.** This field defines the address of the next-hop router to which the packet is delivered.
- **Flags.** This field defines up to five flags. Flags are on/off switches that signify either presence or absence. The five flags are U (up), G (gateway), H (host-specific), D (added by redirection), and M (modified by redirection).
  a. **U (Up).** The U flag indicates the router is up and running. If this flag is not present, it means that the router is down. The packet cannot be forwarded and is discarded.
  b. **G (Gateway).** The G flag means that the destination is in another network. The packet should be delivered to the next-hop router for delivery (indirect delivery). When this flag is missing, it means the destination is in this network (direct delivery).
  c. **H (Host-specific).** The H flag indicates that the entry in the destination field is a host-specific address. When it is missing, it means that the address is only the network address of the destination.
  d. **D (Added by redirection).** The D flag indicates that routing information for this destination has been added to the host routing table by a redirection message from ICMP. We will discuss redirection and the ICMP protocol in Chapter 9.
  e. **M (Modified by redirection).** The M flag indicates that the routing information for this destination has been modified by a redirection message from ICMP. We will discuss redirection and the ICMP protocol in Chapter 9.
- **Reference count.** This field gives the number of users that are using this route at any moment. For example, if five people at the same time are connecting to the same host from this router, the value of this column is 5.
- **Use.** This field shows the number of packets transmitted through this router for the corresponding destination.
- **Interface.** This field shows the name of the interface.

Routing Module

The routing module receives an IP packet from the IP processing module (see Chapter 8). In our example, the routing module goes from entry to entry trying to find a match. When it finds a match, it quits. Because the routing table is hierarchically organized, it is guaranteed that the module first looks for a direct-delivery match. If no match is found, the module looks for a host-specific delivery, and so on.

```
Routing Module

1. For each entry in the routing table
   1. Apply the mask to packet destination address
   2. If (the result matches the value in the destination field)
      1. If (the G flag is present)
         1. Use the next-hop entry in the table as next-hop address
      2. If (the G flag is missing)
         1. Use packet destination address (direct delivery)
      3. Send packet to fragmentation module with next-hop address
      4. Stop
2. If no match is found, send an ICMP error message
3. Stop
```

Some Examples

In this section we give some examples of routing. Figure 6.9 is used for Examples 1, 2, and 3. These three examples also use the routing table shown in Table 6.1.

Figure 6.9 *Configuration for routing examples*

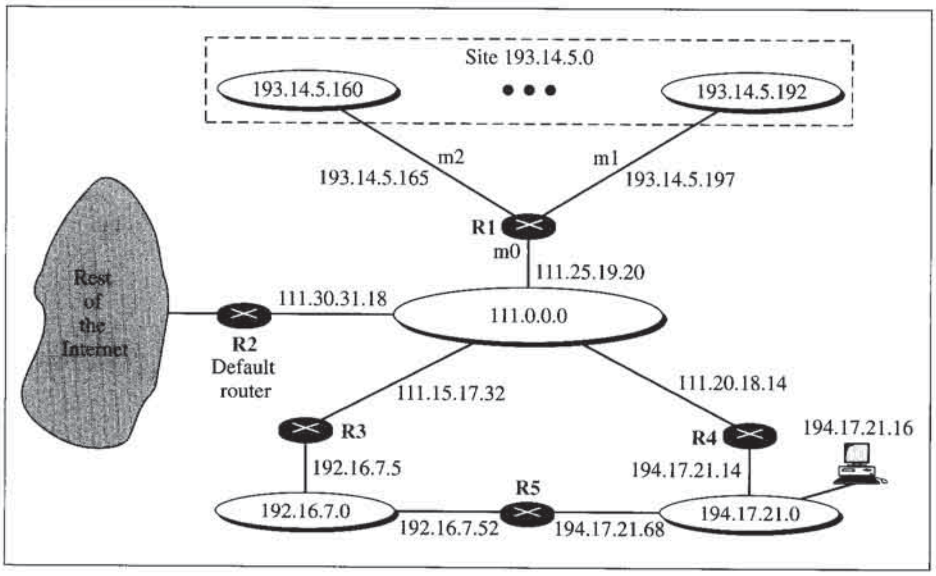


Table 6.1 *Routing table for router R1 in Figure 6.9*

Mask	*Destination*	*Next Hop*	*F.*	*R.C.*	*U.*	*I.*
255.0.0.0	111.0.0.0	-	U	0	0	m0
255.255.255.224	193.14.5.160	-	U	0	0	m2
255.255.255.224	193.14.5.192	-	U	0	0	m1
....	....	....	....	....	...	...
....	....	....	....	....	...	...
....	....	....	....	....	...	...
255.255.255.255	194.17.21.16	111.20.18.14	UGH	0	0	m0
255.255.255.0	192.16.7.0	111.15.17.32	UG	0	0	m0
255.255.255.0	194.17.21.0	111.20.18.14	UG	0	0	m0
0.0.0.0	0.0.0.0	111.30.31.18	UG	0	0	m0

Example 1

Router R1 receives 500 packets for destination 192.16.7.14; the algorithm applies the masks row by row to the destination address until a match (with the value in the second column) is found:

1. Direct delivery
   a. 192.16.7.14 & 255.0.0.0 ➡ 192.0.0.0 no match
   b. 192.16.7.14 & 255.255.255.224 ➡ 192.16.7.0 no match
   c. 192.16.7.14 & 255.255.255.224 ➡ 192.16.7.0 no match
2. Host-specific
   a. 192.16.7.14 & 255.255.255.255 ➡ 192.16.7.14 no match
3. Network-specific
   a. 192.16.7.14 & 255.255.255.0 ➡ 192.16.7.0 **match**

The router sends the packet through interface m0 along with the next-hop IP address (111.15.17.32) to the fragmentation module for further processing. It increments the use field by 500 and the reference count field by 1.

Example 2

Router R1 receives 100 packets for destination 193.14.5.176; the algorithm applies the masks row by row to the destination address until a match is found:

1. Direct delivery
   a. 193.14.5.176 & 255.0.0.0 ➡ 193.0.0.0 no match
   b. 193.14.5.176 & 255.255.255.224 ➡ 193.14.5.160 **match**

The router sends the packet through interface m2 along with the destination IP address (193.14.5.176) to the fragmentation module for further processing. It increments the use field by 100 and the reference count field by 1.

Example 3

Router R1 receives 20 packets for destination 200.34.12.34; the algorithm applies the masks row by row to the destination address until a match is found:

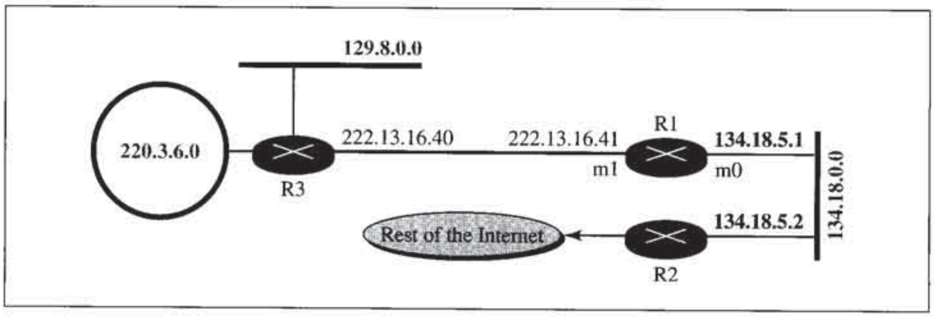
1. Direct delivery
   a. 200.34.12.34 & 255.0.0.0 ➡ 200.0.0.0 no match
   b. 200.34.12.34 & 255.255.255.224 ➡ 200.34.12.32 no match
   c. 200.34.12.34 & 255.255.255.224 ➡ 200.34.12.32 no match
2. Host-specific
   a. 200.34.12.34 & 255.255.255.255 ➡ 200.34.12.34 no match
3. Network-specific
   a. 200.34.12.34 & 255.255.255.0 ➡ 200.34.12.0 no match
   b. 200.34.12.34 & 255.255.255.0 ➡ 200.34.12.0 no match
4. Default
   a. 200.34.12.34 & 0.0.0.0 ➡ 0.0.0.0. **match**

The router sends the packet through interface m0 along with the next-hop IP address (111.30.31.18) to the fragmentation module for further processing. It increments the use field by 20 and the reference count field by 1.

Example 4

Make the routing table for router R1 in Figure 6.10.

Figure 6.10 *Topology for Example 4*



Solution

We know that there are three explicit destination networks, two class B and one class C with no subnetting (default masks). There is also one access to the rest of the Internet (default route). This means that our routing table has four rows. The routing table is shown in Table 6.2. The interface to the network 134.18.0.0 is m0 and there is no next-hop address. Access to the rest of the Internet is through interface m0 and the next-hop address is 134.18.5.2. Access to network 129.8.0.0 is through interface m1 and the next-hop address is 222.13.16.40. Access to network 220.3.6.0 is through interface m1 and the next-hop address is 222.13.16.40.

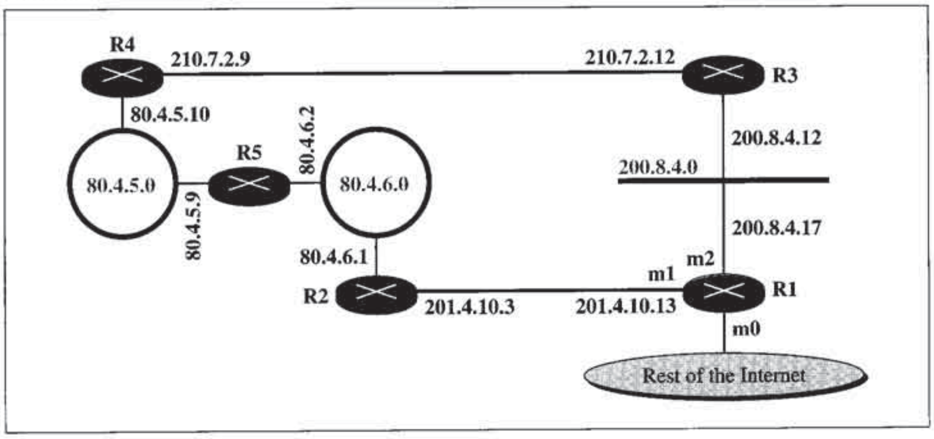
Table 6.2 *Routing table for Example 4*

Mask	*Destination*	*Next Hop*	*F.*	*R.C.*	*U.*	*I.*
255.255.0.0	134.18.0.0	---	U	0	0	m0
255.255.0.0	129.8.0.0	222.13.16.40	UG	0	0	m1
255.255.255.0	220.3.6.0	222.13.16.40	UG	0	0	m1
0.0.0.0	0.0.0.0	134.18.5.2	UG	0	0	m0

Example 5

Make the routing table for router R1 in Figure 6.11.

Figure 6.11 *Topology for Example 5*



Solution

We know that there are five networks here, but two of them are point-to-point with no hosts and need not be in the routing table. The other three networks must be in the table. There is also an entry for the default. Examination of the figure shows that some information is missing. For example, we do not know the IP address of the default router. Also, there are two paths to networks 80.4.5.0 (and 80.4.6.0); one is through router R2 and one is through router R3. We do not know which route is optimal; this subject is covered in Chapter 13. For the moment, we enter both paths in the routing table (see Table 6.3).

Table 6.3 *Routing table for Example 5*

Mask	*Destination*	*Next Hop*	*F.*	*R.C.*	*U.*	*I.*
255.255.255.0	200.8.4.0	----	U	0	0	m2
255.255.255.0	80.4.5.0	201.4.10.3 or 200.8.4.12	UG	0	0	m1 or m2
255.255.255.0	80.4.6.0	201.4.10.3 or 200.4.8.12	UG	0	0	m1 or m2
0.0.0.0	0.0.0.0	????????????	UG	0	0	m0

Example 6

The routing table for router R1 is given in Table 6.4. Draw its topology.

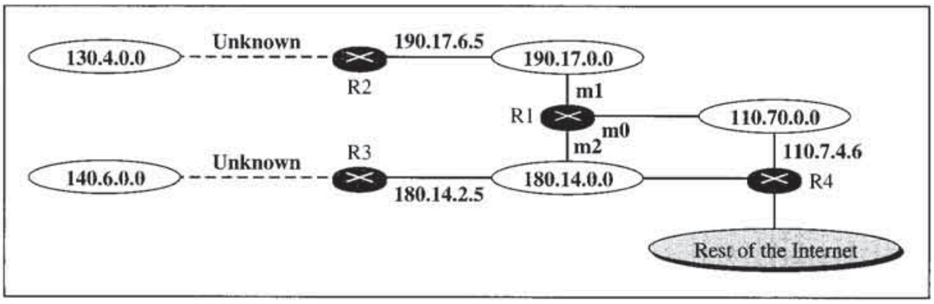
Table 6.4 *Routing table for Example 6*

Mask	*Destination*	*Next Hop*	*F.*	*R.C.*	*U.*	*I.*
255.255.0.0	110.70.0.0	-	U	0	0	m0
255.255.0.0	180.14.0.0	-	U	0	0	m2
255.255.0.0	190.17.0.0	-	U	0	0	m1
255.255.0.0	130.4.0.0	190.17.6.5	UG	0	0	m1
255.255.0.0	140.6.0.0	180.14.2.5	UG	0	0	m2
0.0.0.0	0.0.0.0	110.70.4.6	UG	0	0	m0

Solution

We know some facts but don't have all of them for a definite topology. We know that there are three networks directly connected to router R1. We know that there are two networks indirectly connected to R1. There must be at least three other routers involved (see next-hop column). We know to which networks these routers are connected by looking at their IP addresses. So we can put them at their appropriate places. We know that one router is connected to the rest of the Internet (it is the default router). But there is some missing information.We do not know if network 130.4.0.0 is directly connected to router R2 or through a point-to-point network (WAN) and another router. We do not know if network 140.6.0.0 is connected to router R3 directly or through a point-to-point network (WAN) and another router. Point-to-point networks normally do not have an entry in the routing table because no hosts are connected to them. Figure 6.12 shows our guessed topology.

Figure 6.12 *Guessed topology for Example 6*



6.6 CLASSLESS ADDRESSING: CIDR

So far, the discussion on routing tables concentrated on classful addressing. Now we need to consider classless addressing and Classless InterDomain Routing (CIDR). The shift to classless addressing requires changes to the routing table organization and routing algorithms.

Routing Table Size

When we use classful addressing, there is only one entry in the routing table for each site outside the organization. The entry defines the site even if that site is subnetted. When a packet arrives at the router, the router checks the corresponding entry and forwards the packet accordingly.

When we use classless addressing, the number of entries in the router's table can either decrease or increase. It can decrease if the block of addresses assigned to an organization is larger than the block in classful addressing. For example, instead of having four entries for an organization that creates a supernet from four class C blocks, we can have one entry in classless routing.

It is more likely, however, that the number of routing table entries will increase. This is because the intent of classless addressing is to divide up the blocks of class A and class B addresses. For example, instead of assigning over 16 million addresses to just one organization, the addresses can be portioned out to many many organizations. The problem is that whereas there was just one routing table entry for the class A address, now there are many entries in classless addressing. For example, if a class B block (over 64,000 addresses) is divided up between 60 organizations, there are 60 routing table entries where before there was just one.

Hierarchical Routing

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

Let us take the case of a local ISP. A local ISP can be assigned a single, but large block of addresses with a certain prefix length. The local ISP can divide this block into smaller blocks of different sizes, and assign these to individual users and organizations, both large and small. If the block assigned to the local ISP is A.B.C.D/*N*, the ISP can create blocks of E.F.G.H/*M*, where *M* may vary for each customer and is greater than *N*.

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

Geographical Routing

To decrease the size of the routing table even further, we need to extend hierarchical routing to include geographical routing. We must divide the entire address space into a few large blocks. We assign a block to North America, a block to Europe, a block to Asia, a block to Africa, and so on. The routers of ISPs outside of Europe will have only one entry for packets to Europe in their routing tables. The routers of ISPs outside of North America will have only one entry for packets to North America in their routing tables. And so on. Part of this idea has already been implemented for class C addressing. But, for real efficiency, all of classes A and B need to be recycled and reassigned.

Routing Table Search Algorithms

The algorithms that search the routing table must also be changed to make classless routing more efficient. This includes the algorithms that update routing algorithms. We will discuss this updating issue in Chapter 13.

Searching in Classful Addressing

In classful addressing, the routing table is organized as a list. However, to make searching easier, the routing table can be divided into three buckets (areas), one for each class. When the packet arrives, the router applies the default mask (which is inherent in the address itself) to find the corresponding bucket (A, B, or C). The bucket then searches the corresponding bucket instead of the whole table.

> In classful addressing, each address has self-contained information that facilitates routing table searching.

Searching in Classless Addressing

In classless addressing, we can also use buckets; specifically, 32 buckets, one for each prefix length. However, the problem is that there is no self-contained information in the destination address to help the router decide which bucket to search. The simplest, but not the most efficient, method is called the **longest match.** The router first tries to use the longest prefix (/32). If the destination address is found in this bucket, the search is complete (this bucket is for host-specific routing). If the address is not found, the bucket for the next prefix (/31) is searched. And so on. It is obvious that this type of search takes a long time; on average, 16 buckets must be searched.

The solution is to change the data structure used for searching. Instead of using a list, use other data structures (such as a tree or a binary tree). One of the candidates is a trie (a special kind of tree). However, this discussion is beyond the scope of this book.

> In classless addressing, there is no self-contained information in the destination address to facilitate routing table searching.