Routing fundamentals

This chapter covers

- How end hosts send IP packets to local and remote destinations
- The routing process
- Reading and interpreting a router's routing table
- Configuring static routes on a router
- Using default routes to provide internet connectivity

In this chapter, we will cover routing—the process by which routers forward IP packets between networks. Specifically, we will cover elements of the following CCNA exam topics:

- 3.1 Interpret the components of a routing table
- 3.2 Determine how a router makes a forwarding decision by default
- 3.3 Configure and verify IPv4 and IPv6 static routing

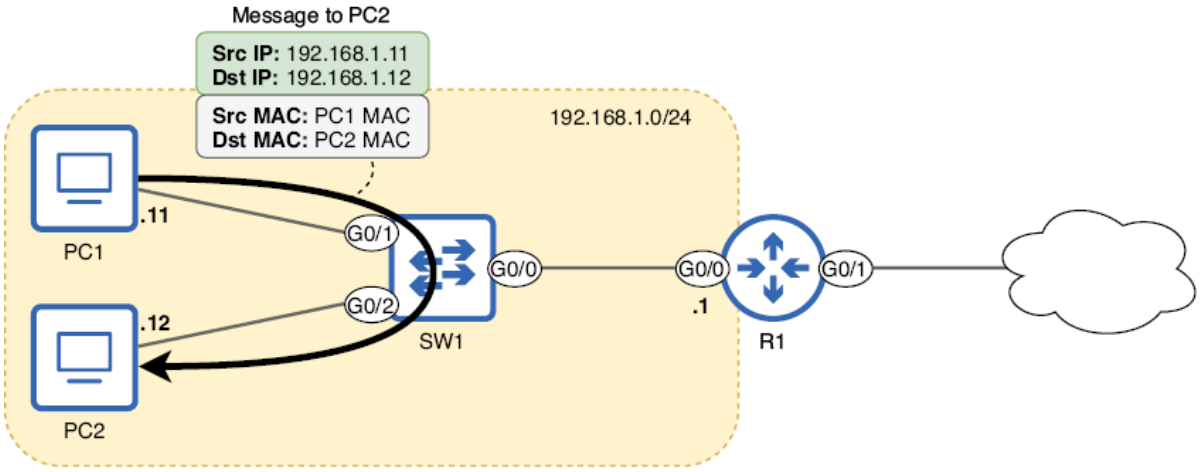
The term *routing* can actually refer to two different processes: the process by which routers build their *routing table* (a database of known destinations and how to forward packets toward them) and the process of actually forwarding packets. In this chapter, we will cover both aspects of routing, and we will build upon this foundation in future chapters of this volume and volume 2.

9.1 How end hosts send packets

Before we examine the details of how routers forward IP packets, let's take a look at the end hosts that send those packets to each other. After a host prepares a packet to send to another host, it must encapsulate that packet in a frame; even though we are focusing on routing, a Layer 3 process, do not forget about Layer 2! Packets are never sent over the cable (or radio waves) without being encapsulated in a frame.

The destination MAC address of the frame depends on the destination IP address of the packet. If the packet is destined for a host in the same network as the sender, the destination MAC address will be that of the destination host; in this case, there is no need for a router. Figure 9.1 demonstrates this process when PC1 sends a packet to PC2. The destination IP and MAC addresses are both PC2's; there is no need for R1 to route the packet, because the source and destination are in the same network (the 192.168.1.0/24 network).

Figure 9.1 PC1 (192.168.1.11) sends a packet to PC2 (192.168.1.12). Because PC1 and PC2 are both in the same network (192.168.1.0/24), PC1 encapsulates the packet in a frame addressed to PC2's MAC address. PC1 does not need to send the packet to R1 for routing. This diagram assumes PC1 already knows PC2's MAC address; if not, PC1 will first send an ARP request to learn PC2's MAC address.

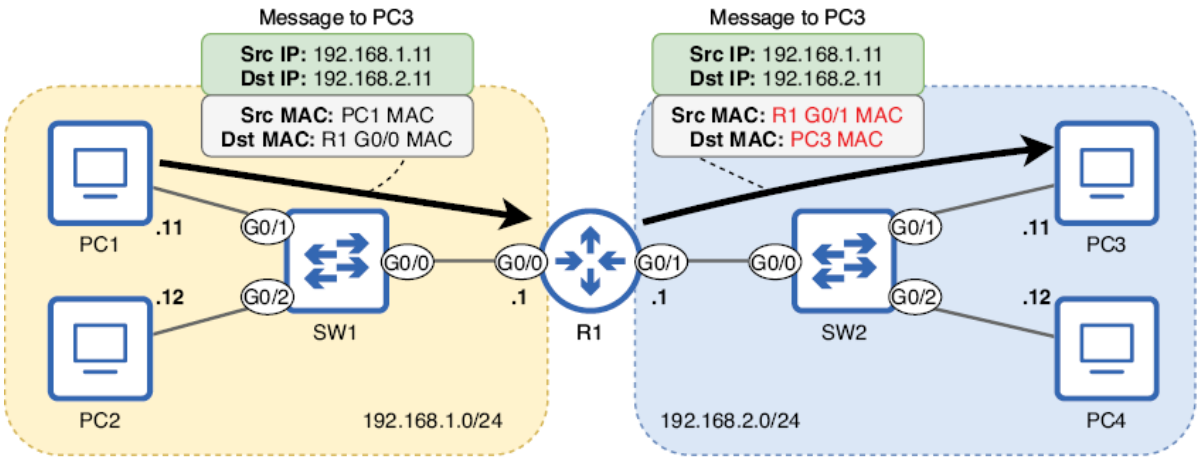


Note

A cloud icon, as shown in figure 9.1, is often used to represent the Internet. However, that is not always its purpose. A cloud icon can be used to summarize elements that are not relevant to the diagram. The cloud in figure 9.1 indicates that R1 connects to another network, the details of which aren't relevant to the diagram. That other network could be the Internet, or it could be another part of the same enterprise's network.

On the other hand, if an end host like a PC wants to send a packet to a destination outside of its local network, it must send the packet to its *default gateway* - the router that provides connectivity to other networks. In figure 9.2, R1 is the default gateway of the 192.168.1.0/24 network. For PC1 and PC2 to send packets to destinations outside of 192.168.1.0/24, they must encapsulate the packet in a frame addressed to the MAC address of R1's G0/0 interface. Figure 9.2 demonstrates how PC1 sends a packet to PC3: it encapsulates the packet in a frame, which is addressed to the MAC address of R1's G0/0 interface. R1 then forwards the packet out of its G0/1 interface, encapsulated in a new frame addressed to PC3's MAC address.

Figure 9.2 PC1 (192.168.1.11) sends a packet to PC3 (192.168.2.11). Because PC1 and PC3 are in separate networks, PC1 sends the packet in a frame addressed to its default gateway's MAC address—that of R1's G0/0 interface. R1 then forwards the packet out of its G0/1 interface, encapsulated in a new frame addressed to PC3's MAC address. This diagram assumes PC1 already knows R1 G0/0's MAC address; if not, PC1 will send an ARP request to learn it. Likewise, R1 must also learn PC3's MAC address.



Note

The default gateway's IP address is usually the first usable address of the network. For example, in the 192.168.1.0/24 network it's 192.168.1.1, and in the 192.168.2.0/24 network it's 192.168.2.1. That doesn't have to be the case, but it's common practice, and I will follow that practice in this book. The IP addresses of the PCs, on the other hand, are arbitrary. In this chapter's examples, the PCs' IP addresses end in ".11" and ".12", but there is no particular significance to those addresses.

How does PC1 know what its default gateway is? An end host can learn the IP address of its default gateway in a couple ways. One way is manual configuration, in which an admin manually specifies the default gateway on each device. However, this is very rare; end hosts usually use the second method - *Dynamic Host Configuration Protocol* (DHCP) - to automatically learn information like their default gateway's IP address, as well as their own IP address (DHCP is covered in chapter 29 of this book). On a Windows device, you can use the ipconfig command in the Command Prompt application to view information like the device's IP address, netmask (called the *subnet mask* in the command's output), and default gateway. The example below shows the output of this command on PC1.

```
C:\Users\jmcdo> ipconfig

. . .

Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :

   IPv4 Address. . . . . . . . . . . : 192.168.1.11

   Subnet Mask . . . . . . . . . . . : 255.255.255.0

   Default Gateway . . . . . . . . . : 192.168.1.1

. . .
```

Note

A host's default gateway is configured as an IP address, not a MAC address. To learn the default gateway's MAC address, the host must send an ARP request to the default gateway's IP address.

There are a couple of main points to take away from this section: first, to send a packet to a destination in the same network, a host will encapsulate the packet in a frame addressed to the destination host's MAC address. The second point is that, to send a packet to a destination in a different network, the sending host will encapsulate the packet in a frame addressed to the default gateway's MAC address. In either case, ARP must be used to learn the appropriate MAC address (that of the destination host or the default gateway).

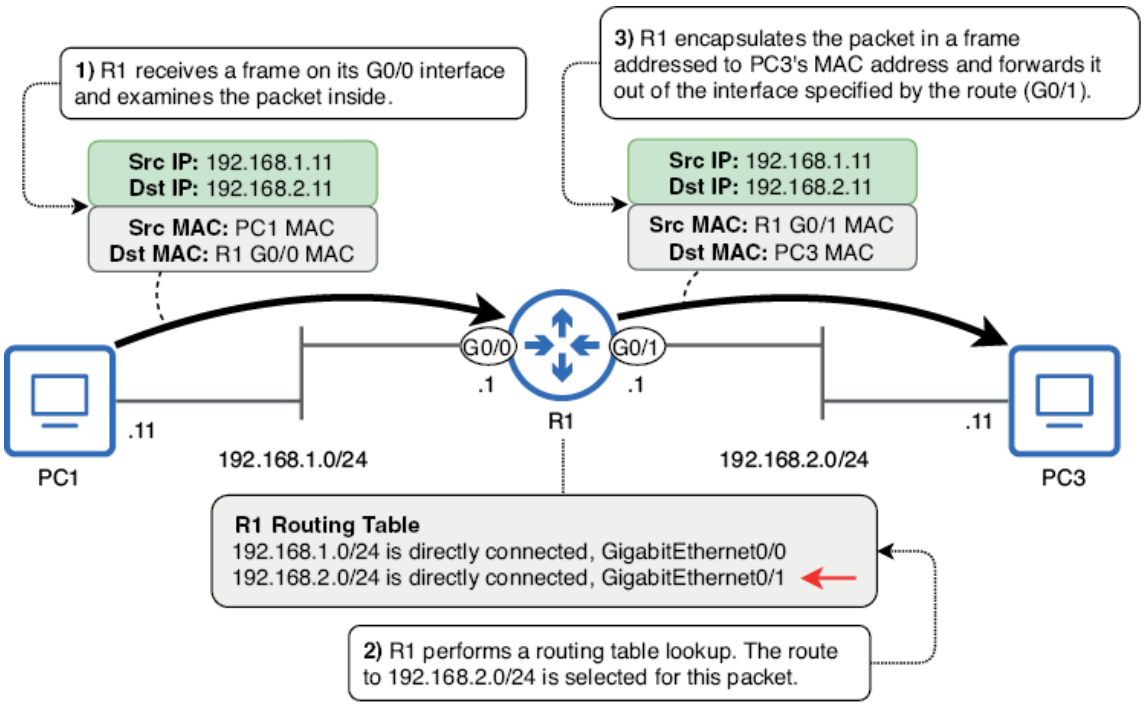
9.2 The basics of routing

In section 9.1, we saw how a host sends packets to destinations outside of its local network; it sends each packet in a frame addressed to the MAC address of the default gateway. Now we'll examine how the default gateway - which is a router - performs its role of forwarding packets between networks, which is called routing. Figure 9.3 gives a high-level overview of how R1 forwards a packet from PC1 to PC3.

R1's routing table in figure 9.3 is simplified - we will examine R1's complete routing table in section 9.2.1.

When a router receives a frame destined for its own MAC address, it will de-encapsulate the frame to examine the packet inside (if the destination is not its own MAC address, it will discard the frame). If the destination IP address of the packet is its own IP address, it will continue to de-encapsulate the message - it is a message for the router itself.

Figure 9.3 R1 receives a packet from PC1 and forwards it to PC3. (1) R1 receives a frame on its G0/0 interface. The frame is addressed to R1's own MAC address, so it examines the packet inside. (2) R1 looks up the packet's destination IP address in its routing table. 192.168.2.11 is in the 192.168.2.0/24 network, so it selects that route to forward the packet. (3) R1 encapsulates the packet in a new frame destined for PC3's MAC address and forwards it out of the interface specified by the route (G0/1).



9.2.1 The routing table

A router's routing table is a database of destinations known by the router. It can be thought of as a set of instructions:

- To send a packet to destination X, send the packet to next hop Y.
- Or, if the destination is in a directly connected network, forward the packet directly to the destination.
- Or, if the destination is the router's own IP address, continue to de-encapsulate the message (don't forward the packet).

The example we saw in figure 9.3 is an example of the second kind of instruction above; the destination of the packet (PC3, 192.168.2.11) is in a network directly connected to R1 (192.168.2.0/24), so R1 forwards the packet directly to the destination (by encapsulating it in a framed addressed to PC3).

Unlike switches, which can build their MAC address table automatically without any configuration, a router's routing table will be empty by default - it will not be able to forward packets. The following example shows R1's routing table before any configuration - the command to view the routing table is show ip route.

```
R1# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR
Gateway of last resort is not set
```

Without any configuration, the output shows some codes which represent the different route types that could appear in the routing table. Finally, Gateway of last resort is not set indicates that R1 doesn't have a default route, something we'll cover in section 9.4

Let's configure R1 and see how the output of show ip route changes. First, we won't actually configure any routes; rather, let's configure R1's IP addresses and enable its interfaces, as in the example below:

```
R1# configure terminal
R1(config)# interface g0/0
R1(config-if)# ip address 192.168.1.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# interface g0/1
R1(config-if)# ip address 192.168.2.1 255.255.255.0
R1(config-if)# no shutdown
```

R1's interfaces are now configured according to the previous diagrams: 192.168.1.1/24 on G0/0 and 192.168.2.1/24 on G0/1. Now let's examine R1's routing table again and see what has changed (omitting some of the codes to save space):

```
R1# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
. . .
Gateway of last resort is not set
     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, GigabitEthernet0/0
```

L 192.168.1.1/32 is directly connected, GigabitEthernet0/0

192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks

C 192.168.2.0/24 is directly connected, GigabitEthernet0/1

L 192.168.2.1/32 is directly connected, GigabitEthernet0/1

Just by configuring IP addresses on and enabling R1's two interfaces, R1 has inserted four routes (highlighted in bold) into its routing table: two connected routes (indicated by code C), and two local routes (indicated by code L).

Note

The line "192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks", is not a route. This statement means that, in the routing table, there are two routes to subnets that fit within the 192.168.1.0/24 class C network, with two different netmasks (/24 and /32). The same applies for the similar line about 192.168.2.0/24. We will cover subnets in chapter 11.

Connected routes

A *connected route* is a route to the network an interface is connected to. One connected route is automatically added to the routing table for each interface that has an IP address and is in an up/up state (you can check the interface's state with show ip interface brief). For example, R1's G0/0 interface has IP address 192.168.1.1/24, so it automatically adds a route to the 192.168.1.0/24 network in its routing table.

192.168.1.0/24 is the network address, with a host portion of all 0s. Because the netmask of the interface is /24, to find the network address you can simply change the final octet (the last 8 bits) of the address to 0.

A connected route will state that the network is directly connected, and it will also state which interface it is connected to. To view only the connected routes in R1's routing table, in the following example I filter the output using the pipe (|) followed by include C, to display only lines that include C.

R1# show ip route | include C

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP

C 192.168.1.0/24 is directly connected, GigabitEthernet0/0

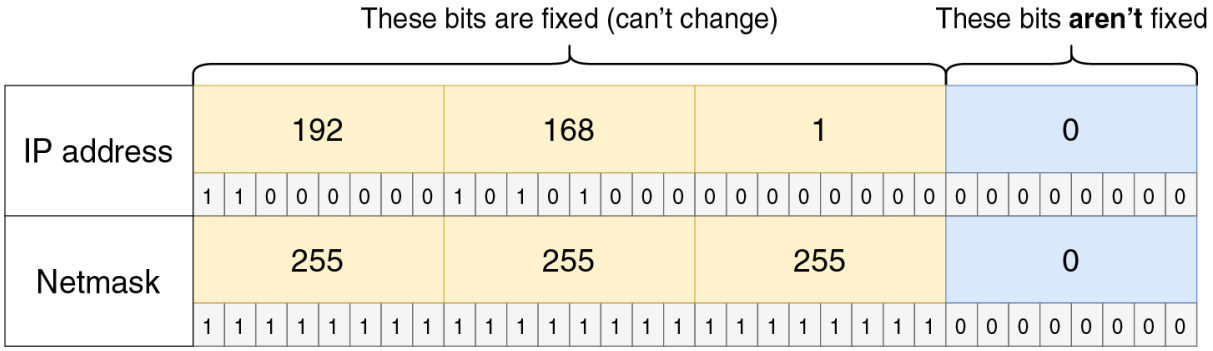
C 192.168.2.0/24 is directly connected, GigabitEthernet0/1

With these routes in its routing table, R1 knows that, to forward a packet to a host with an IP address in the 192.168.1.0/24 or 192.168.2.0/24 networks, it should send the packet out of the interface specified in the route, in a frame addressed directly to the destination host. We saw this in figure 9.3; the destination IP address of the packet was 192.168.2.11 (PC3), so R1 forwarded the packet out of the G0/1 interface, in a frame addressed to PC3's MAC address.

Figure 9.4 shows how the route to 192.168.1.0/24 includes all IP addresses from 192.168.1.0 through 192.168.1.255. The network portion of the route's address is fixed, but the host portion can be any eight-bit number.

Figure 9.4 The IP address and prefix length (written as a netmask) of the route to 192.168.1.0/24. Due to the /24 prefix length, the first three octets are fixed (the bits can't

change). However, the last octet can be any 8-bit number: .1, .11, .100, .179, etc. This means that any packet with a destination IP address beginning with 192.168.1 can be forwarded using this route.



Exam Tip

A route to more than one destination IP address is called a *network route*; it's a route to a network, rather than a route to a single destination IP address. A connected route is an example of a network route. The term "network route" is explicitly mentioned in exam topic 3.3.b, so remember that definition.

Local routes

A *local route* is a route to the exact IP address configured on the router's interface. Like connected routes, one local route is automatically added to the routing table for each interface that has an IP address and is in an up/up state. In the following example, I use show ip route | include L to view only R1's local routes. Note that, like connected routes, local routes also state X is directly connected, followed by the interface.

R1# show ip route | include L

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP

. . .

L 192.168.1.1/32 is directly connected, GigabitEthernet0/0

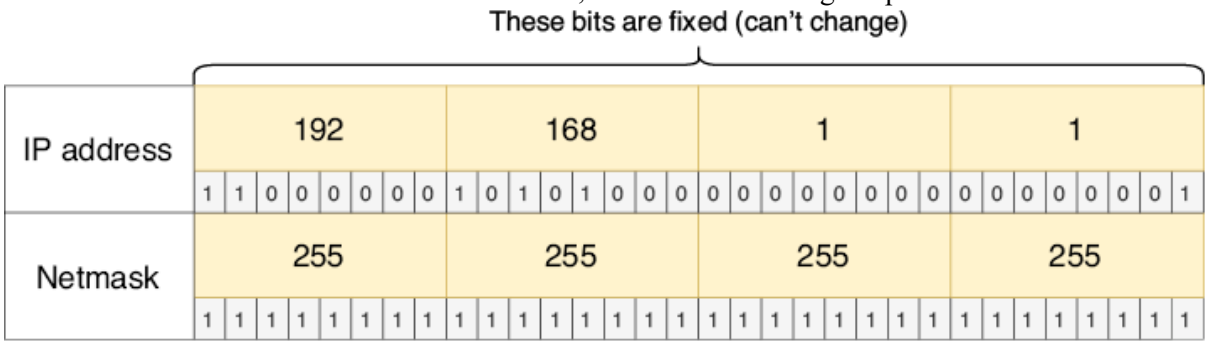
L 192.168.2.1/32 is directly connected, GigabitEthernet0/1

To specify the exact IP address of the interface, a local route uses a /32 prefix length; all bits of the netmask are set to "1". This is the case regardless of the netmask configured on the interface. For example, even though R1's interfaces both have /24 prefix lengths, their local routes are /32. The reason for this is that a local route specifies only a single IP address. As stated previously, a route to 192.168.1.0/24 includes all IP addresses from 192.168.1.0 through 192.168.1.255; because the prefix length is /24, the final octet can be any number from 0 to 255. On the other hand, a route to 192.168.1.1/32 includes only 192.168.1.1, due to its /32 prefix length; all bits are considered part of the network portion, and cannot be changed. Figure 9.5 shows the IP address of R1 G0/0 with a /32 prefix length (written as a netmask).

Figure 9.5 The IP address of R1's G0/0 interface and a /32 prefix length written as a netmask in dotted decimal and binary. With a prefix length of /32, the route only includes a single IP address (192.168.1.1, in this case). A packet destined for 192.168.1.2, for example, cannot use this route.

A local route tells the router that packets destined to the IP address specified in the route are for the router itself; it should continue to de-encapsulate the message and examine its contents. In this case,

the router does not forward the packet; it just receives the packet for itself. The local route is necessary to distinguish the router's own IP address from other IP addresses in the connected network. If R1 only had a connected route to 192.168.1.0/24, but no local route, it would forward packets destined for 192.168.1.1 out of its G0/0 interface, rather than receiving the packets for itself.



Exam Tip

A route to a single destination IP address (with a /32 prefix length) is called a *host route*; it's a route to a single host. A local route is an example of a host route. This is in contrast to a network route, which we covered earlier; a network route is any route with a prefix length shorter than /32. The term "host route" is explicitly mentioned in exam topic 3.3.c, so remember that definition.

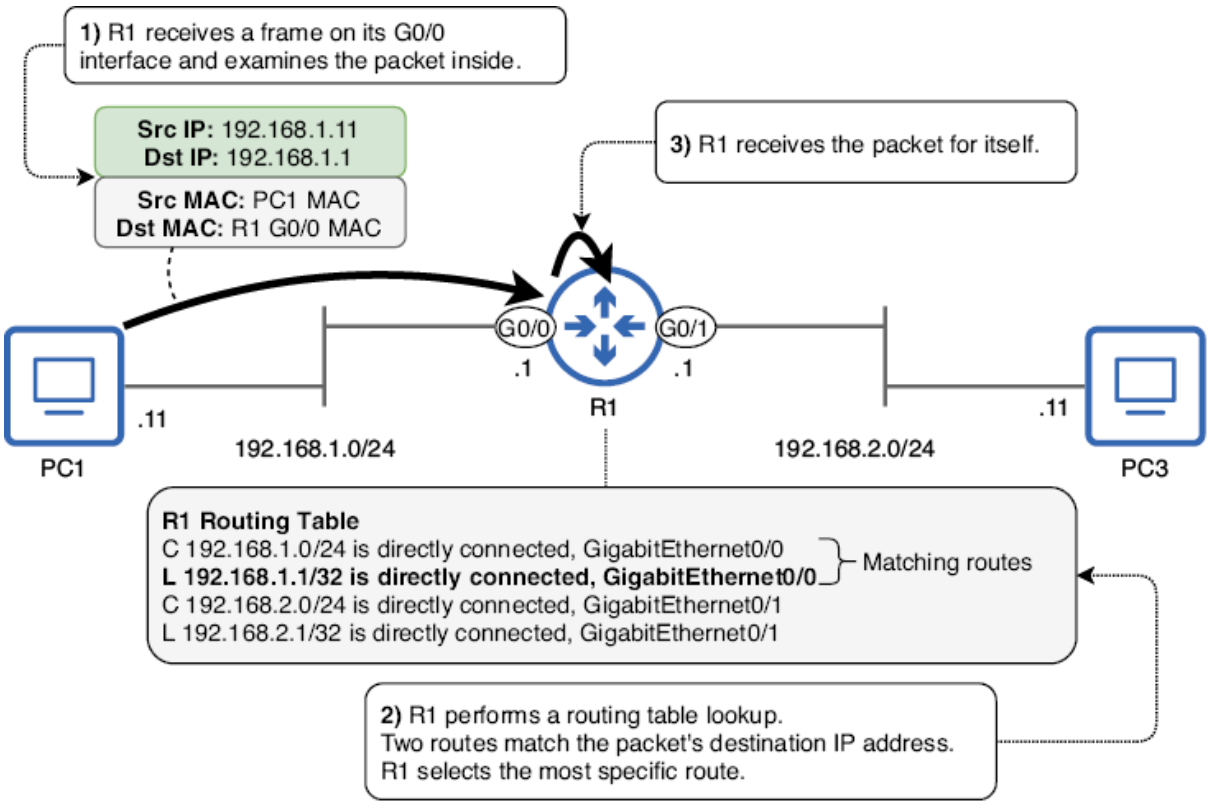
9.2.2 Route selection

When a router forwards a packet, it has to decide which route in its routing table it will use to forward the packet, and this is called *route selection*. To determine how to forward a particular packet, the router will select the *most specific matching route*. Let's define that term:

- *Matching route*: The packet's destination IP address is part of the network specified in the route. If not, the packet can't be forwarded using this route.
- *Most specific*: The route with the longest prefix length.

Let's use an example to clarify that concept. Figure 9.6 shows the route selection process when R1 receives a packet addressed to 192.168.1.1. The packet's destination IP address matches two routes in R1's routing table, so it selects the more specific of the two.

Figure 9.6 R1 receives a packet and selects the best route for that packet. (1) R1 receives a frame on its G0/0 interface. The destination MAC is its own, so it de-encapsulates it and examines the packet inside. (2) The packet's destination IP address is 192.168.1.1. R1 performs a routing table lookup and finds that two routes match the packet's destination IP address: the connected route to 192.168.1.0/24 and the local route to 192.168.1.1/32. R1 selects the most specific route: 192.168.1.1/32. (3) Because R1 selects a local route, it receives the packet for itself; it does not forward the packet.



A route with a /24 prefix length includes 256 different IP addresses. For example, 192.168.1.0/24 includes 192.168.1.0 through 192.168.1.255. On the other hand, a route with a /32 prefix length includes only a single IP address, so a /32 route is more specific than a /24 route. In fact, a /32 route is the most specific route possible; if a packet's destination IP address matches a /32 route, that route will always be selected for that packet, regardless of how many other matching routes there are.

Exam Tip

Be aware of this major difference between layer 3 forwarding done by routers, and layer 2 forwarding done by switches: When a router looks up a packet's destination IP address in its routing table, it looks for the most specific matching route. On the other hand, when a switch looks up a frame's destination MAC address in its MAC address table, it looks for an exact match; partial matches don't count.

What happens if there aren't any routes in the routing table that match a packet's destination IP address? In that case, the router will drop the packet; it won't flood it out of all ports like switches do with unknown unicast frames. A switch sometimes floods frames, but a router never floods packets; it either forwards the packet, receives the packet for itself, or drops the packet. Table 9.1 summarizes the actions a router can take on a packet:

Table 9.1 Actions a router can take on a packet (view table figure)

Matching conditions	Router's action
The packet's destination IP address matches one or more nonlocal routes.	Forward the packet according to the most specific matching route
The packet's destination IP address matches a local route.	Receive the packet for itself

Matching conditions	Router's action
The packet's destination IP address does not match any routes.	Drop the packet

9.3 Static routing

The packet forwarding process outlined in section 9.2 is called routing, but the term "routing" is also used to refer to the processes routers use to learn routes. In addition to the connected and local routes a router automatically inserts into its routing table, there are two main methods by which routers can learn routes:

- Dynamic routing: Routers use *dynamic routing protocols* (ie. OSPF) to share information with each other and build their routing tables.
- Static routing: An engineer/admin manually configures routes on the router.

Connected routes allow the router to forward packets to destinations in networks directly connected to the router, and local routes allow the router to receive packets destined for its own IP addresses. However, to forward packets toward destinations that are not in directly-connected networks, the router must learn of those destinations using one of the above methods (we will cover dynamic routing in chapters 17 and 18).

When forwarding a packet toward a destination that is not directly connected to the router, it must encapsulate the packet in a frame addressed to the MAC address of the *next hop*, which is the next router in the path to the destination. Figure 9.7 demonstrates this process.

Figure 9.7 PC1 sends a packet to PC3 via R1, R2, and R3. (1) PC1 sends the packet in a frame to R1 G0/1. R1 receives it and performs a routing table lookup. (2) R1 forwards the packet in a frame to R2 G0/0. R2 receives it and performs a routing table lookup. (3) R2 forwards the packet in a frame to R3 G0/0's MAC. R3 receives the packet and performs a routing table lookup. (4) R3 forwards the packet in a frame to PC3, which receives and processes it.

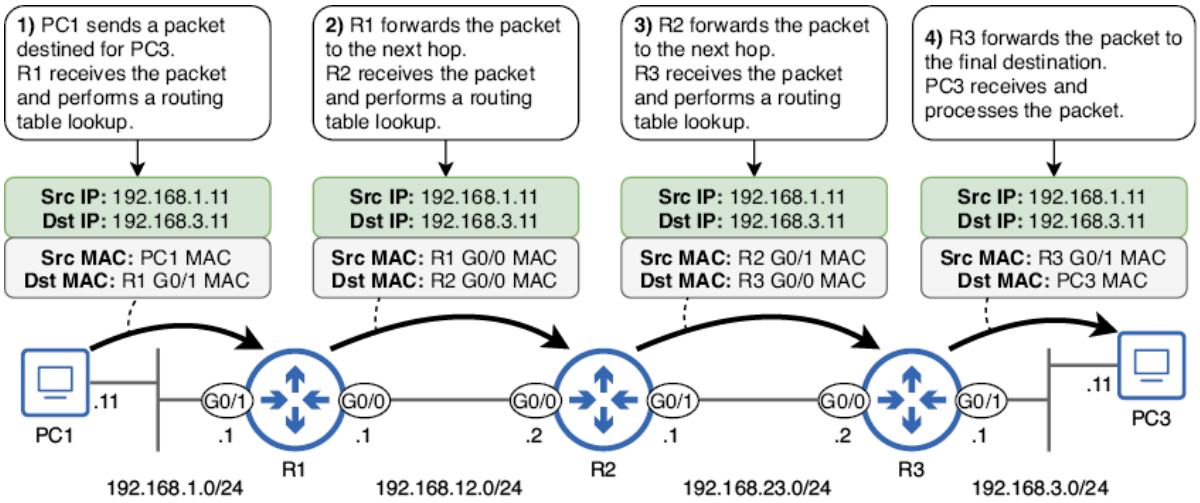


Figure 9.7 shows how routers forward packets toward remote (not directly connected) destinations. However, routers have no such routes in their routing tables by default; those routes must be manually configured. Without configuring any static routes, if R1 receives a packet from PC1 destined for 192.168.3.11, R1 won't find any matching routes when it performs the routing table lookup; it will

have no choice but to drop the packet. Table 9.2 lists the networks that each router in figure 9.7 is aware of without configuring static routes.

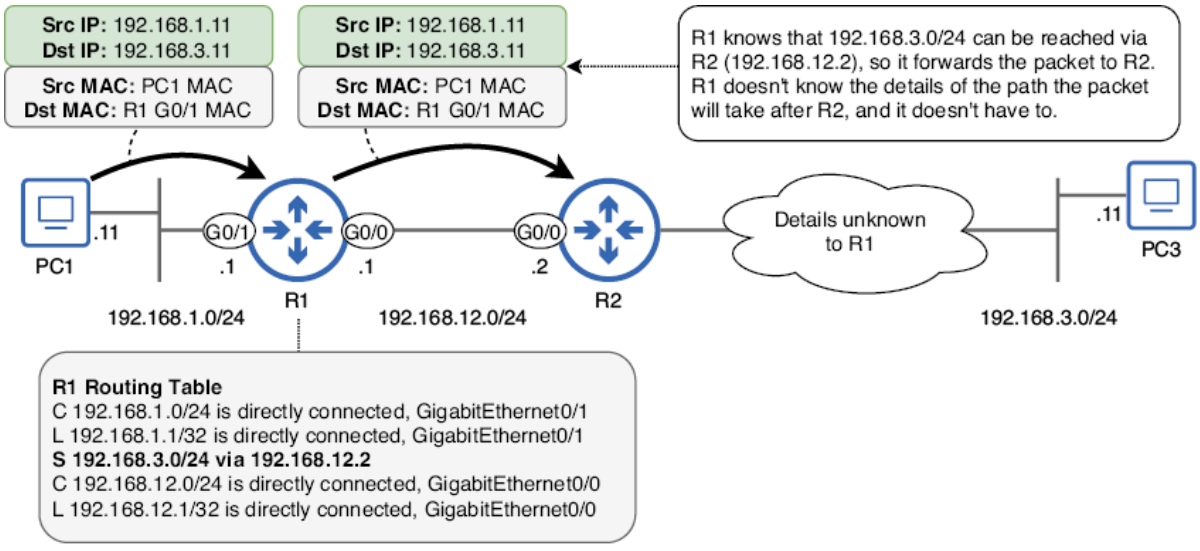
Table 9.2 Each router's known and unknown networks (view table figure)

Router	Known networks	Unknown networks
R1	192.168.1.0/24 192.168.12.0/24	192.168.3.0/24 192.168.23.0/24
R2	192.168.12.0/24 192.168.23.0/24	192.168.1.0/24 192.168.3.0/24
R3	192.168.3.0/24 192.168.23.0/24	192.168.1.0/24 192.168.12.0/24

Given the goal of enabling two-way communication between PC1 and PC3, which routes do we have to configure? Looking at table 9.2, you might assume that we have to configure six routes, so that each router knows about all networks within the greater network.

However, to forward packets between two hosts, each router only needs routes to the networks of the communicating hosts (PC1 and PC3). R1, for example, doesn't need to know about the network between R2 and R3 (192.168.23.0/24); R1 only needs to know that, to forward a packet toward a destination in 192.168.3.0/24, it should forward the packet to R2. Figure 9.8 demonstrates this concept; if we configure a route to 192.168.3.0/24 on R1, with R2's IP address specified as the next hop, R1 will be able to forward the packet to R2. It doesn't need to know the details of the path the packet will take after R2, it just needs to know that R2 is the next hop.

Figure 9.8 R1 forwards a packet destined for 192.168.3.11 (PC3). R1 has a static route to 192.168.3.0/24 via 192.168.12.2 (R2). R1 knows that to forward a packet toward the 192.168.3.0/24 network, it should forward the packet to R2. R1 doesn't know the details of the path the packet will take after R2, and it doesn't have to.



To summarize: each router needs a route to 192.168.3.0/24 so that it can forward packets from PC1 to PC3, and a route to 192.168.1.0/24 so that it can forward packets from PC3 to PC1. R1 already has a connected route to 192.168.1.0/24, and R3 already has a connected route to 192.168.3.0/24. Table 9.3 lists the static routes that we must configure to enable two-way communication between PC1 and PC3.

Table 9.3 Routes required to enable communication between PC1 and PC3 (view table figure)

Router	Required routes	Next hop
R1	192.168.3.0/24	192.168.12.2 (R2 G0/0)
R2	192.168.1.0/24	192.168.12.1 (R1 G0/0)
	192.168.3.0/24	192.168.23.2 (R3 G0/0)
R3	192.168.1.0/24	192.168.23.1 (R2 G0/1)

Note

In this example, we are talking about the routes required to enable two-way communication between PC1 and PC3. Although not necessary for that purpose, there is nothing wrong with configuring a route to 192.168.23.0/24 on R1, and a route to 192.168.12.0/24 on R3.

9.3.1 Configuring static routes

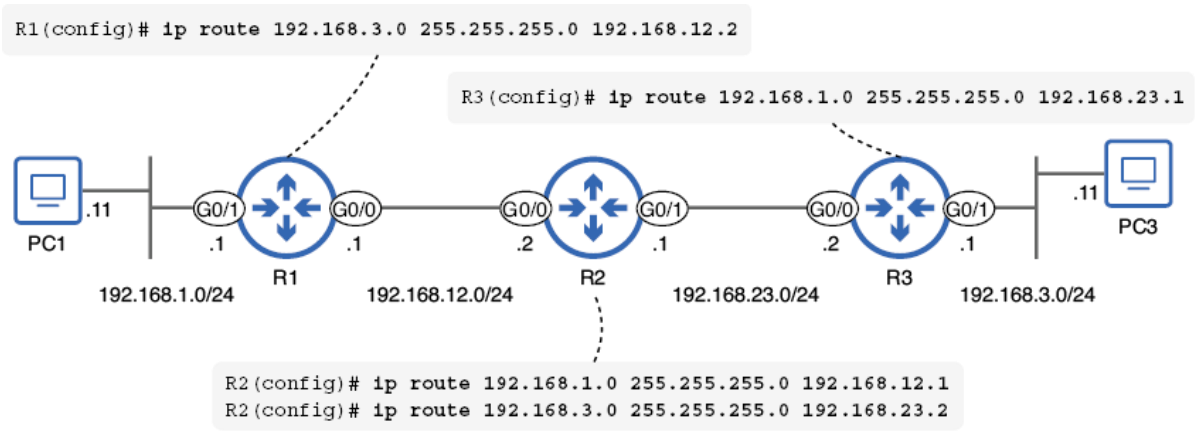
The command to configure a static route is, from global configuration mode, ip route. However, there are a few different options regarding the arguments you provide with the command:

- **ip route** *destination-network netmask next-hop*
- **ip route** *destination-network netmask exit-interface*
- **ip route** *destination-network netmask exit-interface next-hop*

Static routes specifying the next hop

A static route can be configured by specifying the destination network address, the netmask, and the IP address of the next hop. Figure 9.9 shows the commands to configure each of the necessary routes on R1, R2, and R3.

Figure 9.9 Configuring static routes on R1, R2, and R3 to enable two-way communication between PC1 and PC3. The routes specify the next-hop IP address. R1 requires a route to 192.168.3.0/24, R2 requires routes to 192.168.1.0/24 and 192.168.3.0/24, and R3 requires a route to 192.168.1.0/24.



A static route that specifies only the next hop IP address is called a *recursive static route*. The reason for the name "recursive" is that the route necessitates multiple lookups in the routing table to forward a packet:

- A lookup to find the IP address of the next hop.
- A lookup to find which interface the next hop is connected to.

To demonstrate that, let's look at R1's routing table in the example below. When R1 receives a packet destined for 192.168.3.11, it finds that the most-specific matching route (actually, the only matching route) is the static route, as highlighted in bold:

R1# **show ip route**

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP

. . .

192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks

C 192.168.1.0/24 is directly connected, GigabitEthernet0/1

L 192.168.1.1/32 is directly connected, GigabitEthernet0/1

S 192.168.3.0/24 [1/0] via 192.168.12.2

192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks

C 192.168.12.0/24 is directly connected, GigabitEthernet0/0

L 192.168.12.1/32 is directly connected, GigabitEthernet0/0

Note

The [1/0] in the static route indicates the *administrative distance* (AD) and *metric* of the route, respectively. AD and metric will be covered in chapter 17; they are not relevant to this chapter.

The static route states "S 192.168.3.0/24 [1/0] via 192.168.12.2" (note the code S for static), but that information alone doesn't tell R1 which interface to forward the packet out of. To learn that, it then performs a second lookup for the next hop IP address: 192.168.12.2. The most-specific (and only) matching route for 192.168.12.2 is the connected route to 192.168.12.0/24, which specifies the G0/1 interface. Now, after two lookups, R1 knows the next hop IP address and the interface to forward the packet out of.

Note: R1 knows the next hop IP address, but the information R1 really needs is the next hop MAC address. To learn that, it must send an ARP request to the next hop IP address.

Static routes specifying the exit interface

Rather than specifying the next hop IP address of the route, you can specify the *exit interface* - the interface the router should forward packets out of. The following example shows the same static routes as we saw in figure 9.9, but configured using the exit interface.

- R1(config)# ip route 192.168.3.0 255.255.255.0 g0/0
- R2(config)# ip route 192.168.1.0 255.255.255.0 g0/0
- R2(config)# ip route 192.168.3.0 255.255.255.0 g0/1
- R3(config)# ip route 192.168.1.0 255.255.255.0 g0/0

R1# show ip route

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP

. . .

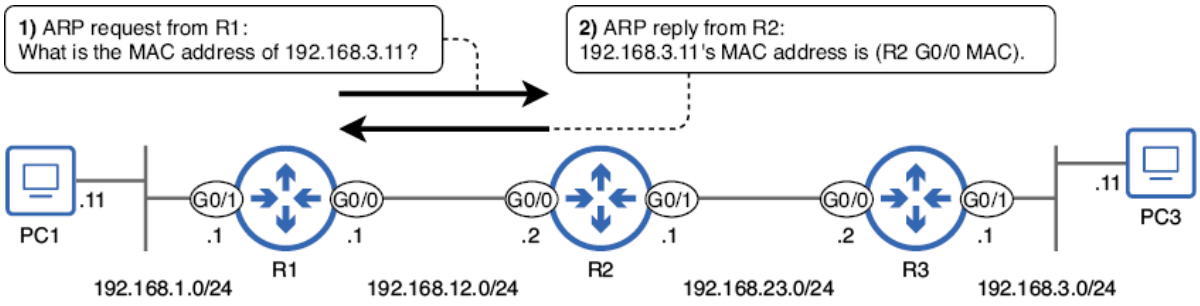
S 192.168.3.0/24 is directly connected, GigabitEthernet0/0

. . .

Note

A router will only use proxy ARP to reply to an ARP request if it has a route to the destination in its routing table. Otherwise, it will ignore the request.

Figure 9.10 A proxy ARP exchange between R1 and R2. (1) R1 sends an ARP request to learn PC3's MAC address. (2) 192.168.3.11 isn't R2's IP address, but R2 has a route to 192.168.3.0/24 in its routing table, so R2 uses proxy ARP to reply on behalf of PC3.



Note

The reliance on proxy ARP is a downside to directly-connected static routes for two reasons. First, although proxy ARP is enabled on Cisco routers by default, in some cases it might be disabled (for example, if R2 is not a Cisco router). If proxy ARP is disabled on R2, it won't reply to R1's ARP request, and R1 won't be able to forward the packet to PC3.

The second downside is that R1 will need to make a separate ARP entry for every host in 192.168.3.0/24. It thinks each host in 192.168.3.0/24 is directly connected, so it will try to learn each host's MAC address; that could waste memory on R1 if there are a lot of hosts in the network. On the other hand, if the next hop IP address is specified instead of the exit interface, R1 will only need one ARP entry to forward packets to 192.168.3.0/24: an ARP entry for the next hop.

R1# show arp

Protocol Address Age (min) Hardware Addr Type Interface

. . .

Internet 192.168.3.11 0 5254.0003.e684 ARPA GigabitEthernet0/0

. . .

Internet 192.168.12.2 0 5254.0003.e684 ARPA GigabitEthernet0/0

Although we are focusing on R1 as an example, the same applies for R2, which will think that the 192.168.1.0/24 and 192.168.3.0/24 networks are directly connected, as well as for R3, which will think that the 192.168.1.0/24 network is directly connected.

You should know the definition of and be able to configure directly-connected static routes, but because of the downsides of relying on proxy ARP, I recommend that you do not use them in a real network. Rather, use recursive static routes or the next option: fully-specified static routes.

Note

The third option when configuring a static route is to specify both the exit interface and the next hop, which is called a *fully-specified static route*. Below are the same four static routes, this time configured as fully-specified static routes:

R1(config)# ip route 192.168.3.0 255.255.255.0 g0/0 192.168.12.2 R2(config)# ip route 192.168.1.0 255.255.255.0 g0/0 192.168.12.1 R2(config)# ip route 192.168.3.0 255.255.255.0 g0/1 192.168.23.2 R3(config)# ip route 192.168.1.0 255.255.255.0 g0/0 192.168.23.1

Static routes specifying both the exit interface and next hop

R1# show ip route Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP . . . S 192.168.3.0/24 [1/0] via 192.168.12.2, GigabitEthernet0/0 . . .

- R1(config)# **ip route 192.168.3.0 255.255.255.0 g0/0 192.168.12.2**
- R2(config)# **ip route 192.168.1.0 255.255.255.0 g0/0 192.168.12.1**
- R2(config)# **ip route 192.168.3.0 255.255.255.0 g0/1 192.168.23.2**
- R3(config)# **ip route 192.168.1.0 255.255.255.0 g0/0 192.168.23.1**

More specific routes can be used for destinations in the internal corporate network, and then all other traffic (that doesn't match any other routes) will be routed using the default route. Figure 9.11 shows an example of this: R1 has specific routes to 192.168.2.0/24 and 192.168.3.0/24, and then a default route to the Internet; packets with destinations that don't match 192.168.2.0/24 or 192.168.3.0/24 (or R1's connected and local routes) will be forwarded using the default route.

R1# show ip route

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP

. . .

S 192.168.3.0/24 [1/0] via 192.168.12.2, GigabitEthernet0/0

. . .

To configure a default route, specify a destination network of 0.0.0.0 and a netmask of 0.0.0.0 in the ip route command; that results in 0.0.0.0/0, which includes all possible IP addresses. If a router does not have a default route configured, you will see the statement Gateway of last resort is not set above the routes in the routing table, as shown in the example below. This means the router does not have a default route.

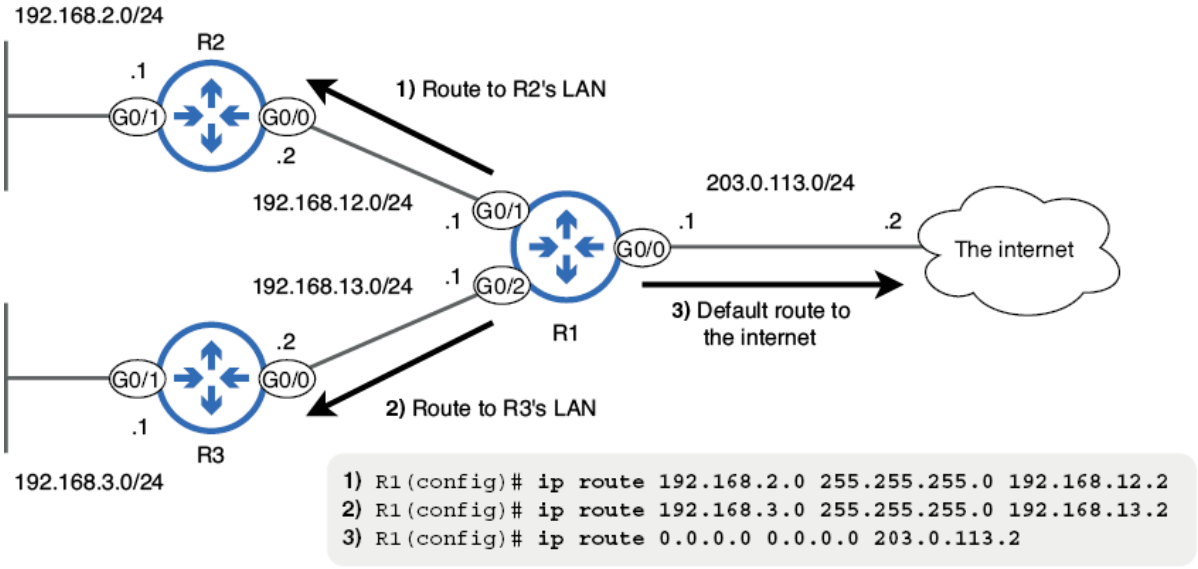
9.3.2 Configuring a default route

Note

"Gateway of last resort" is another term for the default gateway. The default route on a router is like a PC's default gateway; it's used to forward traffic to destinations outside of the router's known networks.

After configuring the static routes shown in figure 9.11, the output changes. In the following example, I use the show ip route static command to view only R1's static routes. Note that the ISP's IP address (203.0.113.2) is now listed as the gateway of last resort.

Figure 9.11 R1 has two routes to specific destination networks and one default route to the internet. (1) A route to 192.168.2.0/24, with R2 G0/0 as the next hop. (2) A route to 192.168.3.0/24, with R3 G0/0 as the next hop. (3) A default route, with the ISP's IP address (203.0.113.2) as the next hop.



R1# show ip route

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP

. . .

Gateway of last resort is not set

. . .

Note

The term *routing* can refer to the process of forwarding packets between networks, and the process of building a routing table.

Hosts in the same network can send packets to each other without the use of a router. However, to send packets to destinations outside of the local network, a router is required.

R1# show ip route static

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP

. . .

ia - IS-IS inter area, * - candidate default, U - per-user static route

. . .

Gateway of last resort is 203.0.113.2 to network 0.0.0.0

S* 0.0.0.0/0 [1/0] via 203.0.113.2

S 192.168.2.0/24 [1/0] via 192.168.12.2

S 192.168.3.0/24 [1/0] via 192.168.13.2

Note

You can use the ipconfig command in the Windows Command Prompt to see information such as the PC's IP address, netmask, and default gateway.

Exam scenarios

For each interface that has an IP address and is in an up/up state, the router will automatically add two routes to its routing table: a connected route and a local route.

1. (multiple choice, multiple answers)

A *local route* is a route to the exact IP address configured on the interface. Local routes use a /32 prefix length to specify a single IP address. If a router receives a packet destined for the IP address of a local route, it means the packet is destined for the router itself; the router will receive the packet for itself, it will not forward it.

A. It is a network route.

B. It is a host route.

C. It is a recursive route.

D. It is a directly connected route.

E. It is a fully specified route.

If there aren't any routes that match a packet's destination IP address, the router will drop the packet.

2. (drag and drop)

To forward a packet toward a remote destination, the router will encapsulate the packet in a frame destined for the MAC address of the *next hop* - the next router in the path to the destination.

(A) `ip route 192.168.1.0 255.255.255.0 GigabitEthernet0/1`	Recursive
(B) `ip route 0.0.0.0 0.0.0.0 203.0.113.120`	Directly connected
(C) `ip route 172.20.0.0 255.255.0.0 GigabitEthernet0/1 192.168.2.1`	Fully specified
(D) `ip route 192.0.2.0 255.255.255.0 172.16.25.209`	

The command to configure a static route is ip route *destination-network netmask* {*next-hop* | *exit-interface* | *exit-interface next-hop*}.

3. (lab simulation)

A static route that specifies only the exit interface is called a *directly-connected static route*, because it causes the router to treat the network as a directly connected network.

Summary

- The term *routing* can refer to the process of forwarding packets between networks and the process of building a routing table.
- Hosts in the same network can send packets to each other without the use of a router. However, to send packets to destinations outside of the local network, a router is required.
- The router a host will send packets destined for external networks to is called the *default gateway*. The host will send the packets in frames addressed to the default gateway's MAC address.
- A host's default gateway can be manually configured or automatically learned via DHCP.
- You can use the **ipconfig** command in the Windows Command Prompt to see information such as the PC's IP address, netmask, and default gateway.
- The routing table is the router's database of known destinations. It is a set of instructions about what action to take on packets. The routing table can be viewed with **show ip route**.
- For each interface that has an IP address and is in an up/up state, the router will automatically add two routes to its routing table: a connected route and a local route.
- A connected route is a route to the network that an interface is connected to. Connected routes are indicated by code C in the routing table. If a router receives a packet destined for a host in a directly connected network, it will forward the packet directly to the destination host (in a frame addressed to the host's MAC address).
- A *local route* is a route to the exact IP address configured on the interface. Local routes use a /32 prefix length to specify a single IP address. If a router receives a packet destined for the IP address of a local route, it means the packet is destined for the router itself; the router will receive the packet for itself—it will not forward it.
- A route to more than one destination IP address (any route with a prefix length shorter than /32) is called a *network route*. A connected route is an example of a network route.
- A route to a single destination IP address (a route with a /32 prefix length) is called a *host route*. A local route is an example of a host route.

- The process of deciding which route is appropriate for forwarding a packet is called *route selection*. To determine how to forward a particular packet, the router will select the *most specific matching route*—the matching route with the longest prefix length.
- A /32 route is the most specific route possible; it specifies only one IP address. A /0 route (default route) is the least specific route possible; it specifies every possible IP address.
- Whereas Layer 3 forwarding involves looking in the routing table for the most specific matching route, Layer 2 forwarding involves looking for an exact match in the MAC address table; partial matches don't count.
- If there aren't any routes that match a packet's destination IP address, the router will drop the packet.
- To route packets to destinations that aren't directly connected to the router, the router needs to learn routes to those destinations either via *dynamic routing* (using a protocol such as OSPF) or *static routing* (in which routes are manually configured on the router).
- To forward a packet toward a remote destination, the router will encapsulate the packet in a frame destined for the MAC address of the *next hop*—the next router in the path to the destination.
- For a router to forward packets between two hosts, the router needs routes to each host's network; it doesn't need routes to every network in the path between each destination.
- The command to configure a static route is **ip route** *destination-network netmask* {*next-hop* | *exit-interface* | *exit-interface next-hop*}.
- A static route that specifies only the next hop is called a *recursive static route*; it requires multiple lookups in the routing table to forward a packet: one to find the next-hop IP address and one to find which interface the next hop is connected to.
- A static route that specifies only the exit interface is called a *directly connected static route* because it causes the router to treat the network as a directly connected network.
- Directly connected static routes require proxy ARP to function. Proxy ARP allows a router to reply to ARP requests on behalf of other hosts. Proxy ARP is enabled on Cisco routers by default but might not be enabled on other vendors' routers.
- A static route that specifies both the exit interface and the next hop is called a *fully specified static route*.
- A *default route* is a route to 0.0.0.0/0. Because it is the least specific route possible, it will only be used to forward packets that don't match any other routes in the routing table.
- If a router has a default route, it won't drop packets that don't match other routes; instead, it will forward those packets using the default route.
- The default route is often used to provide a route to the internet; it is not feasible for a router to learn specific routes to each possible destination over the internet.