This chapter covers the following subjects:

- Typical Features of OSI Layer 3
- IP Addressing Fundamentals
- Network Layer Utilities
- IP Routing and Routing Protocols

CHAPTER 5

Fundamentals of IP

The OSI model assigns the functions of path selection and logical addressing to the OSI network layer (Layer 3). Path selection includes the process of learning all the paths, or routes, in a network and then forwarding packets based on those paths or routes. Often the terms *path selection* and *routing* are used interchangeably. In most Cisco documentation and in this book, *routing* is the more popular term.

In this chapter, you will learn about the core concepts behind OSI Layer 3. Because CCNA focuses on TCP/IP, you also will learn about the main Layer 3 protocol used by TCP/IP—namely, the Internet Protocol (IP). This coverage includes IP addressing, IP routing, and some protocols useful to IP's effort to deliver packets end to end through a network.

"Do I Know This Already?" Quiz

The purpose of the "Do I Know This Already?" quiz is to help you decide whether you really need to read the entire chapter. If you already intend to read the entire chapter, you do not necessarily need to answer these questions now.

The 12-question quiz, derived from the major sections in the "Foundation Topics" portion of the chapter, helps you determine how to spend your limited study time.

Table 5-1 outlines the major topics discussed in this chapter and the "Do I Know This Already?" quiz questions that correspond to those topics.

Table 5-1 *"Do I Know This Already?" Foundation Topics Section-to-Question Mapping*

Foundations Topics Section	Questions Covered in This Section
Typical Features of OSI Layer 3	1, 2, 4, 12
IP Addressing Fundamentals	5–9
Network Layer Utilities	10, 11
IP Routing and Routing Protocols	3

NOTE The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark this question wrong for purposes of the self-assessment. Giving yourself credit for an answer that you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. Which of the following describes the functions of OSI Layer 3 protocols?
   a. Logical addressing
   b. Physical addressing
   c. Path selection
   d. Arbitration
   e. Error recovery
2. Imagine that PC1 needs to send some data to PC2, and PC1 and PC2 are separated by several routers. What are the largest entities that make it from PC1 to PC2?
   a. Frame
   b. Segment
   c. Packet
   d. L5PDU
   e. L3PDU
   f. L1PDU
3. Which of the following does a router normally use when making a decision about routing TCP/IP?
   a. Destination MAC address
   b. Source MAC address
   c. Destination IP address
   d. Source IP address
   e. Destination MAC and IP address
4. Imagine a network with two routers that are connected with a point-to-point HDLC serial link. Each router has an Ethernet, with PC1 sharing the Ethernet with Router1, and PC2 sharing an Ethernet with Router2. When PC1 sends data to PC2, which of the following is true?
   a. Router1 strips the Ethernet header and trailer off the frame received from PC1, never to be used again.

b. Router1 encapsulates the Ethernet frame inside an HDLC header and sends the frame to Router2, which extracts the Ethernet frame for forwarding to PC2.

c. Router1 strips the Ethernet header and trailer off the frame received from PC1, which is exactly re-created by R2 before forwarding data to PC2.

d. Router1 removes the Ethernet, IP, and TCP headers, and rebuilds the appropriate headers before forwarding the packet to Router2.

5. Which of the following are valid Class C IP addresses?

a. 1.1.1.1

b. 200.1.1.1

c. 128.128.128.128

d. 224.1.1.1

e. 223.223.223.255

6. What is the range for the values of the first octet for Class A IP networks?

a. 0 to 127

b. 0 to 126

c. 1 to 127

d. 1 to 126

e. 128 to 191

f. 128 to 192

7. PC1 and PC2 are on two different Ethernets that are separated by an IP router. PC1's IP address is 10.1.1.1, and no subnetting is used. Which of the following addresses could be used for PC2?

a. 10.1.1.2

b. 10.2.2.2

c. 10.200.200.1

d. 9.1.1.1

e. 225.1.1.1

f. 1.1.1.1

8. How many valid host IP addresses does each Class B network contain?
   a. 16,777,214
   b. 16,777,216
   c. 65,536
   d. 65,534
   e. 65,532
   f. 32,768
   g. 32,766
   h. 32,764
9. How many valid host IP addresses does each Class C network contain?
   a. 65,536
   b. 65,534
   c. 65,532
   d. 32,768
   e. 32,766
   f. 256
   g. 254
10. Which of the following protocols allows a client PC to discover the IP address of another computer, based on that other computer's name?
    a. ARP
    b. RARP
    c. DNS
    d. DHCP
    e. BOOTP
11. Which of the following protocols allow a client PC to request assignment of an IP address as well as learn its default gateway?
    a. ARP
    b. RARP
    c. DNS
    d. DHCP

12. Which term is defined by the following phrase: "the type of protocol that is being forwarded when routers perform routing."

    a. Routed protocol

    b. Routing protocol

    c. RIP

    d. IOS

    e. Route protocol

The answers to the "Do I Know This Already?" quiz are found in Appendix A, "Answers to the 'Do I Know This Already?' Quizzes and Q&A Sections." The suggested choices for your next step are as follows:

- **10 or less overall score**—Read the entire chapter. This includes the "Foundation Topics" and "Foundation Summary" sections and the "Q&A" section.
- **11 or 12 overall score**—If you want more review on these topics, skip to the "Foundation Summary" section and then go to the "Q&A" section. Otherwise, move to the next chapter.

Foundation Topics

OSI Layer 3–equivalent protocols use *routing* and *addressing* to accomplish their goals. The choices made by the people who made up addressing greatly affect how routing works, so the two topics are best described together.

This chapter begins with an overview of the functions of routing and network layer logical addressing. Following that, the text moves on to the basics of IP addressing, relating IP addressing to the OSI routing and addressing concepts covered in the first section. The chapter ends with an introduction to IP routing protocols.

Typical Features of OSI Layer 3

A protocol that defines routing and addressing is considered to be a network layer, or Layer 3, protocol. OSI does define a unique Layer 3 protocol called Connectionless Network Services (CLNS), but, as usual with OSI protocols, you rarely see it in networks today. However, you will see many other protocols that perform the OSI Layer 3 functions of routing and addressing, such as the Internet Protocol (IP), Novell Internetwork Packet Exchange (IPX), or AppleTalk Dynamic Data Routing (DDR).

The network layer protocols have many similarities, regardless of what Layer 3 protocol is used. In this section, network layer (Layer 3) addressing is covered in enough depth to describe IP, IPX, and AppleTalk addresses. Also, now that data link layer and network layer addresses have been covered in this book, this section undertakes a comparison between the two.

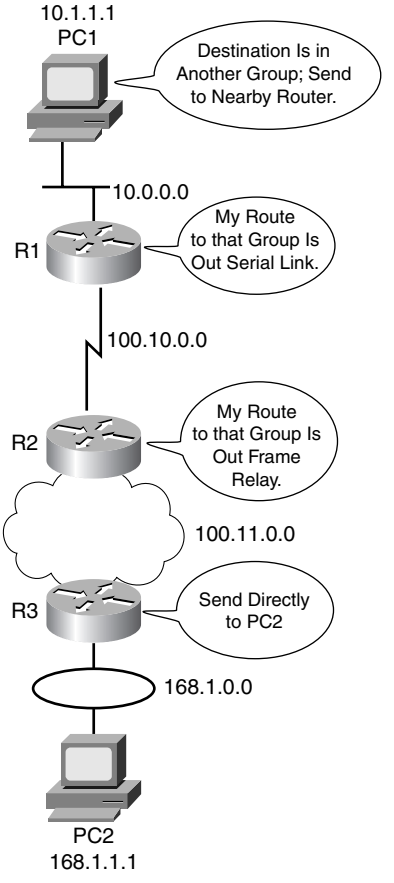
Routing (Path Selection)

Routing focuses on the end-to-end logic of forwarding data. Figure 5-1 shows a simple example of how routing works. The logic seen in the figure is relatively simple. For PC1 to send data to PC2, it must send something to R1, when sends it to R2, then on to R3, and finally to PC2. However, the logic used by each device along the path varies slightly.

PC1's Logic: Sending Data to a Nearby Router

In this example, PC1 has some data to send data to PC2. Because PC2 is not on the same Ethernet as PC1, PC1 needs to send the packet to a router that is attached to the same Ethernet as PC1. The sender sends a data-link frame across the medium to the nearby router; this frame includes the packet in the data portion of the frame. That frame uses data link layer (Layer 2) addressing in the data-link header to ensure that the nearby router receives the frame.

Figure 5-1 *Routing Logic: PC1 Sending to PC2*



The main point here is that the originator of the data does not know much about the network—just how to get the data to some nearby router. In the post office analogy, it's like knowing how to get to the local post office, but nothing more. Likewise, PC1 needs to know only how to get the packet to R1.

R1 and R2's Logic: Routing Data Across the Network

R1 and R2 both use the same general process to route the packet. The *routing table* for any particular network layer protocol contains a list of network layer address *groupings*. Instead of a single entry in the routing table per individual destination address, there is one entry per group. The router compares the destination network layer address in the packet to the entries in the routing table, and a match is made. This matching entry in the routing table tells this router where to forward the packet next. The words in the bubbles in Figure 5-1 point out this basic logic.

The concept of network layer address grouping is similar to the U.S. ZIP code system. Everyone living in the same vicinity is in the same ZIP code, and the postal sorters just look for the ZIP codes, ignoring the rest of the address. Likewise, in Figure 5-1, everyone in this network whose IP address starts with 168.1 is on the Token Ring on which PC2 resides, so the routers can just have one routing table entry that means "all addresses that start with 168.1."

Any intervening routers repeat the same process. The destination network layer (Layer 3) address in the packet identifies the group in which the destination resides. The routing table is searched for a matching entry, which tells this router where to forward the packet next. Eventually, the packet is delivered to the router connected to the network or subnet of the destination host (R3), as previously shown in Figure 5-1.

R3's Logic: Delivering Data to the End Destination

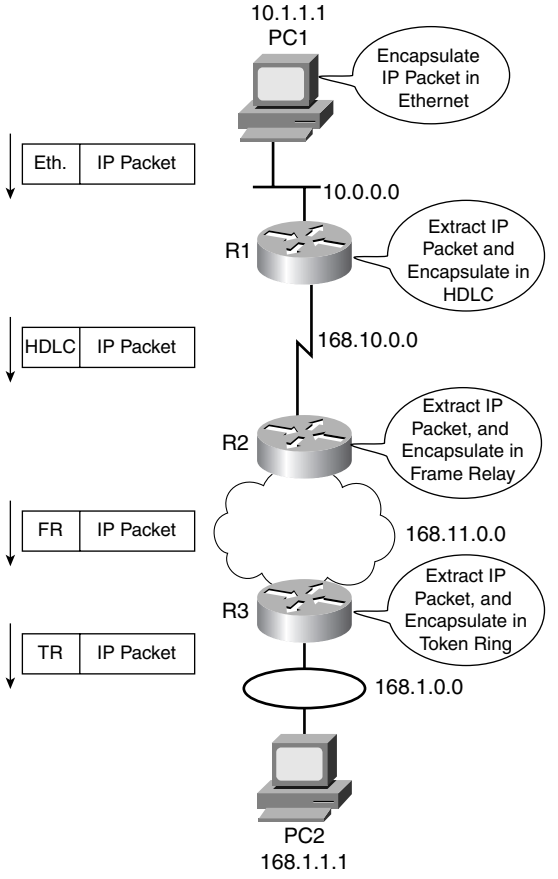
The final router in the path, R3, uses almost the exact same logic as R1 and R2, but with one minor difference. R3 needs to forward the packet directly to PC2, not to some other router. On the surface, that difference seems insignificant. In the next section, when you read about how the network layer uses the data link layer, the significance of the difference will become obvious.

Network Layer Interaction with the Data Link Layer

In Figure 5-1, four different types of data links were used to deliver the data. When the network layer protocol is processing the packet, it decides to send the packet out the appropriate network interface. Before the actual bits can be placed onto that physical interface, the network layer must hand off the packet to the data link layer protocols, which, in turn, ask the physical layer to actually send the data. And as was described in Chapter 3, "Fundamentals of Ethernet LANs," the data link layer adds the appropriate header and trailer to the packet, creating a frame, before sending the frames over each physical network.

The routing process forwards the packet, and only the packet, from end-to-end through the network, discarding data link headers and trailers along the way. The network layer processes deliver the packet end-to-end, using successive data-link headers and trailers just to get the packet to the next router or host in the path. Each successive data link layer just gets the packet from one device to the next. Figure 5-2 shows the same diagram as Figure 5-1 but includes the concepts behind encapsulation.

Figure 5-2 *Network Layer and Data Link Layer Encapsulation*



Because the routers build new data-link headers and trailers (trailers not shown in figure),and because the new headers contain data-link addresses, the PCs and routers must have some way to decide what data-link addresses to use. An example of how the router determines which data-link address to use is the IP Address Resolution Protocol (ARP). *ARP is used to dynamically learn the data-link address of an IP host connected to a LAN*. You will read more about ARP later in this chapter.

In short, the process of routing forwards Layer 3 packets, also called *Layer 3 protocol data units* (*L3 PDUs*), based on the destination Layer 3 address in the packet. The process uses the data link layer to encapsulate the Layer 3 packets into Layer 2 frames for transmission across each successive data link.

Network Layer (Layer 3) Addressing

One key feature of network layer addresses is that they were designed to allow logical grouping of addresses. In other words, something about the numeric value of an address implies a group or set of addresses, all of which are considered to be in the same grouping. In TCP/IP, this group is called a *network* or a *subnet*. In IPX, it is called a *network*. In AppleTalk, the grouping is called a *cable range*. These groupings work just like U.S.P.S. ZIP codes, allowing the routers (mail sorters) to speedily route (sort) lots of packets (letters).

Just like postal street addresses, network layer addresses are grouped based on physical location in a network. The rules differ for some network layer protocols, but the grouping concept is identical for IP, IPX, and AppleTalk. In each of these network layer protocols, all devices on opposite sides of a router must be in a different Layer 3 group, just like in the examples earlier in this chapter.

Routing relies on the fact that Layer 3 addresses are grouped together. The routing tables for each network layer protocol can have one entry for the group, not one entry for each individual address. Imagine an Ethernet with 100 TCP/IP hosts. A router needing to forward packets to any of those hosts needs only one entry in its IP routing table. This basic fact is one of the key reasons that routers can scale to allow tens and hundreds of thousands of devices. It's very similar to the U.S.P.S. ZIP code system—it would be ridiculous to have people in the same ZIP code live somewhere far away from each other, or to have next-door neighbors be in different zip codes. The poor postman would spend all his time driving and flying around the country! Similarly, to make routing more efficient, network layer protocols group addresses together.

With that in mind, most network layer (Layer 3) addressing schemes were created with the following goals:

- The address space should be large enough to accommodate the largest network for which the designers imagined the protocol would be used.
- The addresses should allow for unique assignment.
- The address structure should have some grouping implied so that many addresses are considered to be in the same group.
- Dynamic address assignment for clients is desired.

The U.S. Postal Service analogy also works well as a comparison to how IP network numbers are assigned. Instead of getting involved with every small community's plans for what to name new streets, the post service simply has a nearby office with a ZIP code. If that local town wants to add streets, the rest of the post offices in the country already are prepared because they just forward letters based on the ZIP code, which they already know. The only postal employees who care about the new streets are the people in the local post office. It is

the local postmaster's job to assign a mail carrier to deliver and pick up mail on any new streets.

Also, you can have duplicate local street addresses, as long as they are in different ZIP codes, and it all still works. There might be hundreds of Main streets in different ZIP codes, but as long as there is just one per ZIP code, the address is unique. Layer 3 network addresses follow the same concept—as long as the entire Layer 3 address is unique compared to the other Layer 3 addresses, all is well.

Example Layer 3 Address Structures

Each Layer 3 address structure contains at least two parts. One (or more) part at the beginning of the address works like the ZIP code and essentially identifies the grouping. All instances of addresses with the same value in these first bits of the address are considered to be in the same group—for example, the same IP subnet or IPX network or AppleTalk cable range. The last part of the address acts as a local address, uniquely identifying that device in that particular group. Table 5-2 outlines several Layer 3 address structures.

Table 5-2 *Layer 3 Address Structures*

Protocol	Size of Address in Bits	Name and Size of Grouping Field in Bits	Name and Size of Local Address Field in Bits
IP	32	Network or subnet (variable, between 8 and 30 bits)	Host (variable, between 2 and 24 bits)
IPX	80	Network (32)	Node (48)
AppleTalk	24	Network* (16)	Node (8)
OSI	Variable	Many formats, many sizes	Domain-specific part (DSP—typically 56, including NSAP)

*Consecutively numbered values in this field can be combined into one group, called a cable range.

Routing Protocols

Conveniently, the routing tables in the example based on Figure 5-2 had the correct routing information already in their routing tables. In most cases, these entries are built dynamically by use of a routing protocol. Routing protocols learn about all the locations of the network layer "groups" in a network and advertise the locations of the groups. As a result, each router can build a good routing table dynamically. Routing protocols define message formats and procedures, just like any other protocol. The end goal of each routing protocol is to fill the routing table with all known destination groups and with the best route to reach each group.

The terminology relating to routing protocols sometimes can get in the way. A *routing protocol* learns routes and puts those routes in a routing table. A *routed protocol* is the type of packet forwarded, or routed, through a network. In Figures 5-1 and 5-2, the figures represent how IP packets are routed, so IP would be the *routed protocol*. If the routers used the Routing Information Protocol (RIP) to learn the routes, then RIP would be the *routing protocol*.

Later in this chapter, the section titled "IP Routing Protocols" shows a detailed example of how routing protocols learn routes.

IP Addressing Fundamentals

No one reading this book should be shocked to hear that IP addressing is one of the most important topics for passing the the INTRO and ICND exams. In fact, IP addressing is the only major topic that is covered specifically on both the INTRO and ICND exams. Plus, you need a comfortable, confident understanding of IP addressing and subnetting for success on any Cisco certification. In other words, you had better know addressing and subnetting!

This section introduces IP addressing and subnetting, and also covers the concepts behind the struture of an IP address, including how it relates to IP routing. In Chapter 12, "IP Addressing and Subnetting," you will read about the math behind IP addressing and subnetting.

IP Addressing Definitions

If a device wants to communicate using TCP/IP, it needs an IP address. When the device has an IP address and the appropriate software and hardware, it can send and receive IP packets. Any device that can send and receive IP packets is called an *IP host*.

IP addresses consist of a 32-bit number, usually written in *dotted-decimal notation*. The "decimal" part of the term comes from the fact that each byte (8 bits) of the 32-bit IP address is converted to its decimal equivalent. The four resulting decimal numbers are written in sequence, with "dots," or decimal points, separating the numbers—hence the name *dotted-decimal*. For instance, 168.1.1.1 is an IP address written in dotted-decimal form, but the actual binary version is 10101000 00000001 00000001 00000001. (You almost never need to write down the binary version—but you will need to know how to convert between the two formats in Chapter 12, "IP Addressing and Subnetting.")

Each of the decimal numbers in an IP address is called an *octet*. The term *octet* is just a vendor-neutral term instead of *byte*. So, for an IP address of 168.1.1.1, the first octet is 168, the second octet is 1, and so on. The range of decimal numbers numbers in each octet is between 0 and 255, inclusive.

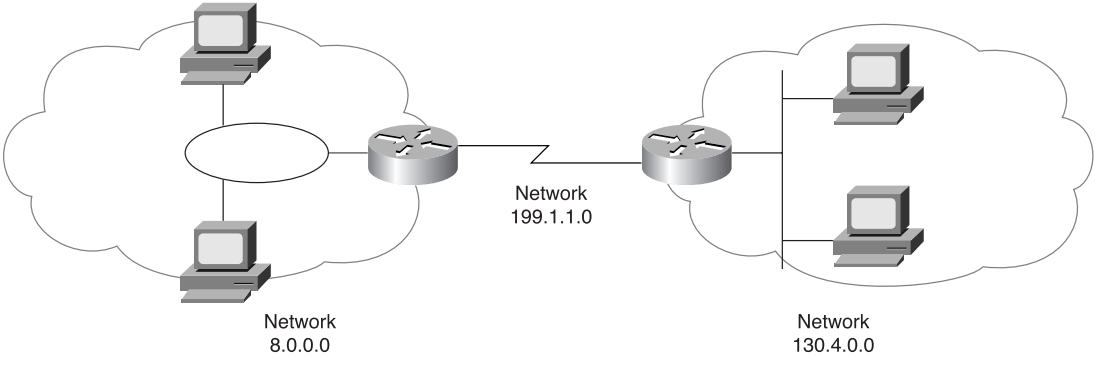
Finally, note that each network interface uses a unique IP address. Most people tend to think that their computer has an IP address, but actually their computer's network card has an IP address. If you put two Ethernet cards in a PC to forward IP packets through both cards, they both would need unique IP addresses. Similarly, routers, which typically have many network interfaces that forward IP packets, have an IP address for each interface.

Now that you have some idea about the basic terminology, the next section relates IP addressing to the routing concepts of OSI Layer 3.

How IP Addresses Are Grouped Together

To fully appreciate IP addressing, you first must understand the concepts behind the grouping of IP addresses. The first visions of what we call the Internet were for connecting research sites. A typical network diagram might have looked like Figure 5-3.

Figure 5-3 *Sample Network Using Class A, B, and C Network Numbers*



The conventions of IP addressing and IP address grouping make routing easy. For example, all IP addresses that begin with 8 are on the Token Ring on the left. Likewise, all IP addresses that begin with 130.4 are on the right. Along the same lines, 199.1.1 is the prefix on the serial link. By following this convention, the routers build a routing table with three entries, one for each prefix, or network number.

So, the general ideas about how IP address groupings can be summarized are as follows:

- All IP addresses in the same group must not be separated by a router.
- IP addresses separated by a router must be in different groups.

As mentioned earlier in this chapter, IP addressing behaves similarly to ZIP codes. Everyone living in my ZIP code lives in my town. If some members of my ZIP code were in California, some of my mail might be sent out there (I live in Georgia, by the way). Likewise, IP routing

counts on the fact that all IP addresses in the same subnet are in the same general location, with the routers in the network forwarding traffic to addresses in my subnet to a router connected to my subnet.

Classes of Networks

In Figure 5-3 and the surrounding text, I claimed that the IP addresses of devices attached to the Token Ring all started with 8 and that the IP addresses of devices attached to the Ethernet all started with 130.4. Why only one number for the "prefix" on the Token Ring and two numbers on the Ethernet? Well, it all has to do with IP address classes.

RFC 790 defines the IP protocol, including multiple different classes of networks. IP defines three different network classes, called A, B, and C, from which individual hosts are assigned IP addresses. TCP/IP defines Class D (multicast) addresses and Class E (experimental) addresses as well.

By definition, all addresses in the same Class A, B, or C network have the same numeric value *network* portion of the addresses. The rest of the address is called the *host* portion of the address.

Using the post office example, the network part of an IP address acts like the ZIP code, and the host part acts like the street address. Just as a letter-sorting machine three states away from you cares only about the ZIP code on a letter addressed to you, a router three hops away from you cares only about the network number that your address resides in.

Class A, B, and C networks each have a different length for the part that identifies the network:

- Class A networks have a 1-byte-long network part. That leaves 3 bytes for the rest of the address, called the host part.
- Class B networks have a 2-byte-long network part, leaving 2 bytes for the host portion of the address.
- Class C networks have a 3-byte-long network part, leaving only 1 byte for the host part.

For instance, Figure 5-3 lists network 8.0.0.0 next to the Token Ring. Network 8.0.0.0 is a Class A network, which means that only 1 byte is used for the network part of the address. So, all hosts in network 8.0.0.0 begin with 8. Similarly, Class B network 130.4.0.0 is listed next to the Ethernet; because it is Class B, 2 bytes define the network part, and all addresses begin with those same two bytes. When written down, network numbers have all decimal 0s in the host part of the number. So, Class A network "8" is written 8.0.0.0, Class B network 130.4 is written 130.4.0.0, and so on.

Now consider the size of each class of network. Class A networks need 1 byte for the network part, leaving 3 bytes, or 24 bits, for the host part. There are 2^{24} different possible values in the host part of a Class A IP address. So, each Class A network can have 2^{24} IP addresses—except for two reserved host addresses in each network, as shown in the last column of Table 5-3. The table summarizes the characteristics of Class A, B, and C networks.

Table 5-3 *Sizes of Network and Host Parts of IP Addresses with No Subnetting*

Any Network of This Class	Number of Network Bytes (Bits)	Number of Host Bytes (Bits)	Number of Addresses per Network*
A	1 (8)	3 (24)	$2^{24} - 2$
B	2 (16)	2 (16)	$2^{16} - 2$
C	3 (24)	1 (8)	$2^{8} - 2$

*There are two reserved host addresses per network.

Network numbers look like actual addresses because they are in dotted-decimal format. However, network numbers are not actually IP addresses because they cannot be assigned to an interface as an IP address. Conceptually, network numbers represent the group of all IP addresses in the network, much like a ZIP code represents the group of all addresses in a community. Based on the three examples from Figure 5-3, Table 5-4 provides a closer look at the numerical version of the three network numbers: 8.0.0.0, 130.4.0.0, and 199.1.1.0.

Table 5-4 *Example Network Numbers, Decimal and Binary*

Network Number	Binary Representation, with Host Part Bold
8.0.0.0	00001000 **00000000 00000000 00000000**
130.4.0.0	10000010 00000100 **00000000 00000000**
199.1.1.0	11000111 00000001 00000001 **00000000**

Two numbers inside each Class A, B, or C network are reserved, as mentioned at Table 5-3. One of the two reserved values is the network number itself. For instance, each of the numbers in Table 5-4 is reserved. The other reserved value is the one with all binary 1s in the host part of the address—this number is called the *network broadcast* or *directed broadcast* address. Also, because the network number is the lowest numerical value inside that network and the broadcast address is the largest, all the numbers between the network number and the broadcast address are the valid, useful IP addresses that can be used to address interfaces in the network.

The Actual Class A, B, and C Network Numbers

Many different Class A, B, and C networks exist. If your firm connects to the Internet, it must use registered, unique network numbers. To that end, the Network Information Center (NIC) assigns network numbers so that all IP address are unique. By assigning one company a particular network number, and not assigning that same network number to any other company, all IP addresses can be unique throughout the Internet. Table 5-5 summarizes the possible network numbers, the total number of each type, and the number of hosts in each Class A, B, and C network.

Table 5-5 *List of All Possible Valid Network Numbers**

Class	First Octet Range	Valid Network Numbers	Total Number of This Class of Network	Number of Hosts per Network
A	1 to 126	1.0.0.0 to 126.0.0.0	$2^7 - 2$	$2^{24} - 2$
B	128 to 191	128.1.0.0 to 191.254.0.0	$2^{14} - 2$	$2^{16} - 2$
C	192 to 223	192.0.1.0 to 223.255.254.0	$2^{21} - 2$	$2^8 - 2$

*The Valid Network Numbers column shows actual network numbers. There are several reserved cases. For example, networks 0.0.0.0 (originally defined for use as a broadcast address) and 127.0.0.0 (still available for use as the loopback address) are reserved. Networks 128.0.0.0, 191.255.0.0, 192.0.0.0, and 223.255.255.0 also are reserved.

Memorizing the contents of Table 5-5 should be one of the first things you do in preparation for the CCNA exam(s). Engineers should be able to categorize a network as Class A, B, or C with ease. Also memorize the number of octets in the network part of Class A, B, and C addresses, as shown in Table 5-4.

IP Subnetting

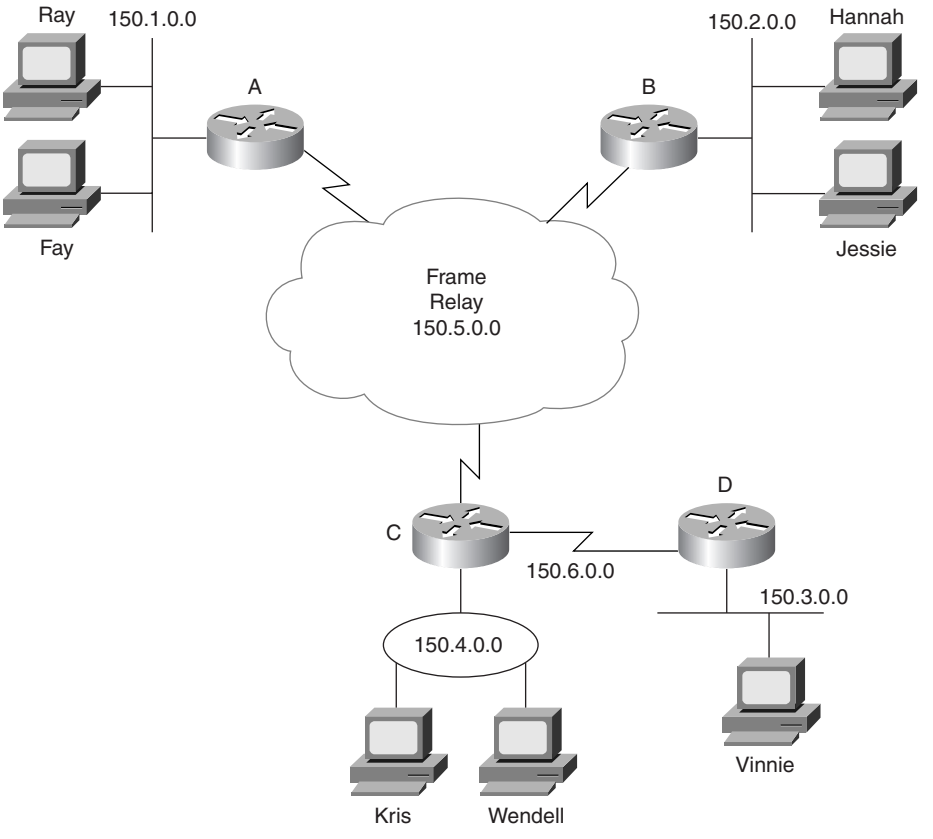
One of the most important topics on both the INTRO and ICND exams is the topic of subnetting. You need to know how it works and how to "do the math" to figure out issues when subnetting is in use, both in real life and on the exam.

Chapter 12 covers the details of subnetting concepts, motivation, and math, but you should have a basic understanding of the concepts before covering the topics between here and Chapter 12. So, this section describes the basics.

IP subnetting creates vastly larger numbers of smaller groups of IP addresses, compared with simply using Class A, B, and C conventions. The Class A, B, and C rules still exist—but now, a single Class A, B, or C network can be subdivided into many smaller groups. Subnetting treats a subdivision of a single Class A, B, or C network as if it were a network itself. By doing so, a single Class A, B, or C network can be subdivided into many nonoverlapping subnets.

Comparing a single network topology using subnetting with the same topology without subnetting drives home the basic concept. Figure 5-4 shows such a network, without subnetting.

Figure 5-4 *Backdrop for Discussing Numbers of Different Networks/Subnetworks*



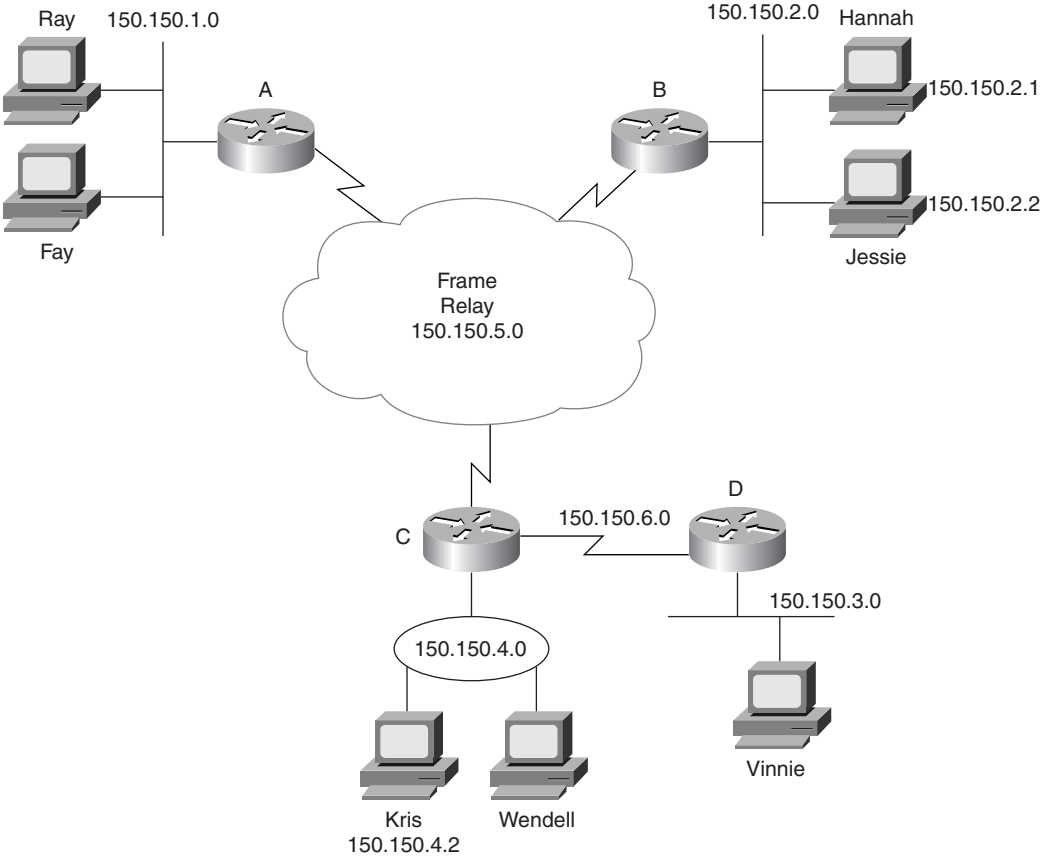
The design in Figure 5-4 requires six groups, each of which is a Class B network in this example. The four LANs each use a single Class B network. In other words, each of the LANs attached to routers A, B, C, and D is in a separate network. Additionally, the two serial interfaces composing the point-to-point serial link between routers C and D use the same network because these two interfaces are not separated by a router. Finally, the three router interfaces composing the Frame Relay network with routers A, B, and C are not separated by an IP router and would compose the sixth network.

Each Class B network has $2^{16} - 2$ hosts addresses in it—far more than you will ever need for each LAN and WAN link. In fact, this design would not be allowed if it were connected to the Internet. The NIC would not assign six separate registered Class B network numbers—

in fact, you probably would not even get one Class B network because most of the Class B addresses already are assigned. You more likely would get a couple of Class C networks, and the NIC would expect you to use subnetting.

Figure 5-5 illustrates a more realistic example that uses basic subnetting.

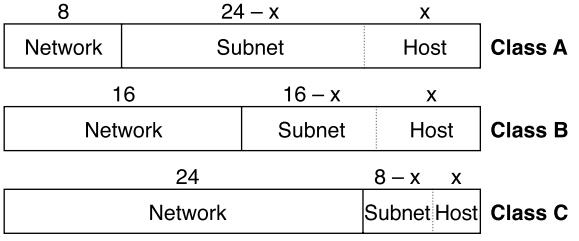
Figure 5-5 *Using Subnets*



As in Figure 5-4, the design in Figure 5-5 requires six groups. Unlike Figure 5-5, this figure uses six subnets, each of which is a subnet of a single Class B network. This design subnets Class B network 150.150.0.0, which has been assigned by the NIC. To perform subnetting,the third octet (in this example) is used to identify unique subnets of network 150.150.0.0. Notice that each subnet number in the figure shows a different value in the third octet, representing each different subnet number. In other words, this design numbers or identifies each different subnet using the third octet.

When subnetting, a third part of an IP address appears between the network and host parts of the address—namely, the subnet part of the address. This field is created by "stealing" or "borrowing" bits from the host part of the address. The size of the network part of the address never shrinks—in other words, Class A, B, and C rules still apply when defining the size of the network part of an address. The host part of the address shrinks to make room for the subnet part of the address. Figure 5-6 shows the format of addresses when subnetting.

Figure 5-6 *Address Formats When Subnetting Is Used*



Now, instead of routing based on the network part of an address, routers can route based on the combined network and subnet parts. In fact, most people do not even bother distinguishing between the network part and the subnet part—they just call both fields together the subnet part of an address.

Finally, IP addressing with subnetting uses a concept called a *subnet mask*. A subnet mask helps define the structure of an IP address, as shown in Figure 5-6. Chapter 12 explains the details.

Network Layer Utilities

The TCP/IP network layer uses several utility protocols to help it complete its task. For instance, in the first section of this chapter, you read that the Address Resolution Protocol (ARP) could be used to discover the MAC address of another IP host. In this section, you will learn about some basic IP utilities, using other protocols beside IP that together help IP deliver packets end to end through an IP network.

Address Resolution Protocol and the Domain Name System

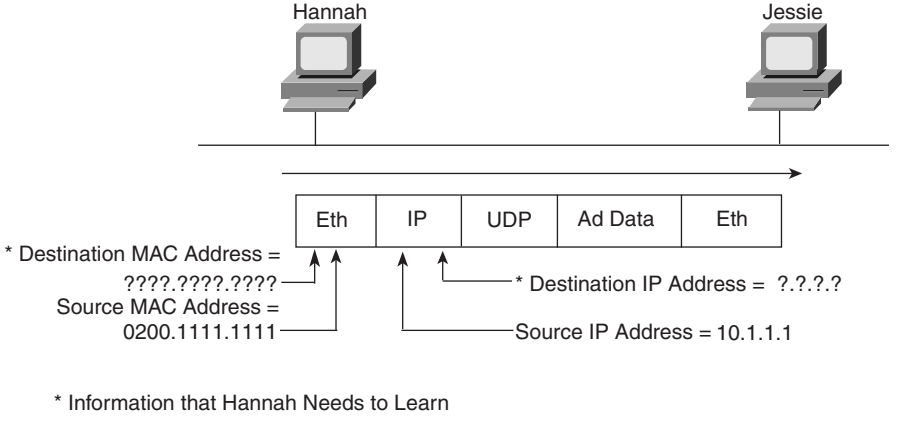
Network designers should try to make using the network as simple as possible. At most, users might want to remember the name of another computer with which they want to communicate, such as remembering the name of a web site. They certainly do not want to remember the IP address, nor do they want to try to remember any MAC addresses! So, TCP/

IP needs to have protocols that dynamically discover all the necessary information to allow communications, without the user knowing more than a name.

You might not even think that you need to know the name of another computer. For instance, when you open your browser, you probably have a default home page configured that the browser immediately downloads. You might not think of that URL string as a name, but the universal resource locator (URL) for the home page has a name embedded in it. For instance, in a URL such as www.skylinecomputer.com/Train_Welcome.asp, the www.skylinecomputer.com part is actually the name of the web server for the company that I work for. So, whether you type in the name of another networked computer or it is implied by what you see on your screen, the user typically identifies a remote computer by using a name.

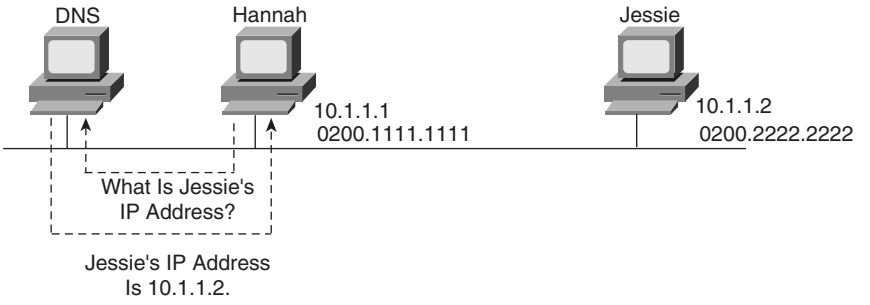
So, TCP/IP needs a way to let a computer find the IP address of another computer based on its name. TCP/IP also needs a way to find MAC addresses associated with other computers on the same LAN subnet. Figure 5-7 outlines the problem.

Figure 5-7 *Hannah Knows Jessie's Name, Needs IP Address and MAC Address*



Hannah knows her own name, IP address, and MAC address because those things are configured in advance. *What Hannah does not know are Jessie's IP and MAC addresses*. To find the two missing facts, Hannah uses the Domain Name System (DNS) and the Address Resolution Protocol (ARP). Hannah knows the IP address of a DNS server because the address was preconfigured on Hannah's machine. Hannah now sends a *DNS request* to the DNS, asking for Jessie's IP address. The DNS replies with the address, 10.1.1.2. Figure 5-8 shows the simple process.

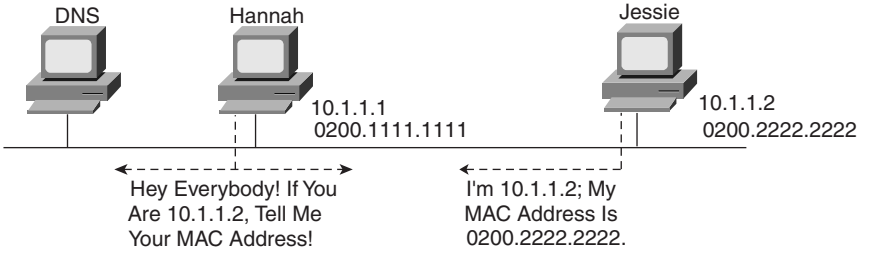
Figure 5-8 *DNS Request and Reply*



Hannah simply sends a DNS request to the server, supplying the name jessie, or jessie.skylinecomputer.com, and the DNS replies with the IP address (10.1.1.2, in this case). Effectively, the same thing happens when you surf the Internet and connect to any web site. Your PC somehow knows the IP address of the DNS; that information can be preconfigured or learned using Dynamic Host Configuration Protocol (DHCP), which is covered later in this chapter. Your PC sends a request, just like Hannah's request for Jessie, asking the DNS to resolve the name into an IP address. After that happens, your PC can start requesting that the web page be sent.

Back to the example with Hannah. Hannah still needs to know the Ethernet MAC address used by 10.1.1.2, so Hannah issues something called an *ARP broadcast*. An ARP broadcast is sent to a broadcast Ethernet address, so everyone on the LAN receives it. Because Jessie is on the LAN, Jessie receives the ARP broadcast. Because Jessie's IP address is 10.1.1.2 and the ARP broadcast is looking for the MAC address associated with 10.1.1.2, Jessie replies with her own MAC address. Figure 5-9 outlines the process.

Figure 5-9 *Sample ARP Process*

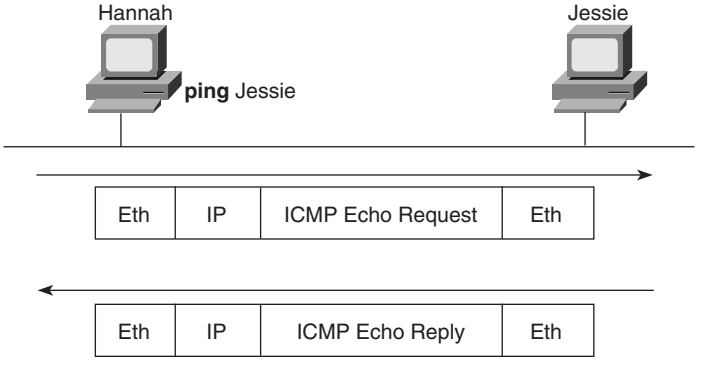


Now Hannah knows the destination IP and Ethernet addresses that she should use when sending frames to Jessie, and the packet in Figure 5-7 can be sent successfully.

ICMP Echo and the ping Command

IP needs to have a way to test basic IP connectivity, without relying on any applications to be working. Hannah, being a great network troubleshooter (in spite of being my 2-year-old daughter), can test basic network connectivity using the **ping** command. **ping** (Packet INternet Groper) uses the *Internet Control Message Protocol (ICMP)*, sending a message called an *ICMP echo request* to another IP address. The computer with that IP address should reply with an *ICMP echo reply*. If that works, you successfully have tested the IP network. ICMP does not rely on any application, so it really just tests basic IP connectivity—Layers 1, 2, and 3 of the OSI model. Figure 5-10 outlines the basic process.

Figure 5-10 *Sample Network,* **ping** *Command*



ICMP contains many features, which are discussed in detail in Chapter 13, "Basic Router Configuration and Operation."

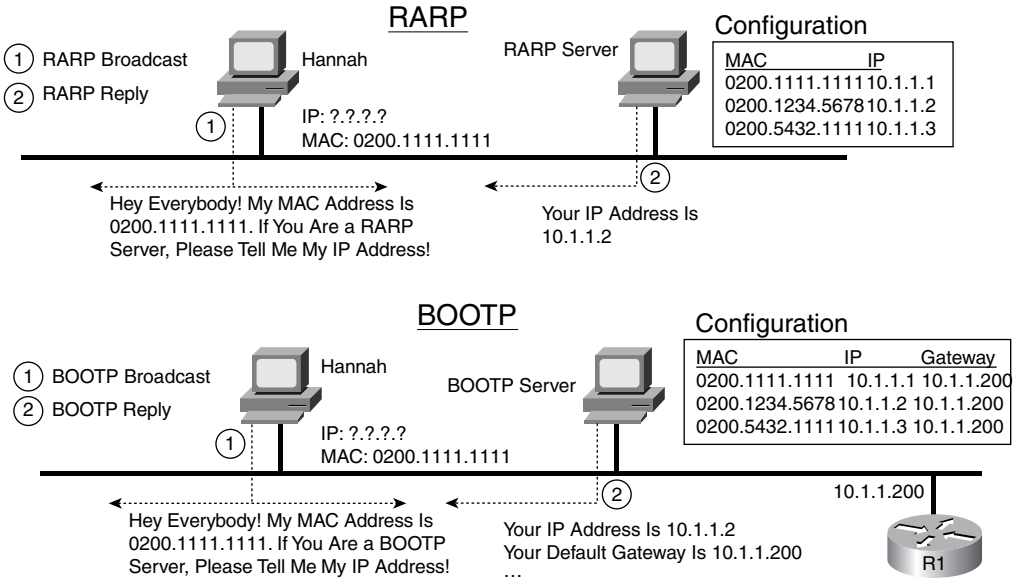
RARP, BOOTP, and DHCP

Over the years, three protocols have been popular to allow a host computer to discover the IP address it should use:

- Reverse ARP (RARP)
- Boot Protocol (BOOTP)
- Dynamic Host Configuration Protocol (DHCP)

RARP and BOOTP work using the same basic process. To use either protocol, a PC needs a LAN interface card. The computer sends a LAN broadcast frame announcing its own MAC address and requests that someone assign it an IP address. Figure 5-11 outlines the process for both RARP and BOOTP.

Figure 5-11 *RARP and BOOTP*



RARP and BOOTP requests sent to the LAN broadcast address simply ask for an IP address assignment. Both protocols allow for IP address assignment, but that is all that RARP can ask for—it can't even ask for the subnet mask used on the LAN. RARP is defined in RFC 903, whereas BOOTP was defined later in RFC 1542, including several improvements over RARP. So, BOOTP allows many more tidbits of information to be announced to a BOOTP client—its IP address, its subnet mask, its default gateway IP addresses, its other server IP addresses, and the name of a file that the computer should download.

Both RARP and BOOTP were created with the motivation to allow a diskless workstation to come up and start operating. With RARP, the creators of the protocol just wanted to get the machine an IP address so that a knowledgeable user could type in commands and copy the correct files from a server onto the diskless computer's RAM memory so that they could be used. The creators of BOOTP, anticipating a less sophisticated user in the future, wanted to automate as much of the process as possible—including the dynamic assignment of a default gateway (router) IP address.

BOOTP's name really comes from the feature in which BOOTP supplies the name of a file to the BOOTP client. Typically, the diskless workstations had enough permanent memory to boot a very simple operating system, with the expectation that the computer would use a simple protocol, such as the Trivial File Transfer Protocol (TFTP), to transfer a file containing a more sophisticated operating system into RAM. So, with the ultimate goal being to let a diskless computer complete the processing of initializing, or *booting*, a full operating system, BOOTP was aptly named.

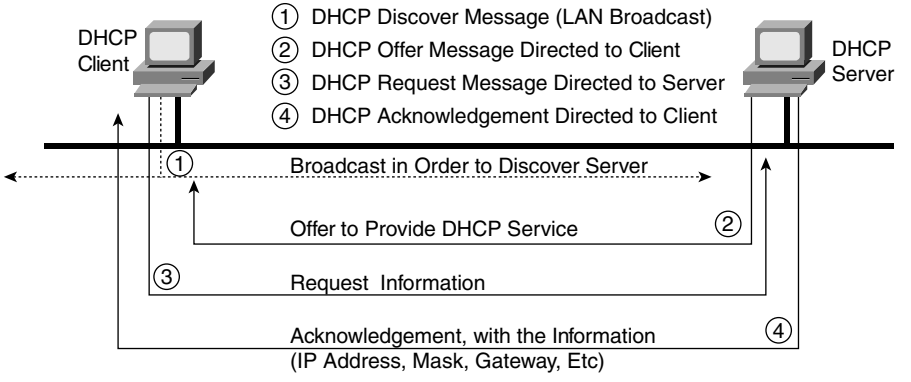
Neither RARP nor BOOTP is used much today. (They are possible topics for the INTRO exam, though.) One of the problems with both RARP and BOOTP is that they required a computer to act as a server, and the server was required to know the MAC address of every computer and the corresponding configuration parameters that each computer should be told. So, administration in a network of any size was painful.

DHCP, which is very popular in real networks today, solves some of the scaling and configuration issues with RARP and BOOTP, while supplying the same types of information. The main protocols for DHCP are defined in RFC 2131, but a couple of dozen additional RFCs define extensions and applications of DHCP for a variety of other useful purposes.

Like BOOTP, DHCP uses the concept of the client making a request and the server supplying the IP address to the client, plus other information such as the default gateway, subnet mask, DNS IP address, and other information. The biggest advantage of DHCP compared to BOOTP and RARP is that DHCP does not require that the DHCP server be configured with all MAC addresses of all clients. DHCP defines a process by which the server knows the IP subnet in which the DHCP client resides, and it can assign an IP address from a pool of valid IP addresses in that subnet. So, the DHCP server does not need to know the MAC address ahead of time. Also, most of the other information that DHCP might supply, such as the default router IP address, is the same for all hosts in the same subnet, so DHCP servers simply can configure information per subnet rather than per host and save a lot of administrative hassle compared to BOOTP.

The basic DHCP messages for acquiring an IP address are shown in Figure 5-12.

Figure 5-12 *DHCP Messages to Acquire an IP Address*

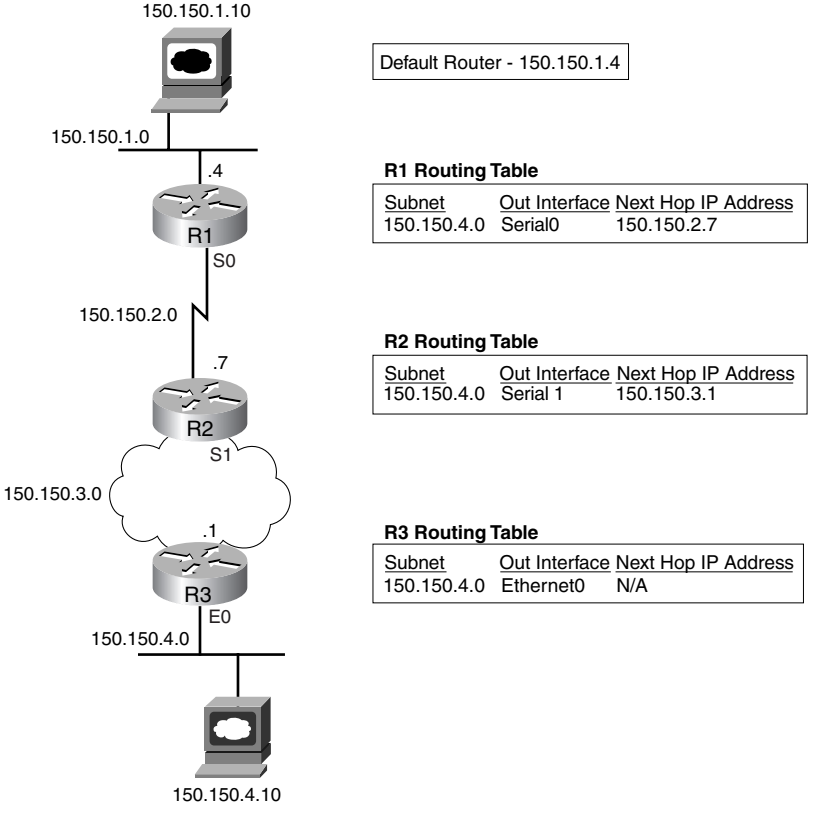


DHCP has become a very prolific protocol, with most end-user hosts on LANs in corporate networks getting their IP addresses and other basic configuration via DHCP.

IP Routing and Routing Protocols

In the first section of this chapter, you read about the basics of routing using a network with three routers and two PCs. Armed with more knowledge of IP addressing, you now can take a closer look at the process of routing IP. Figure 5-13 repeats the familiar network diagram, this time with subnets of network 150.150.0.0 used.

Figure 5-13 *Simple Routing Example, with IP Subnets*



First, a few detail about the figure need to be explained. The subnet numbers are shown, with the whole third octet used for the subnet part of the addresses. The actual IP addressed for PC1 and PC2 are shown. However, the full IP addresses of the routers are not shown in the figure. Many times, to reduce clutter, only the host part of the address is listed in a figure. For instance, R2's IP address on the serial link to R1 is 150.150.2.7. The subnet is 150.150.2.0, and the .7 shown beside R2 in the figure represents the host part of the address, which is the fourth octet in this case.

A detailed examination of the routing logic used by PC1, R1, R2, and R3 is listed earlier in this chapter. That same logic is repeated here, using the more detailed information contained in the figure:

Step 1 **PC1 sends the packet to R1—**PC1 first builds the IP packet, with a destination address of PC2's IP address (150.150.4.10). PC1 needs to send the packet to R1 because it knows that its default router is 150.150.1.4. PC1 first checks its ARP cache, hoping to find R1's Ethernet MAC address. If it is not found, PC1 ARPs to learn R1's Ethernet MAC address. Then PC1 places the IP packet into an Ethernet frame, with a destination Ethernet address of R1's Ethernet address. PC1 sends the frame onto the Ethernet.

Step 2 **R1 processes the incoming frame and forwards the packet to R2—**Because the incoming Ethernet frame has a destination MAC of R1's Ethernet MAC, R1 copies the frame off the Ethernet for processing. If the FCS passes, meaning that the Ethernet frame did not have any errors in it, R1 looks at the Protocol Type field to discover that the packet inside the frame is an IP packet. R1 then discards the Ethernet header and trailer.

Next, R1 looks for the routing table entry that matches the destination address in the packet, 150.150.4.10. The routing table entry is listed in the figure—a route to subnet 150.150.4.0, with outgoing interface Serial0 to next-hop router R2 (150.150.2.7).

Now R1 just needs to build an HDLC frame and send it out its Serial0 interface to R2. As mentioned earlier, ARP is not needed on a point-to-point HDLC WAN link. R1 knows all the information necessary to out the packet inside an HDLC frame and send the frame.

Step 3 **R2 processes the incoming frame and forwards the packet to R3—**R2 repeats the same general process as R1 when it receives the HDLC frame. After stripping the HDLC header and trailer, R2 also needs to find the routing table entry that matches destination 150.150.4.10. R2's routing table has an entry for 150.150.4.0, outgoing interface serial1, to next-hop router 150.150.3.1, which is R3.

Before R2 can complete the task, the correct DLCI for the VC to R3 must be decided. The details of how R2 knows the right DLCI are covered in Chapter 11, "Frame Relay," of the *CCNA ICND Exam Certification Guide*. With that mapping information, R2 can complete the Frame Relay header and send the frame to R3.

Step 4 **R3 processes the incoming frame and forwards the packet to PC2—** Like R1 and R2 before it, R3 checks the FCS in the data-link trailer, looks at the type field to decide whether the packet inside the frame is an IP packet, and then discards the Frame Relay header and trailer. The routing table entry for 150.150.4.0 shows that the outgoing interface is R3's Ethernet interface, but there is no next-hop router because R3 is connected directly to subnet 150.150.4.0. All R3 has to do is encapsulate the packet inside a Ethernet header and trailer, and forward the frame. Before R3 can finish building the Ethernet header, an IP ARP broadcast must be used to find PC2's MAC address (assuming that R3 doesn't already have that information in its IP ARP cache).

The routing process relies on the rules relating to IP addressing. For instance, why did 150.150.1.10 (PC1) assume that 150.150.4.10 (PC2) was not on the same Ethernet? Well, because 150.150.4.0, PC2's subnet, is different than 150.150.1.0, which is PC1's subnet. Because IP addresses in different subnets must be separated by some router, PC1 needed to send the packet to some router—and it did. Similarly, all three routers list a route to subnet 150.150.4.0, which, in this example, includes IP addresses 150.150.4.1 to 150.150.4.254. What if someone tried to put PC2 somewhere else in the network, but still using 150.150.4.10? The routers then would forward packets to the wrong place. So, Layer 3 routing relies on the structure of Layer 3 addressing to route more efficiently.

IP Routing Protocols

IP routing protocols fill the IP routing table with valid, (hopefully) loop-free routes. Each route includes a subnet number, the interface out which to forward packets so that they are delivered to that subnet, and the IP address of the next router that should receive packets destined for that subnet (if needed).

Before examining the underlying logic, you need to consider the goals of a routing protocol. The goals described in the following list are common for any IP routing protocol, regardless of its underlying logic type:

- To dynamically learn and fill the routing table with a route to all subnets in the network.
- If more than one route to a subnet is available, to place the best route in the routing table.
- To notice when routes in the table are no longer valid, and to remove those routes from the routing table.
- If a route is removed from the routing table and another route through another neighboring router is available, to add the route to the routing table. (Many people view this goal and the preceding one as a single goal.)

- To add new routes, or to replace lost routes, with the best currently available route as quickly as possible. The time between losing the route and finding a working replacement route is called *convergence* time.
- To prevent routing loops.

Routing protocols can become rather complicated, but the basic logic that they use is relatively simple. Routing protocols take the routes in a routing table and send a message to their neighbors telling them about the routes. After a while, everyone has heard about all the routes.

Figure 5-14 shows a sample network, with routing updates shown. Table 5-6 lists Router B's routing table before receiving the routing updates, and Table 5-7 lists Router B's routing table after receiving the routing updates.

Figure 5-14 *Router A Advertising Routes Learned from Router C*

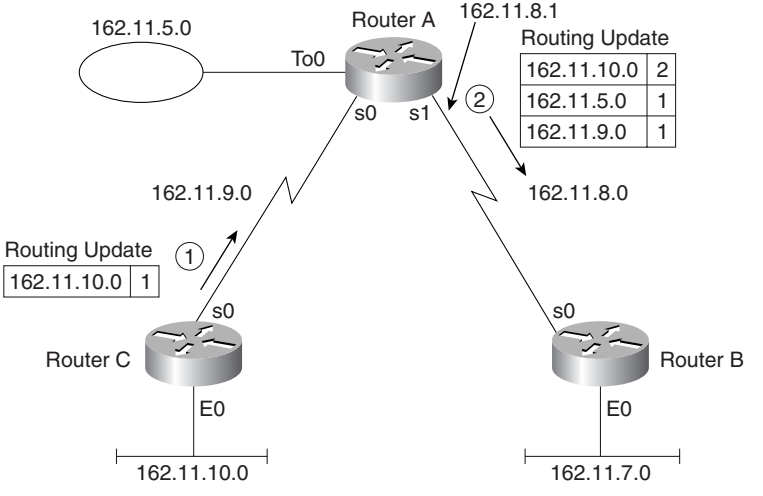


Table 5-6 *Router B Routing Table Before Receiving the Update Shown in Figure 5-14*

Group	Outgoing Interface	Next-Hop Router	Metric	Comments
162.11.7.0	E0	—	0	This is a directly connected route.
162.11.8.0	S0	—	0	This is a directly connected route.

Table 5-7 *Router B Routing Table After Receiving the Update Shown in Figure 5-14*

Group	Outgoing Interface	Next-Hop Router	Metric	Comments
162.11.5.0	S0	162.11.8.1	1	Learned from Router A, so next-hop is Router A.
162.11.7.0	E0	—	0	This is a directly connected route.
162.11.8.0	S0	—	0	This is a directly connected route.
162.11.9.0	S0	162.11.8.1	1	Learned from Router A, so next-hop is Router A.
162.11.10.0	S0	162.11.8.1	2	This one was learned from Router A, which learned it from Router C.

Router B adds routes for directly connected subnets when the interfaces first initialize. In fact, no routing protocols are needed for a router to learn routes to the directly connected subnets. So, before Router B receives any routing updates, it knows about only two routes—the two connected routes—as listed in Table 5-6.

After receiving the update from Router A, Router B has learned three more routes. Because Router B learned those routes from Router A, all three of B's routes point back to Router A as the next hop router. That makes sense because it is obvious from the figure that B's only path to the other subnets lies through Router A.

Router A learned about subnets 162.11.5.0 and 162.11.9.0 because A is connected directly to those subnets. Router A, in turn, learned about subnet 162.11.10.0, the subnet off Router C's Ethernet, from routing updates sent by Router C.

Foundation Summary

The "Foundation Summary" section of each chapter lists the most important facts from the chapter. Although this section does not list every fact from the chapter that will be on your CCNA exam, a well-prepared CCNA candidate should know, at a minimum, all the details in each "Foundation Summary" section before going to take the exam.

The routing process forwards the packet, and only the packet, from end to end through the network, discarding data-link headers and trailers along the way. The network layer processes deliver the packet end to end, using successive data-link headers and trailers just to get the packet to the next router or host in the path. Figure 5-15 shows the concepts behind encapsulation used by routers.

Figure 5-15 *Network Layer and Data Link Layer Encapsulation*

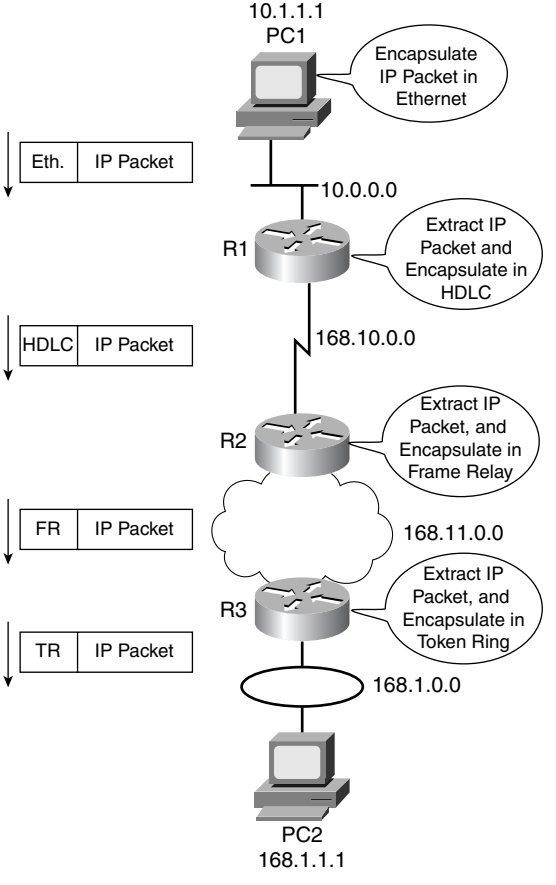


Table 5-8 outlines several Layer 3 address structures.

Table 5-8 *Layer 3 Address Structures*

Protocol	Size of Address in Bits	Name and Size of Grouping Field in Bits	Name and Size of Local Address Field in Bits
IP	32	Network or subnet (variable, between 8 and 30 bits)	Host (variable, between 2 and 24 bits)
IPX	80	Network (32)	Node (48)
AppleTalk	24	Network* (16)	Node (8)
OSI	Variable	Many formats, many sizes	Domain-specific part (DSP—typically 56, including NSAP)

*Consecutively numbered values in this field can be combined into one group, called a cable range.

The general ideas about how IP address groupings can be summarized as follows:

- All IP addresses in the same group must not be separated by a router.
- IP addresses separated by a router must be in different groups.

Table 5-9 summarizes the characteristics of Class A, B, and C networks.

Table 5-9 *Sizes of Network and Host Parts of IP Addresses with No Subnetting*

Any Network of This Class	Number of Network Bytes (Bits)	Number of Host Bytes (Bits)	Number of Addresses per Network*
A	1 (8)	3 (24)	$2^{24} - 2$
B	2 (16)	2 (16)	$2^{16} - 2$
C	3 (24)	1 (8)	$2^{8} - 2$

*There are two reserved host addresses per network.

Network numbers look like actual addresses because they are in dotted-decimal format. However, network numbers are not actually IP addresses because they cannot be assigned to an interface as an IP address.

Table 5-10 summarizes the possible network numbers, the total number of each type, and the number of hosts in each Class A, B, and C network.

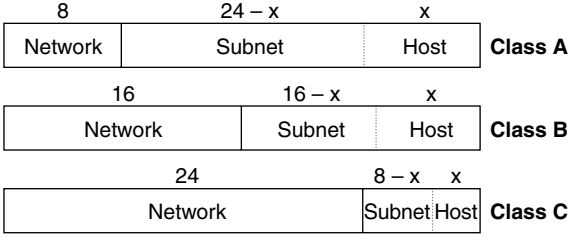
Table 5-10 *List of All Possible Valid Network Numbers**

Class	First Octet Range	Valid Network Numbers*	Total Number of This Class of Network	Number of Hosts per Network
A	1 to 126	1.0.0.0 to 126.0.0.0	$2^7 - 2$	$2^{24} - 2$
B	128 to 191	128.1.0.0 to 191.254.0.0	$2^{14} - 2$	$2^{16} - 2$
C	192 to 223	192.0.1.0 to 223.255.254.0	$2^{21} - 2$	$2^8 - 2$

*The Valid Network Numbers column shows actual network numbers. There are several reserved cases. For example, networks 0.0.0.0 (originally defined for use as a broadcast address) and 127.0.0.0 (still available for use as the loopback address) are reserved. Networks 128.0.0.0, 191.255.0.0, 192.0.0.0, and 223.255.255.0 also are reserved.

When subnetting, the host part of the address shrinks to make room for the subnet part of the address. Figure 5-16 shows the format of addresses when subnetting.

Figure 5-16 *Address Formats When Subnetting Is Used*



The goals described in the following list are common for any IP routing protocol, regardless of its underlying logic type:

- To dynamically learn and fill the routing table with a route to all subnets in the network.
- If more than one route to a subnet is available, to place the best route in the routing table.
- To notice when routes in the table are no longer valid, and to remove those routes from the routing table.
- If a route is removed from the routing table and another route through another neighboring router is available, to add the route to the routing table. (Many people view this goal and the preceding one as a single goal.)

- To add new routes, or to replace lost routes with the best currently available route, as quickly as possible. The time between losing the route and finding a working replacement route is called *convergence* time.
- To prevent routing loops.

Q&A

As mentioned in the introduction, you have two choices for review questions. The questions that follow give you a bigger challenge than the exam itself by using an open-ended question format. By reviewing now with this more difficult question format, you can exercise your memory better and prove your conceptual and factual knowledge of this chapter. The answers to these questions are found in Appendix A.

For more practice with exam-like question formats, including questions using a router simulator and multiple-choice questions, use the exam engine on the CD.

1. What are the two main functions of each OSI Layer 3–equivalent protocol?
2. Assume that PC1 sends data to PC2, and PC2 is separated from PC1 by at least one router. Are the IP addresses of the PCs in the same IP subnet? Explain your answer.
3. Assume that PC1 sends data to PC2, and PC2 is not separated from PC1 by at least one router. Are the IP, addresses of the PCs in the same IP subnet? Explain your answer.
4. How many bits are present in an IP address?
5. How many bits are present in an IPX address?
6. How many bits are present in an AppleTalk address?
7. Name the two main parts of an IPX address. Which part identifies which group this address is a member of?
8. Name the two main parts of an IP address. Which part identifies which group this address is a member of?
9. PC1 sends data to PC2 using TCP/IP. Three routers separate PC1 and PC2. Explain why the statement “PC1 sends an Ethernet frame to PC2” is true or false.
10. In IP addressing, how many octets are in 1 byte?
11. Describe the differences between a routed protocol and a routing protocol.
12. Name at least three routed protocols.
13. Name at least three IP routing protocols.
14. Imagine an IP host on an Ethernet, with a single router attached to the same segment. In which cases does an IP host choose to send a packet to this router instead of directly to the destination host, and how does this IP host know about that single router?
15. Name three items in an entry in any routing table.

16. Name the parts of an IP address when subnetting is used.
17. How many valid IP addresses exist in a Class A network? (You may refer to the formula if you do not know the exact number.)
18. How many valid IP addresses exist in a Class B network? (You may refer to the formula if you do not know the exact number.)
19. How many valid IP addresses exist in a Class C network? (You may refer to the formula if you do not know the exact number.)
20. What values can a Class A network have in the first octet?
21. What values can a Class B network have in the first octet?
22. What values can a Class C network have in the first octet?
23. When subnetting a Class B network, do you create the subnet field by taking bits from the network part of the address or the host part?
24. When subnetting a Class B network, using the entire third octet for the subnet part, describe the number of possible subnets created.
25. When subnetting a Class A network using the entire second octet for the subnet part, describe the number of hosts in each subnet.
26. When a router hears about multiple routes to the same subnet, how does it choose which route to use?
27. What is the primary purpose of a routing protocol?
28. True or false: "Routing protocols are required to learn routes of directly connected subnets."
29. Which IP routing protocols are Cisco proprietary?
30. List the similarities and differences between RARP and BOOTP.
31. List the similarities and differences between DHCP and BOOTP.
32. List the similarities and differences between ARP and DNS.