Chapter 3 Three

Routing and Forwarding Processes

This chapter covers generic questions about the routing and forwarding processes. Operation of datagram networks, basics of routing table construction and packet-forwarding processes, and details of router operation in classful and classless environments are discussed.

For a long time, the term *routing* has been used interchangeably with the term *forwarding*. However, it is common today to differentiate these two notions.

In this book, as well as in the industry, the term *routing* is used to describe the functionality performed by the control software of the routers. This includes routing table maintenance, processing of static routes, dynamic routing protocols, and so on. The term *forwarding* is used to refer to the process of moving transit packets from one interface to another. The forwarding process includes looking through the forwarding table, making the forwarding decision, and sending the packet out of an interface. It is essential to understand the difference between the two processes, and this book emphasizes this in several places. This chapter discusses the concepts of both routing and forwarding processes.

3.1 Packet-Switched Technologies

The two main types of packet-switched technologies used in networks today are *virtual-circuit* (VC) and *datagram* networks. The difference between the two is in the data-delivery mechanism used on the network layer. Both types belong to the packet-switching group of telecommunication technologies, so the intermediate network devices decide where to send the packets carrying user data. The criteria taken into consideration while making this decision, as well as the procedures performed by the hosts and intermediate devices, depend on the type of the network.

Networks that use the virtual-circuit strategy can be compared to common circuit-switching telephone networks. When users of a telephone network need to communicate

information—a voice or fax message—they explicitly establish a connection with the remote end by dialing a number. The network reacts by selecting and connecting the circuits, routing the call and allocating available resources for the connection. When the connection is established, it is used for the information transfer. When data is sent through the connection, the network does not perform any additional routing tasks for this connection, because the data flows along an already defined path.

Operation of VC-oriented packet-switching networks is similar. When one host needs to send some data to another, a connection through the network is opened before the transmission starts. In opening the connection, the source device specifies the called address—similar to dialing a telephone number—which is used by the network to route the call and to allocate bandwidth. When the network receives the call request, the network devices consult their *routing tables* to determine how the call-establishment request should be directed. The routing table is used only during the call setup. As the call traverses the network, the intermediate network devices create virtual circuits on the physical links and update their *switching tables*. The entries in the switching tables contain information about the virtual-circuit cross-connections. This information is used during data transfer. After the connection is established, the two hosts do not specify source or destination addresses in the data packets but rather include the identifier of the virtual circuit. This identifier is used to index within the switching table and to find the outbound interface and the new VC identifier. If either side decides to disconnect, it sends an explicit command to the network device it is connected to, which in turn requests other devices along the connection to release the resources.

Datagram networks operate on a different concept, which can be compared to that of the postal service. People sending letters to each other don't have to establish a connection beforehand. They simply provide proper address information and drop the letter at their local post office. The post office sends the letter to another post office closer to the destination. The letter traverses a set of post offices until it reaches the one local to the addressee.

In a datagram network, a host constructs a packet that includes the source and destination addresses. The packet is sent to the nearest network device, which in turn passes it to another device, closer to the destination. When it receives the packet, a network device performs a *routing decision*, also called a *forwarding decision*, based on the destination address of the packet and the contents of its routing table. The decision is made for every packet in every intermediate device.

This is the main difference between a datagram network and a virtual-circuit network. Devices in VC-oriented networks have to route only one packet per connection, the call setup; all other packets that belong to the same VC are switched using the switching table. Datagram network devices have to route every packet; each network device looks up its routing table and decides where to send the packet.

Both technologies have their pros and cons. For example, in virtual-circuit network devices, the code responsible for data transfer is very simple and efficient. Using only the

switching table, it needs fewer CPU and RAM read cycles for data processing than do datagram network devices. At the same time, rerouting of already established connections, necessary in case of a topology change, can be a problem in virtual-circuit networks. Datagram networks adapt much faster, as every packet is routed at each hop. The adaptation, or convergence, time depends on how long it takes to propagate correct routing information. In most situations, it depends on the convergence time of the dynamic routing protocol used in the network. With modern protocols, such as OSPF and EIGRP, this time can be measured in seconds.

3.2 Router Operation Overview

As you know, every IP host in a network is normally configured with not only its own IP address and mask but also the IP address of the default gateway (see Figure 3-1). If the host needs to send an IP packet to a destination address that does not belong to a subnet the host is directly attached to, the host passes the packet to the default gateway (router).

A common misunderstanding is how the address of the default gateway is used. People tend to incorrectly think that when a packet is sent to the default router, the host sets the destination address in the IP packet to the configured default router address. However, the router would then consider the packet addressed to itself and would not forward it any further. Why configure the default gateway's IP address, then? The answer is that the host uses the *Address Resolution Protocol* (ARP) to find the *Media Access Control* (MAC) address of the specified router. Having acquired the router's MAC address, the host sends the packets directly to it as data link unicast submissions.

What happens when a router receives a packet on one of its interfaces? The first thing to remember is that normally, routers do not check the source IP address of the packet when it is received. It seems obvious. However, people are sometimes very surprised when they see, for example, packets coming from an interface attached to subnet 195.100.10.0 with the source address set to, say, 130.10.86.15.

> As we will see in Chapter 5, however, Cisco routers can be explicitly configured to perform so-called unicast Reverse-Path Forwarding (RPF) check, whereby the router does verify the source address in the packets to prevent denial-of-service attacks.

Routers never change the source and the destination addresses in IP packets, except for *Network Address Translation* (NAT), which is not considered in this book. Routers send packets to each other by setting correct data link layer addresses—for example, MAC addresses for Ethernet or *Data Link Connection Identifier* (DLCI) for Frame Relay—in the data link frames or just pushing them through point-to-point links, using associated encapsulation mechanisms. If they changed the source and destination IP addresses, routers would lose information about where the packets were coming from

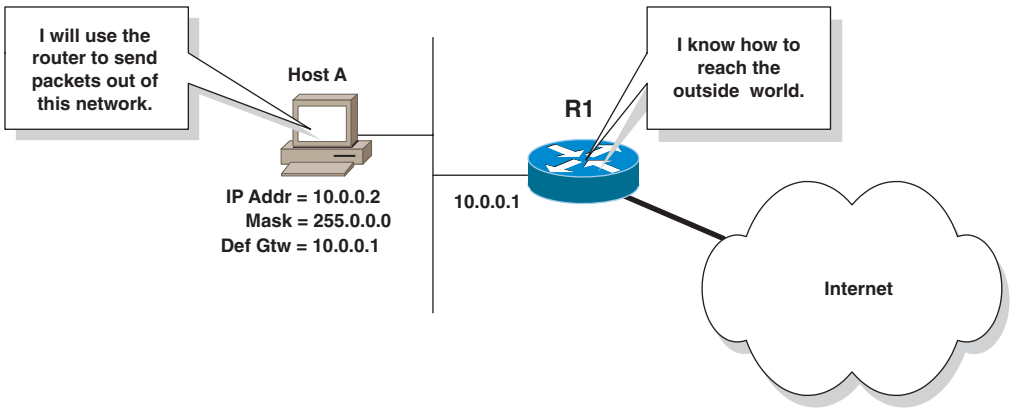

Figure 3-1. *Use of default gateway*

and going to. In our example, a packet with a source address of 130.10.86.15 could be originated by a host outside the network and forwarded by another router on subnet 195.100.10.0. The intermediate routers won't change the packet's IP addresses but instead just pass the packet to the next closest neighbor toward the destination address.

When a packet is received, the router checks its validity and determines whether the packet must be delivered locally—it is addressed to the router itself—or forwarded further. If the packet must be forwarded, the router makes the routing decision and determines the outbound interface and the IP address of the router that should be the next hop in the path, if the destination network is not directly attached.

Consider a simple example. Suppose that a router is connected to two networks—10.0.0.0 and 20.0.0.0—as illustrated in Figure 3-2. Host A on the first segment sends an IP packet to host B on the second segment. Host A passes the packet to the router by specifying the router's MAC address as the destination address in the Ethernet frame. (The destination IP address is the IP address of B.)

When it receives the frame, R1 examines the IP packet and uses its routing table to decide where to forward the packet. The table contains information in the form "to reach hosts on network N, use interface X and next hop Y." Such a combination of routing parameters is called a *route*. How does the router know where the networks are, though?

Part of every router's configuration task is assignment of IP addresses to the router's interfaces. A router therefore functions like a normal IP host on each network to which it is connected. Even if it never originated IP packets itself, a router would still need this information to answer the ARP requests sent by end nodes while trying to find the MAC address of the default router or by neighboring routers looking for the MAC address of the next hop.

Because a router's interfaces are configured with IP addresses and corresponding address masks, the router can derive information about the networks connected to its

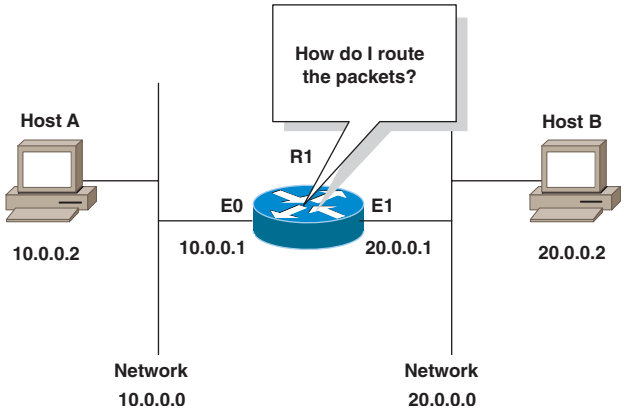

Figure 3-2. *Two segments connected by a router*

interfaces by applying address masks to the associated addresses. That is the way routers obtain their startup information about directly connected networks and put this information into the routing tables. In our example, the routing table of R1 would look like the following:

```
Network 10.0.0.0 is directly connected to interface Ethernet 0
Network 20.0.0.0 is directly connected to interface Ethernet 1
```

This table contains enough information to route the packet from host A to host B. The router just takes the destination address from the IP packet header and looks through the table. Having found the information about network 20.0.0.0, the router understands that the packet destined for a host on this network should be delivered on interface Ethernet 1. The decision is made. Now the router has to encapsulate the IP packet into an Ethernet frame and send it to host B. If this is the first time the router is sending a packet to this host, the router sends an ARP request, asking for B's MAC address. Otherwise, the router uses its ARP cache. This example is quite simple, as both networks are directly connected.

Now look at a network constructed of several routers (Figure 3-3). Every router in the network has information only about directly attached networks:

- R1:

```
Network 10.0.0.0 is directly connected to the interface Ethernet 0
Network 20.0.0.0 is directly connected to the interface Ethernet 1
```

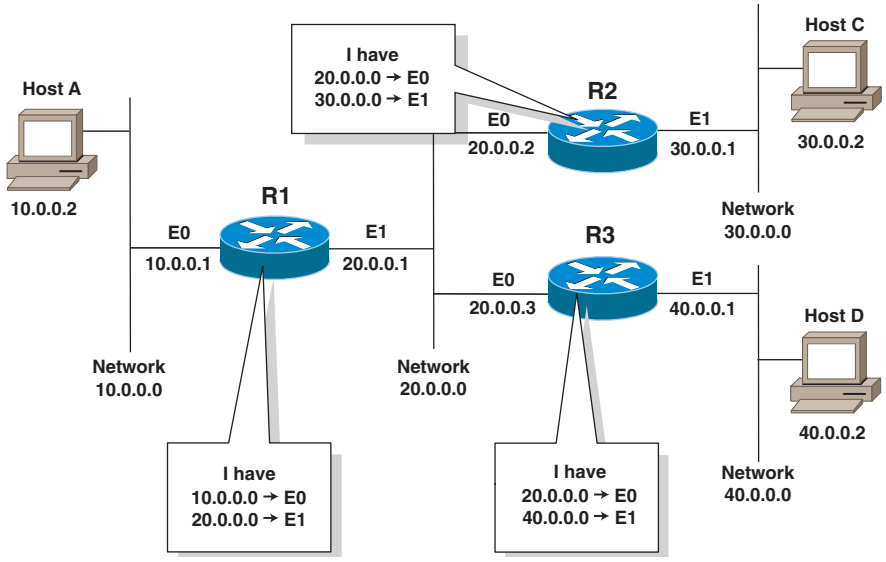

Figure 3-3. *More complex routed network*

- R2:

```
Network 20.0.0.0 is directly connected to the interface Ethernet 0
Network 30.0.0.0 is directly connected to the interface Ethernet 1
```

- R3:

```
Network 20.0.0.0 is directly connected to the interface Ethernet 0
Network 40.0.0.0 is directly connected to the interface Ethernet 1
```

Now host A from network 10.0.0.0 sends an IP packet to host C on network 30.0.0.0. When it receives the packet, R1 cannot make the forwarding decision, because it has no information about destination network 30.0.0.0. R1 will drop the packet and send an ICMP "Destination Unreachable" message to host A. What information would R1 need in its routing table to successfully route the packet to host C? There should obviously be a line saying that network 30.0.0.0 is reachable via router R2. The routing table of R1 would have to look like the following.

```
Network 10.0.0.0 is directly connected to the interface Ethernet 0
Network 20.0.0.0 is directly connected to the interface Ethernet 1
Network 30.0.0.0 is accessible via 20.0.0.2
```

With this information, the router would make its forwarding decision as follows.

1. The packet is destined to host 30.0.0.2 (host C).
2. Look through the routing table for information about address 30.0.0.2.
3. Address 30.0.0.2 belongs to network 30.0.0.0 that is accessible via host 20.0.0.2.
4. Look through the routing table for information about address 20.0.0.2.
5. Network 20.0.0.2 belongs to network 20.0.0.0 that is directly connected to interface Ethernet 1.
6. Send the packet through the Ethernet 1 interface, using R2's MAC address as the destination MAC address.

As you see, the router performs recursive table lookup, trying to find a route to the next-hop address, specified in the last route in the table. If there were another router, R4, on network 30.0.0.0 and connected to network 50.0.0.0, router R1 would have the following routing table:

```
Network 10.0.0.0 is directly connected to the interface Ethernet 0
Network 20.0.0.0 is directly connected to the interface Ethernet 1
Network 30.0.0.0 is accessible via 20.0.0.2
Network 50.0.0.0 is accessible via 30.0.0.5         (router R4's IP address)
```

The steps taken by the router in this case would be the same: "Find information about the destination network; if it goes through another network, find information about that one, too." The router would continue looking through the table recursively until it found a reference to a router's address that belonged to a directly connected network or it realized that there was no route for the address.

Following are some rules of thumb about routing in datagram networks. (IP networks belong to the datagram network group.) Knowing these basic principles is required for network maintenance and troubleshooting.

- *Every router makes its decision alone, based on the information it has in its own routing table*. When making the routing decision, a router can use only information in its own routing table. There is no way for a router to check whether its neighbors are going to make a consistent decision. Routers route packets according to the information they have in the routing tables at a particular instance. When it forwards a packet to the next router, a router assumes that the next router will do the same: make its decision according to the information in its own routing table. Only consistent routing information can guarantee a consistent forwarding decision throughout the network.

- *The fact that one router has certain information in its routing table does not mean that other routers have the same information*. Even if the first-hop router—the router nearest to the source—has required information about a remote network, other routers on the way to the destination may have no information about it. Therefore, even if the first-hop router forwards a packet successfully, the next router may drop the packet if it doesn't have enough routing information to forward it.
- *Routing information about a path from one network to another does not provide routing information about the reverse, or return, path*. Even though all routers along the way to a destination have information about the destination network, the remote routers may have no information about how to route packets coming back. In the example, if host C on network 30.0.0.0 sent a reply to host A on network 10.0.0.0, router R2 would need additional information about how to reach network 10.0.0.0. If it had no information about this network, R2 would have to drop the packet.

According to these rules, the administrator needs to make sure that all routers in a network have adequate and consistent information about every network that might be involved in the communication process.

Lack of routing information about a destination network is not the only reason for a router to drop a packet. A router can also drop packets because of output queue overflow or because of a lack of CPU time needed for the router to take packets out of the input queues. A packet is also dropped when the value of its *Time-to-Live* (TTL) field reaches 0; each router decrements it by1. This is a protective measure introduced to make sure that even in the presence of temporary or permanent routing loops, the network does not accumulate—forward endlessly—packets destined for the networks for which the loops are experienced. Another reason for a packet drop is inability to fragment an IP packet while trying to send it through one of the router's interfaces.

Normally, every router's interface is assigned a value that specifies the maximum size of a data block that can be sent over it. This value is called *maximum transmission unit* (MTU) and is usually specific for a given media type. For example, the default MTU for Ethernet and serial interfaces in Cisco routers is 1,500 bytes; for the 16Mbps Token Ring, it is 8,136 bytes. When it is about to send an IP packet over an interface, a router checks whether the packet fits into the interface MTU. If the packet is bigger than the MTU, the router breaks the packet into pieces that fit into it and sends them as separate IP packets. This process is called *IP packet fragmentation*. Routers can fragment an IP packet if necessary unless it has the *do not fragment* (DF) bit set in the header. When this bit is set and a router sees that the packet must be fragmented, the router drops the packet and sends an ICMP "Destination Unreachable" message with the code field set to "Fragmentation needed and DF set" to the originator.

The difference between an IP packet and an IP datagram is that hosts always send datagrams, which can be fragmented into several IP packets. Therefore, any IP packet can be either a whole IP datagram or a fragment of it. IP packets can be further fragmented if they need to be sent over a link with even smaller MTUs.

If a datagram is fragmented while going through the network, the receiving host performs IP datagram reassembly. Routers do not reassemble IP datagrams from IP packets not destined for themselves, for several reasons. First, it would add extra delays in routing: A router would have to wait until all fragments of a given datagram came to it. Second, the router would need to store all fragments of all datagrams before reassembly. (Imagine an Internet core router doing this.) Third, and maybe most important, because routers perform load sharing—send packets to the same destination along parallel paths—and because IP packets are sometimes dropped on their way through the network, a router may never receive all fragments of a datagram.

3.3 Routing Information Sources

In Figure 3-3, which shows three routers connecting four networks, router R1 would need additional information to be able to route to network 30.0.0.0. A line in R1's routing table would say that the network is accessible via router R2. How does this piece of information get into the routing table?

Routing information can get into the routing table in only two ways. As in any computer, information in a router can be either entered manually or gathered automatically. The administrator can investigate the network, sketch the topology, formalize the information—that is, represent it in the form "For router R1, network N2 is accessible via router R3"—and feed it to every router in the network. This is what happened at the beginning of the network world: Administrators used to watch the networks and edit the routing tables manually whenever the topology changed. This type of routing information, provided and entered into routers by administrators, is called *static*. Static routes reflect the administrators' knowledge of the topology and the policy they want to apply to the traffic going to a certain destination.

The drawbacks of static routing are obvious. Static routing generally cannot adapt to changes in topology or to load and error rates of channels. To have this adaptation, administrators would need to constantly keep an eye on the network and change the routing tables in real time. It may not seem difficult for a network with five routers, but imagine what would happen if this technique were used to manage a network consisting of hundreds of routers and channels connecting them. It would be impossible. This is the reason *dynamic routing protocols*—automatic distributed network discovery and route calculation algorithms—were proposed, designed, and implemented.

The function of dynamic routing protocols is to exchange information about the networks routers know how to reach. Routing protocols provide routers with real-time

information about where the networks are, which path is better for accessing a given network, which parallel paths can be used to balance the load among multiple channels, and so on. If routers have this information, they can maintain adequate routing tables.

3.4 Static Routing

Although dynamic routing protocols are widely used now, many networks still use static routing. It is effective when networks have stable topology and don't need real-time adaptation. In static routing, the administrator configures routers with all the information necessary for successful packet forwarding. The administrator constructs the routing table in every router by putting in the entries for every network that could be a destination.

In connecting a branch office to the central site, for example, static routing does everything needed for hosts on the central and remote networks to communicate (Figure 3-4). The administrator can just add static routes to the routers in the central and the remote offices, saying that respective networks are accessible via the corresponding routers. This arrangement doesn't demand a dynamic routing protocol, as neither site cares about the specific subnets behind the routers and there's no redundancy in the topology. Router R1 needs to know only that it must send packets to router R2 to reach any host in the remote office, no matter what IP subnet the host is in. R2, in turn, needs to know only that all hosts in the central office's network can be reached via router R1. Note, however, that if more routers were in the central or the remote network, they would also need to be configured with corresponding static routes.

The next example, closer to real life, is a hub-and-spoke topology, with multiple remote branches connected to the central site, which is connected to another, newly acquired company's network (see Figure 3-5). The addressing scheme used in this example is a bit more realistic: Remote offices are assigned subnet numbers from major network 10.0.0.0. The central site also contains subnet 10.9.0.0, to which the global resources are connected.

As you see in Figure 3-5, every router is configured with static routes that define which router should be used as the next hop to move the packet to the destination network. The routers in the remote offices have information that their local subnets are reachable via the interfaces connected to them and that all other parts of network 10.0.0.0 can be accessed via router R1. This route is called a *summary route*. Use of a summary route means that it doesn't matter which specific subnets are there, and packets destined for any subnet of the major network can be sent through router R1. The same principle works for other major networks—in this case, network 30.0.0.0. Routers can have only summary routes for them.

Let's take a more careful look at the routes to network 30.0.0.0 in R2 and R1. Both routers have routes to it, illustrating the first and second routing rules given in Section 3.2. Every router along the path to a destination must have adequate routing information. Router R0 in network 30.0.0.0 demonstrates the third rule: Every router must have infor-

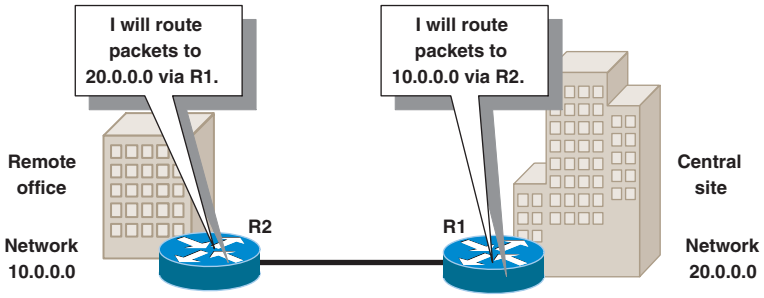

Figure 3-4. *Use of static routes—simple example*

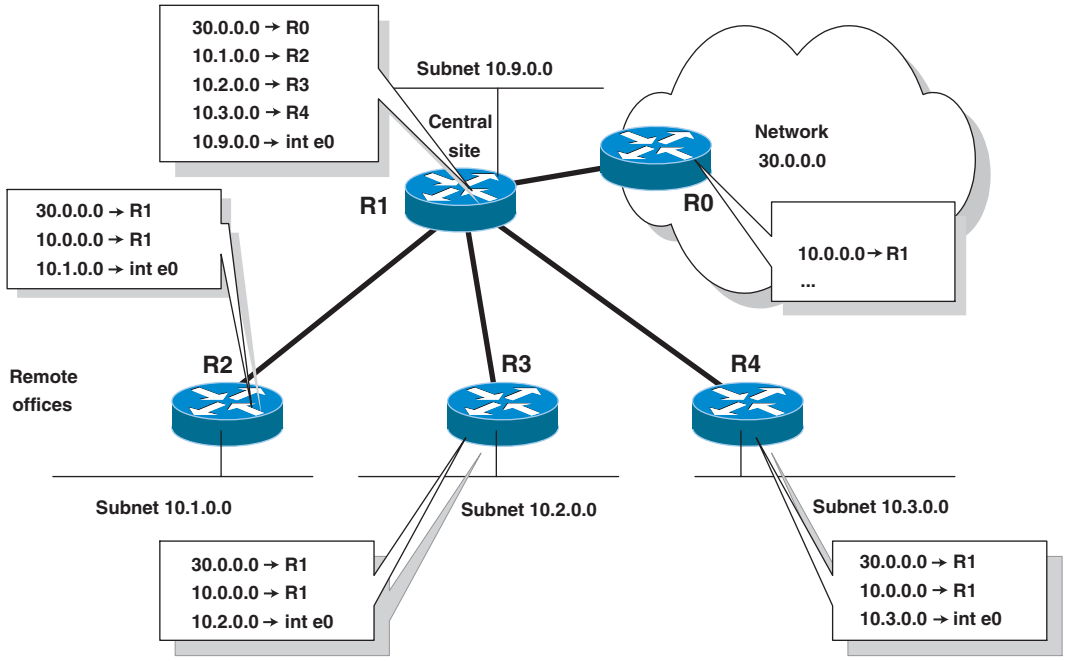

Figure 3-5. *Static routes—hub-and-spoke topology*

mation about the return path. For this reason, a static route to major network 10.0.0.0 was added to R0's routing table. Without it, R0 would drop the packets going back from 30.0.0.0.

In general, when configuring a static route, the administrator explicitly or implicitly specifies the next-hop address and the outgoing interface to be used to reach a certain network. If a static route is configured specifying only the address of the next router, this

is the next hop, configured explicitly, and the outgoing interface is determined by the recursive routing table lookup operation. If a static route points to a certain router's interface, the interface is explicitly specified, and another step is taken to find the next-hop address (see Chapter 4).

As you can see, the larger the network, the more difficult it is to control. The administrator must be very careful while configuring static routes, as it is easy to create a routing loop. This can happen because when a static route is configured, there is no way for the router to check and see whether the destination network is really behind the specified next hop. Dynamic routing is not a panacea for this problem—it can also experience temporary routing loops in transition periods—but dynamic routing protocols are based on the algorithms that guarantee convergence of the network within a finite period of time.

3.5 Dynamic Routing

You already know that dynamic routing protocols are a means of exchanging routing information between routers. Thousands of complex networks run different types of routing protocols. The best-known example of such a network is the Internet, a community of independent providers, customer networks, and *Internet exchanges* (IXs), each having its own set of routers, which share common traffic policies and administration (Figure 3-6). Such a set of routers, controlled by a single organization or provider, is usually called an *autonomous system* (AS). This term is widely used in the Internet community.

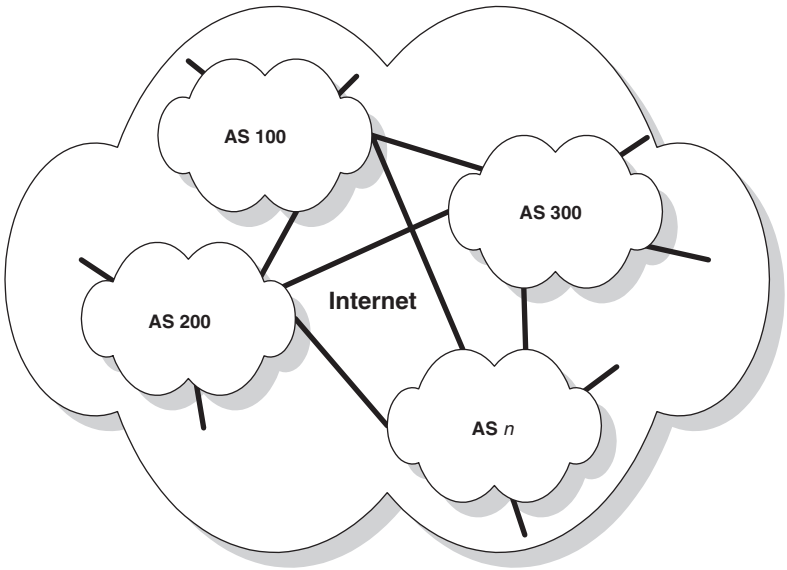

Figure 3-6. *The Internet, a group of autonomous systems*

The term *routing domain* is frequently confused with the term autonomous system. A routing domain is a set of routers running a single dynamic routing protocol, such as RIP, OSPF, or IGRP. An autonomous system may consist of many routing domains, each running its own protocol, and still be a single entity to the outside world, as shown in Figure 3-7.

All routing protocols can be classified as *interior gateway protocols* (IGPs) or *exterior gateway protocols* (EGPs). IGPs run inside an autonomous system and perform so-called *intra-domain routing* functions. This set of protocols consists of RIP v1/v2, IGRP, OSPF, EIGRP, integrated IS-IS, and some other, rarely used ones. EGPs run between autonomous systems. The set of EGPs includes two protocols—EGP and BGP. BGP version 4 is now the de facto standard for inter-AS routing, also called *inter-domain routing*. The main difference between the IGPs and EGPs is in the goals the two types are designed to achieve. IGPs are implemented to provide fast convergence within ASs, whereas EGPs are designed to share network reachability information—which networks are in which ASs—and to permit application of routing policies, influencing the results of the local and remote best-path selection algorithms. This book describes intra-domain routing via the most widely used IGPs—RIP, IGRP, OSPF, and EIGRP. (Inter-domain routing is not considered here, as this topic deserves a separate book.)

All routing protocols share the same basic concept: They exchange messages containing information about the networks routers know about. These messages are called *routing updates*. Every routing protocol has its own format for routing updates and its own

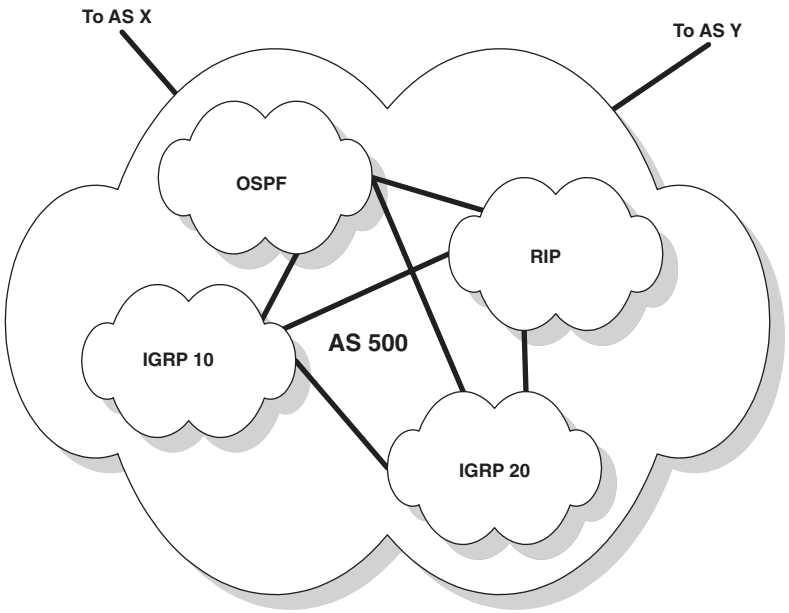

Figure 3-7. *Autonomous systems—a set of routing domains*

algorithm for exchanging and analyzing them. Routing updates contain information about one or many remote networks that the sending router has information about. While sending information about a network in an update, routing protocols supply additional information that can be used to understand how far the network is from the advertising router and how optimal the route is. Based on this information, routers calculate *metrics* for the routes and use those metrics to characterize the quality of routes and to perform route comparison. Different routing protocols use different information for metric calculation. The simplest example is RIP, which uses the number of hops. More sophisticated protocols, such as IGRP and EIGRP, use a set of parameters, such as minimum path bandwidth, maximum path delay, and so on.

When it receives a dynamic routing update about a remote network, a router selects the best routes—one or many—to the destination, based on the metrics associated with the routes. Having chosen the best route, the router can determine the two parameters for the routing table entries: the outbound interface and the next-hop address. In the simplest case, the outgoing interface is the one the routing update was received on, and the next-hop address is the address of the router that sent the update. Some protocols include the next-hop address in the routing update, and this address can be different from that of the advertising router. Also, link state protocols such as OSPF, do not care about the interface the update is received on. They calculate the routes on the basis of the network topology information.

3.6 Default Routing

In some situations, it is not desirable for all routers to have complete routing information. For example, in Figure 3-8, router R1 should know only about subnets 10.1.0.0–10.3.0.0; if it is sending a packet somewhere else—the Internet or another company's network—it sends it via router Rc, which has a full routing table. The routing decision for router R1 would then be, "If I don't have information about a destination network in my routing table, I use the route to Rc." This type of routing information is called a *default route*. "Default" means that the route is used only if there is no more specific information about the destination network a packet is going to.

Use of default routes simplifies network management, as it saves a router from having to know all networks and makes it possible for routers to reference a "smarter guy" that can do the work. Default routes are widely used in real networks.

Like any other route, default routes can be either static or dynamic. The administrator can configure a static default route manually, saying, "For all other destinations, use router Rx." In case of a dynamically propagated default route, the administrator configures a router to originate it and say, "Everyone around, use me to reach any network you don't know about." After receiving such a routing update, other routers install the default route in their routing tables; the administrator doesn't need to configure it manually (Figure 3-9).

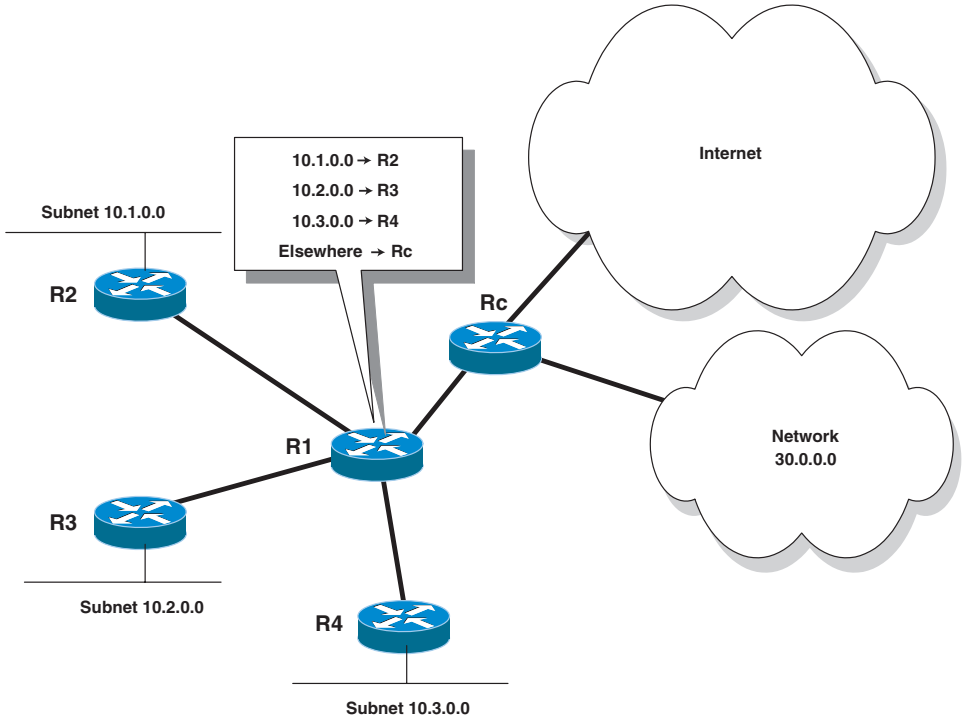

Figure 3-8. *Principle of default routing*

A default route is usually represented as a pseudonetwork with all 0s in the address and mask parts, that is, 0.0.0.0 0.0.0.0 or 0.0.0.0/0. Most dynamic protocols use this convention to provide default routing information. IGRP, however, has its own method, considered in Chapter 8.

Default routes are similar to summary routes. A summary route to a major network hides all details of that network, saying, "Want to reach someone in this network? Go this way!" A default route hides all details about all networks, saying, "Want to reach someone somewhere? Come on over here!" The way routers treat summary and default routes, as well as the algorithm of router operations, depends on the kind of environment—classful or classless—the routers work in. The following sections cover the basic forwarding algorithm and the details of the routing table lookup performed in classful and classless modes.

3.7 Basic Forwarding Algorithm

This section provides an overview of the forwarding algorithm performed by the router when it is clear that the packet is not destined for the router itself and should be delivered to a remote network. The algorithm uses the following data structures:

- *Packet:* the IP packet being forwarded. Each packet has fields described in Chapter 2. In particular, the forwarding algorithm uses such fields as Destination Address and Time-To-Live (TTL).
- *Interface:* the network attachment description. Various characteristics are associated with each interface, including the following, which are considered interesting from the forwarding perspective:
  - *Type*—Can be point-to-point, point-to-multipoint, or broadcast, depending on the type of encapsulation. For example, PPP and High-Level Data Link Control (HDLC) interfaces are point-to-point; Frame Relay and X.25 interfaces are point-to-multipoint; Ethernet and Token Ring interfaces are broadcast.
  - *State*—Operational status—up or down—of the interface. The state of the interface is determined by the status of the physical and the data link layer protocols.

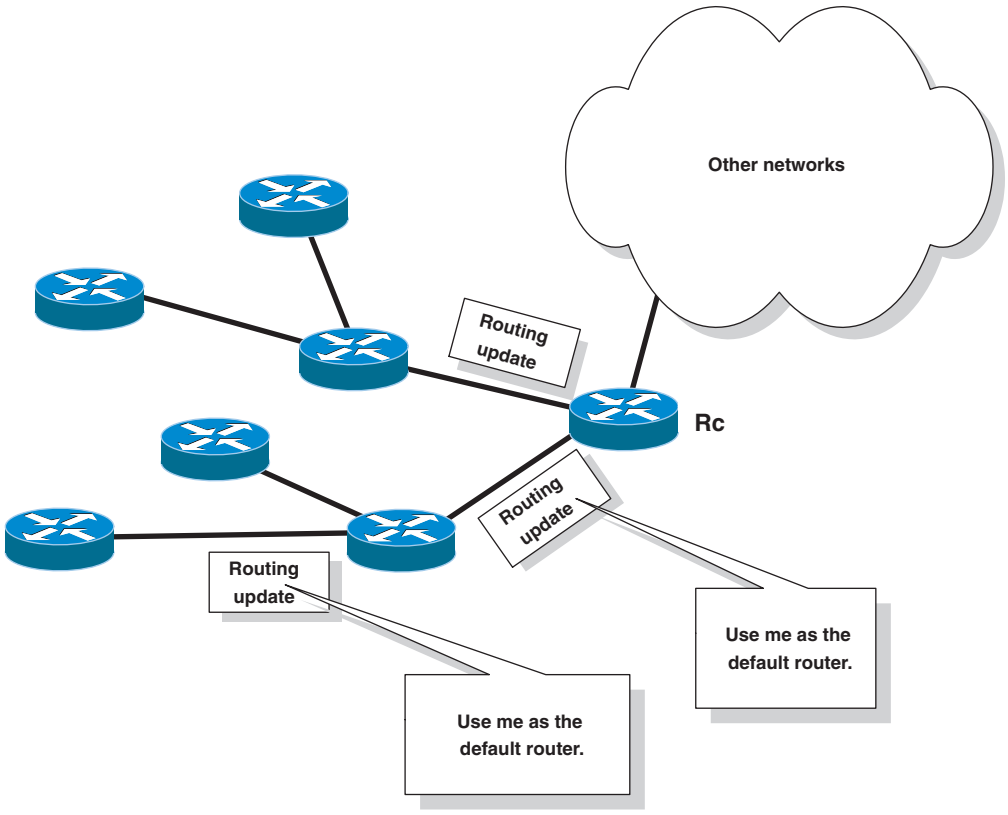

Figure 3-9. *Dynamic propagation of a default route*

- *IP status*—Flag specifying whether IP processing is enabled on the interface.
- *IP unnumbered*—Flag indicating that the interface—point-to-point—is configured as unnumbered.
- *Reference interface*—Interface whose IP address should be used when the packets are generated for the unnumbered interface.
- *IP address*—Address assigned to the interface.
- *Address mask*—Mask configured together with the IP address to specify the border between the network and hosts parts of the address.

- *Routing table*—Collection of *routing entries* (routes). The following parameters are associated with each entry:
  - *Network prefix*—IP prefix—in the form of the prefix value and its length, or a network address and a route mask—that describes a collection of destinations. For example, 192.0.0.0/8, or 192.0.0.0 255.0.0.0, describes all IP hosts that are assigned IP addresses starting with 192, such as 192.1.1.1 or 192.200.150.129.
  - *Default candidate*—Flag indicating that the route should be considered a candidate for becoming the default route
  - *Paths*—Collection of next-hop structures, each corresponding to a distinct path to the destination through the network. The following parameters are associated with each path, and at least one of the two must be present.
    - *Outbound interface*—The interface that should be used to forward packets to the collection of destinations described by the route. If the path does not specify the interface, the route is considered *recursive*.
    - *Intermediate address*—If the path specifies the interface, this is the next-hop address that should be used to find out the data link layer details. If the path does not specify the interface, this is the address that should be used for the next iteration of the recursive routing table lookup operation.

The following algorithm is an outline of functionality performed by the routers. The packet is assumed to have passed initial checks: the sanity check (basic IP header validity verification), the inbound packet filtering policy, the TTL field check, and so on. These checks and the forwarding algorithm are discussed in more detail in Chapter 5.

1. Set the next-hop address to the destination address in the packet.
2. Perform recursive routing table lookup operation as follows.
   a. Find the route for the current next-hop address in the routing table.

   b. If a route is found and it specifies the intermediate address, set the next-hop address to the address in the route.
   c. If the route is found and it does not specify the interface, loop back to step 2.a.
3. If the recursive route lookup did not succeed—no matching route was found or a route could not be resolved—send an ICMP "Destination Unreachable, Host Unreachable" message to the packet originator, using the source IP address in the packet as the destination IP address in the ICMP message, and drop the packet.
4. Otherwise, if the current value of the next-hop address equals the prefix value of the found route, set the next-hop address back to the destination IP address in the packet.
5. Pass the packet to the packet-delivery function. Provide the interface in the route and the current next-hop address as the arguments.

The algorithm is pretty simple. First, the routing table is searched for a route that can be used to route to the destination IP address in the packet. (The routing table lookup algorithm is discussed later.) If a route is found and it specifies only an interface—describes a directly connected network—the packet is sent out of the specified interface, using the destination IP address in the packet as the next-hop address. If the route specifies both the address and the interface—this is how IGP routes are installed—the packet is sent out of the interface to the next-hop router corresponding to the address in the route. If the route is recursive—only the intermediate address is specified—the intermediate address becomes the current next-hop route, and a routing table lookup operation is performed again.

The check in step 4 needs more explanation. That check is required when the routing table contains information similar to that shown in the following example:

```
10.0.0.0 is accessible via 20.1.1.0
20.1.1.0 is directly connected to the interface Ethernet 1
```

The recursive route to network 10.0.0.0 specifies a subnet address (20.1.1.0) as the intermediate address. Without the check, the subnet address would be used as the next-hop address. The check makes sure that the destination address in the packet is used as the next-hop address in this situation.

The packet-delivery procedure is initiated by the forwarding algorithm and receives the packet, the outbound interface, and the next-hop IP address as the arguments from it. Following is the outline of the steps taken by the packet-delivery process.

1. If the interface state is down or IP processing is not enabled on the interface, send an ICMP "Destination Unreachable, Host Unreachable" message to the source host, and stop processing the packet.

A route in the routing table can reference an interface in down state while the routing table is converging—it takes time to remove invalid routes—or because a static route through an interface was configured to be never removed from the routing table (see Chapter 6).

2. If the interface type is point-to-point, pass the packet directly to the packet encapsulation procedure specific to the interface. There is no need to look up data link layer details for point-to-point interfaces. They are either not necessary (such as HDLC or PPP encapsulation) or statically configured for the interface, such as, a point-to-point Frame Relay interface.
3. Otherwise, if the interface type is point-to-multipoint, perform the following steps.
   a. Search the map table associated with the interface, using the next-hop address as the search parameter.
   b. If no map for the next-hop address is found, log an encapsulation failure, send an ICMP "Destination Unreachable, Host Unreachable" message to the source host, and stop processing the packet,
   c. Otherwise, pass the packet to the packet encapsulation procedure specific to the interface, and pass the located map table entry as a parameter; it will be used to construct the data link layer frame for the packet.
4. Otherwise, if the interface type is broadcast, perform the following steps.
   a. Search the ARP cache for the MAC address corresponding to the next-hop address and outbound interface.
   b. If no ARP entry is found, log an encapsulation failure, send an ARP request message for the next-hop address, send an ICMP "Destination Unreachable, Host Unreachable" message to the source host, and drop the packet.

Note that the router does not wait for the ARP reply message to come in and does not queue the packet.

   c. Otherwise, pass the packet to the packet encapsulation procedure specific to the interface, providing the found ARP entry as a parameter; it will be used to construct the data link layer frame for the packet.

As you can see, the data link parameters vary by type of interface. Point-to-point interfaces require very little additional work. Point-to-multipoint links, such as Frame Relay or X.25, require the DLCI or X.121, which should be used to reach a specific next-hop router. The mapping between the next-hop addresses and the data link layer details is usually configured manually by the administrator (see Chapter 5 for details). Broadcast interfaces require knowledge of the next-hop router's MAC address that is discovered using ARP. Also note that the interface MTU check and the IP packet fragmentation functionality are performed by the packet encapsulation function.

3.8 Classful Routing Operations

Before we proceed to the principles of classful IP routing, an important detail about routing table entries needs to be discussed. The routing table structure defined in the previous section specified that each route describes a set of destinations in the form of a network prefix with its value and length. However, all examples given before showed routes in the routing tables just as network addresses (10.0.0.0) instead of as network prefixes (10.0.0.0/8). Indeed, when they install routes in the routing table, Cisco routers provide the corresponding route mask as well. This discussion of the idea behind the route mask was intentionally delayed until the topic of routing table lookup functionality. For better understanding of why routing table entries need masks, let's consider several examples.

Suppose that a summary route to major network 10.0.0.0 with mask 255.0.0.0 and a route to subnet 10.0.0.0 255.255.0.0 (zero subnet) are installed in the routing table as shown in the following example. Note that if routes did not have associated masks, these two routes would be indistinguishable.

```
10.0.0.0/8      – via 10.2.0.1
10.0.0.0/16     – via 10.2.0.3
```

The number after / in the routes represents the number of significant bits in the prefix value; the number is equal to the number of bits in the route mask. So, /8 implies a mask of 255.0.0.0, and /16 means a mask of 255.255.0.0. The route mask specifies the portion of the route's network address that must be compared with the destination address in the packet. A route is declared matching a destination address if the bits in the address corresponding to the bits set to 1 in the route mask are equal to the bits in the same positions in the route's prefix value. In our example, both routes would match address 10.0.0.15 because the 10.0.0.0/8 route matches the first octet of the address, and the 10.0.0.0/16 route matches the first two octets of it. In this situation, 10.0.0.0/8 is said to be a less specific route, and 10.0.0.0/16 is said to be a more specific one. When multiple matching routes are available to the same destination, routers choose the longest matching route to forward the packets. So, for a packet to IP address 10.0.0.15,

the 10.0.0.0/16 route would be used. The other route, 10.0.0.0/8, would be used to forward packets to all other, unknown subnets of major network 10.0.0.0. This is an example of a so-called *network default route*.

Now let's move on to the classful routing topic. The principles of classful addressing discussed in Chapter 2 state that the whole major network must use a single subnet mask. Moreover, the routing protocols designed for classful environments do not send the route masks in their updates. The route mask is determined on the basis of the address masks configured on the interfaces on which the updates are received. Indeed, routers do not have any other source of information about the subnet mask used on remote subnets. Let's see how dynamic routing protocols work in a classful environment.

An update message of a classful protocol can carry routes of the following types.

- Host routes
- Subnet routes
- Network routes
- Default routes

Consider two routers connected to subnets of the same major network—R1 to 10.1.0.0, 10.2.0.0 and 10.3.0.0 and R2 to 10.3.0.0, 10.4.0.0, and 10.5.0.0 (see Figure 3-10). The subnet mask used on the interfaces of the routers is 255.255.0.0. When it receives the update from router R2 about subnets 10.4.0.0 and 10.5.0.0, router R1 installs the routes to these subnets in the routing table with the subnet mask taken from its own interface connected to subnet 10.3.0.0, that is, mask 255.255.0.0. Provided that all hosts and routers in major network 10.0.0.0 use this subnet mask—a basic rule of classful addressing—this approach works.

Now imagine that R2 is connected to another major network, say, 20.0.0.0, which uses a different subnet mask—255.255.255.0. One of R2's interfaces is attached to subnet 20.1.1.0. If R2 sent information about this subnet to R1, R1 would make a wrong decision about the route mask. It would use its own, 255.255.0.0, and it would see that the network address is 20.1.0.0 and that the host part of the route is 0.0.1.0. If it saw that the host part was not all zeros, R1 would assume that the route was a host route and would insert it with the route mask of 255.255.255.255, which is wrong.

To prevent this misunderstanding, routers connected to multiple major networks do not send subnet information about one major network into another. Instead, they send summary network routes. The receiving routers are supposed to install these routes in the routing tables with the default class address masks. A route from any major network with a nonzero host part of the address is considered a host route. (Some implementations ignore host routes coming from remote major networks.)

The behavior when subnet routing information is not propagated from one major network to another is called *automatic route summarization*. It has its own pros and cons.

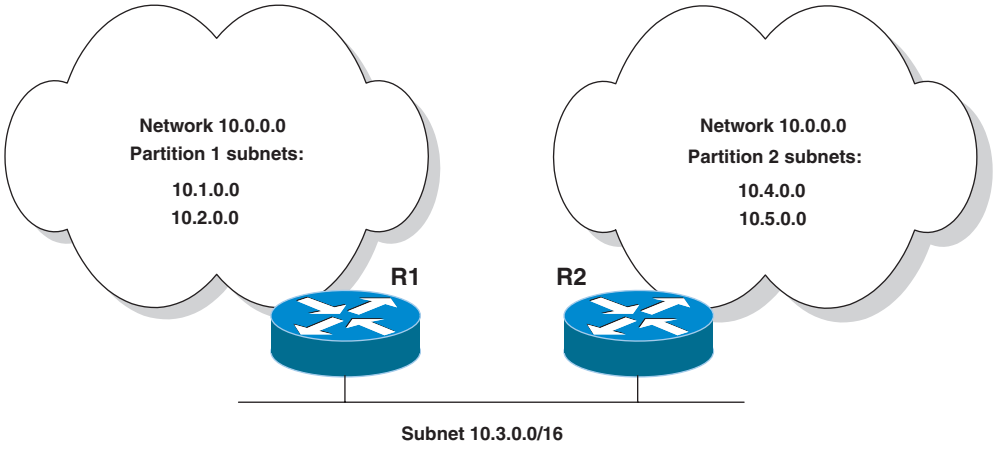

Figure 3-10. *Example of classful routing*

On the one hand, autosummarization decreases the size of the routing tables. On the other hand, it can cause routing problems when *discontiguous networks* are used. For example, consider a situation in which major network 10.0.0.0 is geographically divided by major network 20.0.0.0 (Figure 3-11). The border router R1 sends only a summary route, 10.0.0.0, to routers in major network 20.0.0.0. Border router R2 does the same. So, the router in major network 20.0.0.0 has no information about subnets of network 10.0.0.0 and hence cannot properly route packets going there. In the worst case, routers inside major network 20.0.0.0 choose only the best route to network 10.0.0.0 and install it. This leads to a situation in which a router can see only one partition of the network and cannot send packets to the other. In the best case, when the routers have routes with equal metrics, the routers use both and load-balance the traffic between them. This is still not good, as packets can be sent in a wrong direction. In both cases, the routing is not functional.

Another interesting subject is how the routing table lookup operation is performed when routers are working in a classful environment. The algorithm follows.

1. If the routing table contains a route to a destination in the major network that the destination address belongs to, including a route to the major network itself, perform the following steps.
   a. Look up the longest matching route limiting the set of routes to those describing the destinations in that major network.
   b. If no route is found, do not consider the supernet routes or the default route; indicate a route lookup failure, and stop the algorithm.

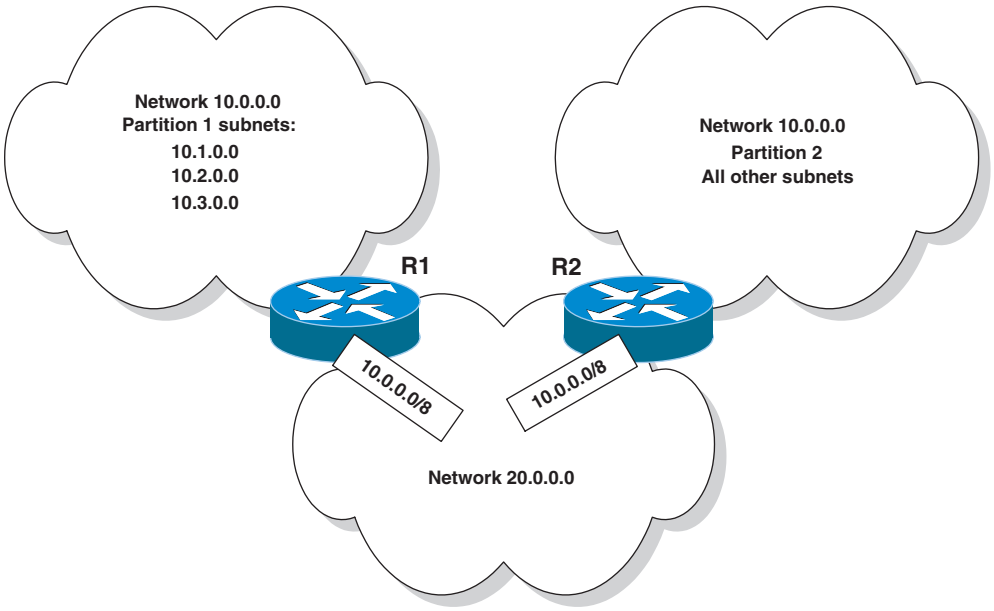

Figure 3-11. *Classful routing when one major network is split by another*

2. Otherwise, look up the best matching route among the supernet routes. If a route is found and it is not a route to pseudonetwork 0.0.0.0, use the route for packet forwarding.
3. Otherwise, if the default route is available, use it to forward the packet.
4. Otherwise (no matching route and no default route is available), indicate a route lookup failure.

The behavior of the lookup algorithm depends on the result of the first step. If the router has any route for a network in the destination major network—that is, the router is assumed to have attachments to the major network, the only routes considered are those describing address ranges in the same major network: subnet and network default routes. If no route belongs to the destination major net, a simple best-matching route is chosen. Note that the default route is used only if the router is not attached to the destination major network. (We discuss the reason for this later in this section.)

The following example illustrates how the routing table lookup algorithm works. Suppose that a router needs to forward a packet with the destination address 20.1.2.3. The routing table contains the following routes:

```
1: Network 10.0.0.0/8 is directly connected to the interface Ethernet 0
2: Network 20.0.0.0/8 is accessible via 10.1.1.1
```

```
3: Subnet  20.1.1.0/24 is directly connected to the interface Ethernet 1
4: Subnet  20.1.2.0/24 is accessible via 10.1.1.2
```

Following is the log of the steps taken by the algorithm.

1. The destination address is 20.1.2.3, and the destination major network is 20.0.0.0.
2. The router has routes in the destination major network—routes 2, 3, and 4—so the algorithm branches to step 1.a.
3. All routes in major network 20.0.0.0 are processed as follows.
   a. Route 2—20.0.0.0/8. The route's mask is applied to the destination address, which gives 20.0.0.0. This value is the same as the network address part of the route, so the route matches. So far, this route is the best, but the router proceeds to the next route, as there may be better ones.
   b. Route 3—20.1.1.0/24. The binary AND between the route mask and the destination address results in 20.1.2.0, which is different from this route's network address (20.1.1.0). This route does not match, so it is skipped.
   c. Route 4—20.1.2.0/24. The result of 20.1.2.3 AND 255.255.255.0 is 20.1.2.0, which is equal to the network address portion of the route. The current best route is route 2 (20.0.0.0/8). The route mask of route 4 (255.255.255.0) is longer than that of route 2 (255.0.0.0), so 20.1.2.0/24 is better and is selected as the current best route.
   d. Because no other routes are in the routing table belonging to the destination major network, the algorithm stops, and the 20.1.2.0/24 route is considered the best.

Consider a situation in which a router has somewhat different routing information, the presence of the default route:

```
1: Network 10.0.0.0/8 is directly connected to the interface Ethernet 0
2: Subnet  20.1.1.0/24 is directly connected to the interface Ethernet 1
3: Subnet  20.1.3.0/24 is accessible via 10.1.1.2
4: Network 0.0.0.0/0 is accessible via 10.1.1.3 (Default route)
```

The router finds out that the destination (20.1.2.3) belongs to the same major network but cannot find a matching route. Now, because the destination address is in the same major network as the router, the algorithm does not try to find a default route and the route lookup fails, causing the packet to be dropped.

The idea behind treating the default route this way in classful routing is that if a router does not know about some subnet of a "known" major network—the administrator didn't configure a static route or a routing protocol didn't send any information about it—it considers this subnet either nonexistent or down. If the packet were destined for another major network that the router is not connected to, it would take the default route and send the packet to the corresponding router (10.1.1.3).

For better understanding, let's consider another example (Figure 3-12). Routers R1 and R2 are boundary routers. Router R1 has announced the summary route to its major network (10.0.0.0) to R2, as has R2 (route 20.0.0.0 in R1's routing table). Now assume that the administrator of network 10.0.0.0 configured a default route pointing to R2. Consider a situation in which a station from network 20.0.0.0 sends an IP packet with the destination address 10.4.1.1. R2 routes the packet to R1. Looking through the routing table, R1 does not find a route to the destination subnet. Assume that R1 takes the default route, sending the packet back to R2, which in turn sends it to R1 again. The packet is looped until the TTL field in the header reaches 0 and the packet is dropped by one of the routers.

In other words, because a default route is used to describe destinations in other networks, sending a packet along a default route is equivalent to sending a packet in the direction of the exit from the local major network and, finally, out of it. Because major

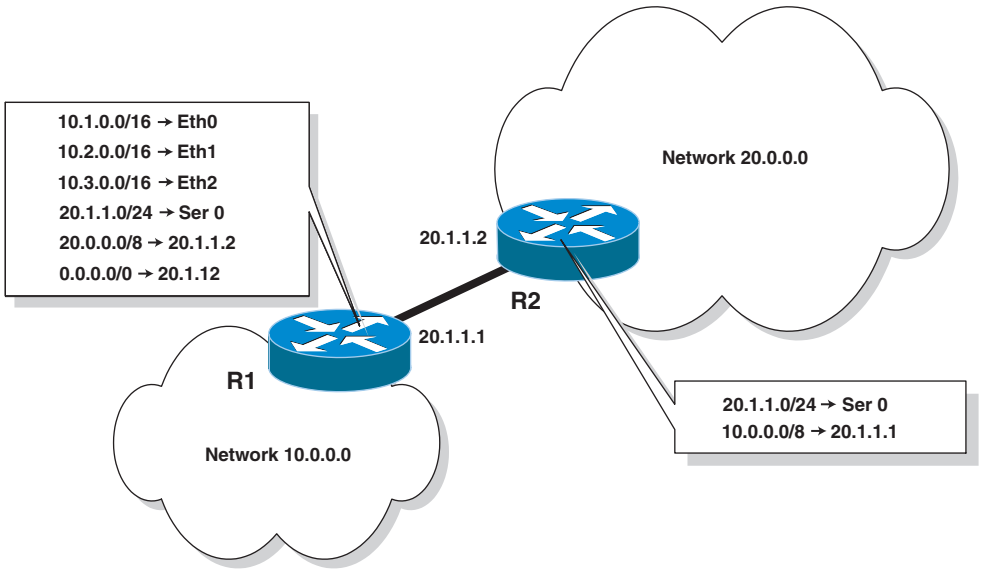

Figure 3-12. *Use of default in classful routing*

networks must be contiguous in classful routing, forwarding a packet out of its destination major network doesn't make much sense and leads to routing loops because routers in other major networks believe that a given subnet can be found in the major network the subnet belongs to.

This concept also applies to the *supernet routes:* routes installed in the routing table with route masks shorter than the default class masks. The only difference is that a supernet route aggregates information about several major networks, whereas the default route aggregates information about the rest of the world.

A router's behavior in a classful environment can be summarized as follows:

- Classful routing protocols do not include subnet masks in routing updates.
- Classful routing protocols hide subnet information from other major networks by announcing only summary network routes into them.
- Classful routing protocols can announce host routes, which are used when hosts do not reside on the same segment as the rest of the subnet.
- Each classful router can have the following types of routing information in its table:
  - *Host routes*—Routes received with nonzero host address parts and implicitly assigned the network mask of 255.255.255.255 or static routes with explicitly configured masks.
  - *Subnet routes*—Routes to subnets within the major network to which a router has an attachment. These routes are inserted into the routing table with subnet masks on the interface the update is received from used as route masks, unless it is a static route and was configured with different subnet mask.
  - *Network summary routes*—Routes to other major networks. These routes are inserted into the routing table with the default classful address mask (without subnets) and represent other major networks if they are provided by routing protocols. This type of route can also be used to represent the rest of the local major network (the network default route). Such a route must be statically configured by the administrator.
  - *Default routes*—Either a 0.0.0.0/0 route, which is marked as default by the router automatically, or other routes to any networks explicitly marked as default by the administrator or a routing protocol (see Chapter 4 for a detailed explanation).
- While routing packets, routers pay attention to whether the destination major network is local (some of its subnets are directly connected).

- If the destination major network is local, the router needs to have either a host route with a /32 mask or an explicit route to the subnet or a summary network route describing the rest of the local major network. If this condition is not met, the packet is dropped, and the default and supernet routes are not considered.
- If the destination major network is not local, the routing table lookup algorithm is changed. The router looks for the best-matching route, paying attention to the length of the route masks; supernet routes may be taken. If no match was found, the router checks the default route. If there's no default route, the packet is dropped.

Following are some problems that can be seen in classful environments.

- Variable-length subnet masks cannot be used, as routing updates do not contain route masks.
- Automatic summarization to classful networks prevents use of noncontiguous addressing plans, such as private IP addresses for WAN links.
- Use of default routes is limited, which can be a problem in very large networks because the only type of summarized route that can be distributed within one major network by routing protocols is the default. This occurs because if the routing protocol sends a network summary route for a local major network, this route is considered an update for zero subnet, not a network summary. So, every router must have either a full routing table or a network summary route representing the rest of the local major network configured manually.

These problems are addressed by the classless routing approach, discussed in the next section.

3.9 Classless Routing Operations

Let's try to understand what enhancements should be made to the routing process and routing protocols to guarantee normal routing of classless addressing schemes. Because classless routing assumes use of VLSMs, routes in the routing updates should be augmented with route masks. Once this condition has been met, the problem with mandatory summarization is automatically canceled; even if information about subnets is sent into another major network, the routers have explicit information about the corresponding route mask in the update. This is not enough, however.

Even though automatic route summarization can cause problems, the technique in general is necessary to scale the networks. Without route summarization, every router would have to know about every prefix in the network. One of the problems that stimulated

deployment of CIDR in the Internet was the growing size of the routing tables in the backbone routers. At that time, every backbone router had to know every major network in the Internet, and because the number of networks connected to the Internet grew at an exponential rate, the growth of the routing table size followed the same trend. There had to be a method that would stop this. *Route aggregation*, a part of CIDR, is a technique similar to classful summarization and allows routes to be aggregated at an arbitrary boundary and hence permits aggregation of major networks. Arbitrary aggregation also helps reduce the size of routing tables within major networks.

The principle of route aggregation states that routes to specific subnets or networks can be represented by a smaller number of aggregate routes. The route masks of the aggregate routes do not need to equal the default classful mask or the subnet mask used in the major network. To illustrate this, consider an example in which a network is geographically split in three parts (Figure 3-13).

The first remote site is assigned subnets 10.1.1.0–10.1.14.0; the second one uses subnets 10.1.17.0–10.1.30.0. All other subnets are located in the central site. The classful routing approach requires either a full routing table or network default routes on each router in this network. With classless routing, another method can be used.

Look at the first three address bytes of the subnets in remote site 1; the third byte is represented in binary notation in Table 3-1. Only the four lowest-order bits change in the third byte; the highest-order four bits do not. This means that all subnets in the site do not have to be announced, provided that all outside routers know that subnets with the four highest-order bits in the third byte, set to 0, and the first two bytes, set to 10 and 1, can be reached via router R1. This can be achieved by propagating only an aggregate route, 10.1.0.0/20, to R2 and R3. In this case, the route represents not a real subnet but a group of subnets. The routing tables in the routers could be as shown in Figure 3-14.

Note that networks in remote site 2 are also represented by an aggregate route: 10.1.16.0/20. Another interesting detail is that all subnets in the central site are described in R1 and R3 via a network summary route. Also, pay attention to the fact that router R3 has two routes with the same address parts but different route masks. Because the routing table lookup algorithm selects the most specific—the longest matching—route for the destination address, routing is unambiguous. This rule can be demonstrated by the forwarding decision that router R3 makes.

Suppose that R3 receives a packet destined for address 10.1.2.10. The router starts looking through the routing table. In its first iteration, the best route it finds is 10.1.0.0/16, but when R3 looks at the second one—10.1.0.0/20—it finds that this route also matches and has a longer route mask. So, route 10.1.0.0/20 is selected as the best and the packet is forwarded to R1. If it had to route a packet destined for one of the subnets within the central site, R3 would see that the second route doesn't match, and the first one would remain the best.

The same principle works in aggregating routes describing classful networks. Consider an example of a small Internet provider that is given eight class C networks,

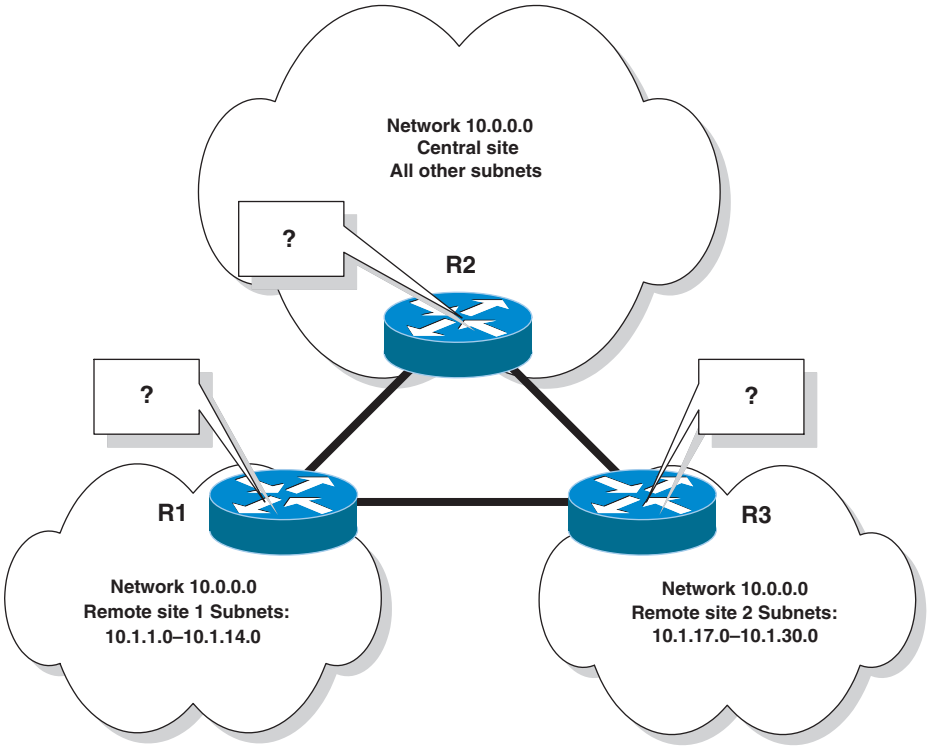

Figure 3-13. *Classless routing—a sample network*

Table 3-1. *Aggregation of Subnet Routes*

Subnet	First Byte	Second Byte	Third Byte	Fourth Byte
10.1.1.0	10	1	0000 0001	0
10.1.2.0	10	1	0000 0010	0
10.1.3.0	10	1	0000 0011	0
10.1.4.0	10	1	0000 0100	0
....				
10.1.14.0	10	1	0000 1110	0

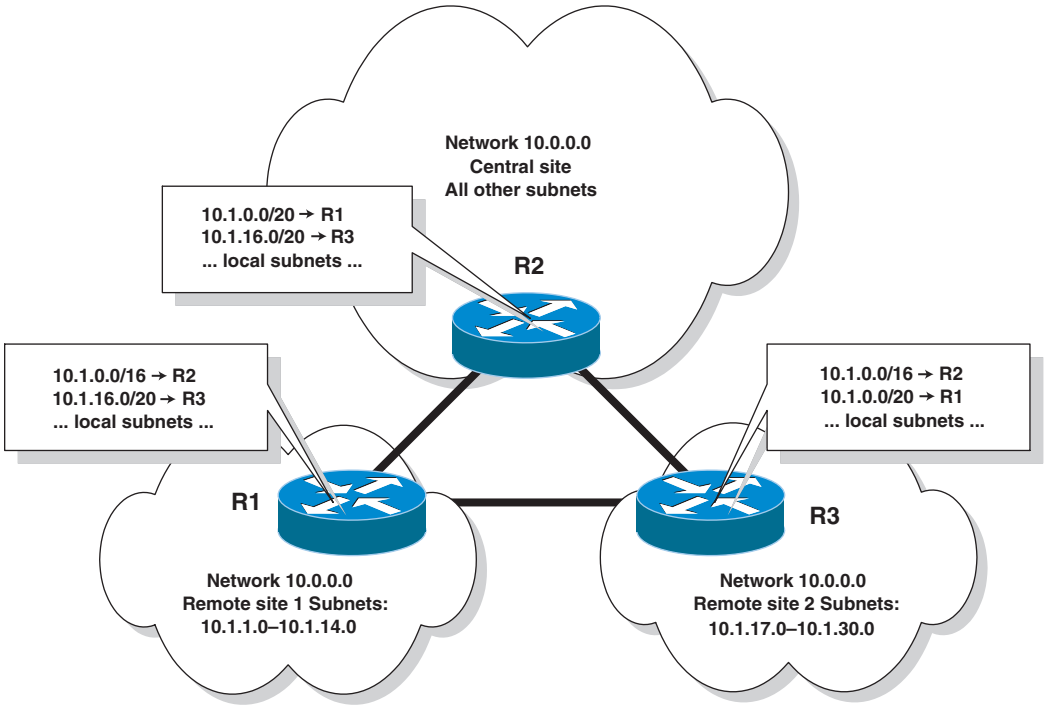

Figure 3-14. *Classless routing—route aggregation*

190.150.16.0–190.150.23.0. The method used to aggregate subnet routes is used to aggregate class C networks to a supernet (Table 3-2). Note that only the three rightmost bits change. This means that all these networks can be announced with one aggregate route, 190.150.16.0/21, so we have only one route instead of eight. Note that this aggregate route uses an arbitrary address mask, which is shorter than the default classful mask. This makes it a supernet route in contrast to a subnet route having the mask longer than the default. Note that supernet routes could not be used with classful routing protocols, because routes in the updates are not accompanied by the route masks. It is the route mask announced in the updates that allows classless routing protocols to announce supernet routes or arbitrary aggregate routes in general.

In classful routing, default routes are used very carefully. Classless routing principles state that default routes must be taken every time a router doesn't know where to route the packet. This is easily understood because in classless routing, there are no classes and therefore no major networks. In classless routing, the whole network address is not divided into network, subnet, and host portions but rather is considered as a combination of a network prefix and a host part, where the network prefix is derived by using the mask of a given route. A router looks through the routing table for a route whose net-

Table 3-2. *Building a Supernet Route*

Network	First Byte	Second Byte	Third Byte	Fourth Byte
190.150.16.0	190	150	00010 000	0
190.150.17.0	190	150	00010 001	0
190.150.18.0	190	150	00010 010	0
190.150.19.0	190	150	00010 011	0
....				
190.150.23.0	190	150	00010 111	0

work prefix matches the destination address best, that is, the longest matching route. If no matching route is available, the default route is used.

This treatment of the default route seems to be acceptable, unless we recall why classful routing restricted use of the default route. This restriction prevents routing loops caused by sending packets out of the local major network if they are destined for one of its unknown subnets. Consider the same network similar to the one in the previous example, but assume that it uses default routes instead of network summary routes (see Figure 3-15).

Suppose that router R2 receives a packet with destination address 10.1.2.15 and that subnet 10.1.2.0 is down. Router R2 follows the aggregate route 10.1.0.0/20 and forwards the packet to R1. R1 does not have information about subnet 10.1.2.0 (because it is down), takes the least specific route in the routing table—the default route (0.0.0.0/0)—and sends the packet back to R2. The routers keep ping-ponging until the packet's TTL field is decreased to zero, when the packet is dropped. This situation represents the simplest two-hop routing loop.

To remedy this problem, classless routing requires that every router that announces an aggregate route drops a received packet if it's destined for one of the aggregated networks and the router has no information about the destination. In our example, when router R1 receives the packet from R2 and sees that it has no information about the destination subnet in the routing table, it must drop it. Analyzing which routes have been announced could be time consuming, and in any case, the aggregate route in R2 could be configured statically, so R1 would never know about it. This is why so-called discard routes are installed in the routers, announcing aggregates (R2 in the example). While looking for a route in the table, routers should consider the discard route as a normal one, but packet delivery along such a route must result in a packet drop. It may seem that having a route like 10.1.0.0/20–>Discard on R1 would cause the router to drop every

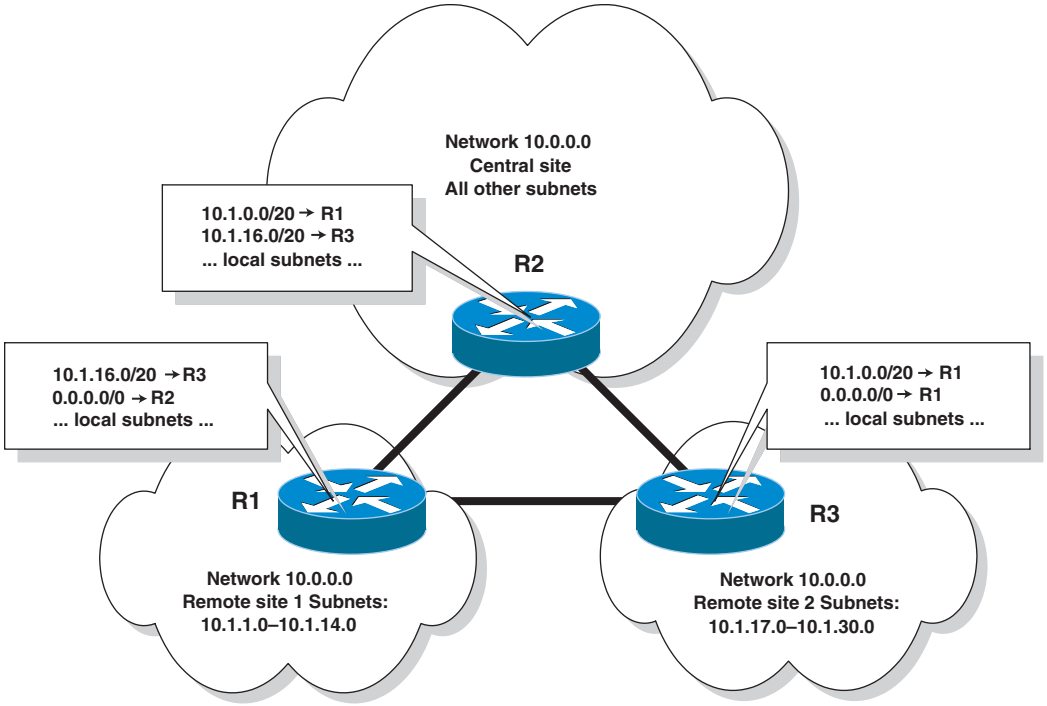

Figure 3-15. *Use of default route in classless routing*

packet that matches this route, but this is not what happens. Remember that the best-matching route is always picked up. So if the router has a more specific route to the destination, it will route the packet properly; only if the destination is not explicitly listed in the routing table will the router take the discard route and drop the packet.

The algorithm of the classless routing table lookup operation is really simple—look up the longest matching route in the routing table; if no matching route is available, use the default route. (If the default route is announced as 0.0.0.0/0, the last step is not needed, as this route matches all addresses.) As we will see in the following chapters, the exact details of the router operation are slightly different, but the behavior follows the principles described here.