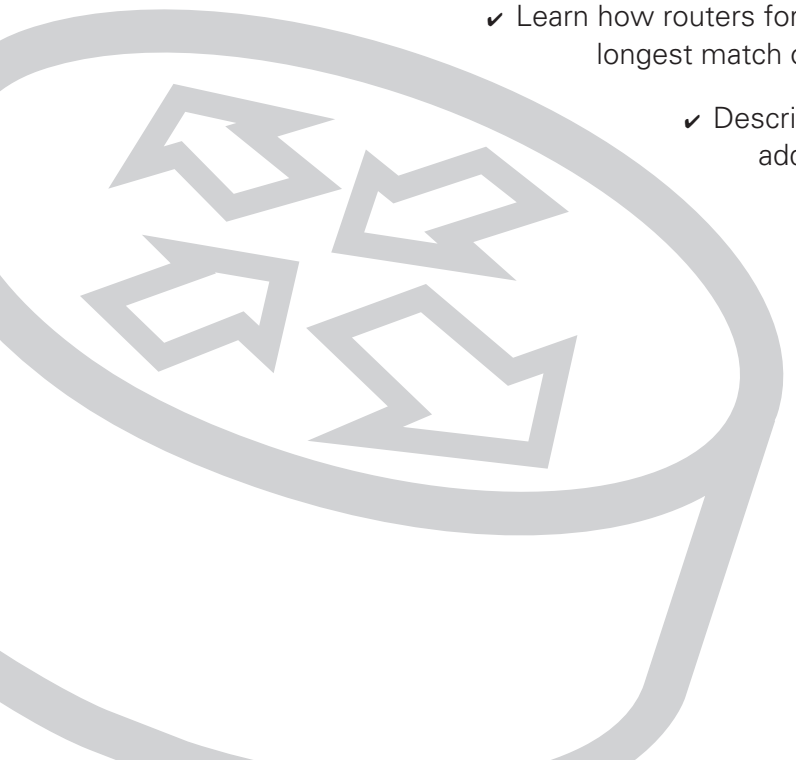
What You Will Learn

After reading this chapter, you should be able to

- ✔ Understand the IP version 4 (IPv4) addressing protocol, and the similarities to the addresses used in the postal delivery system
- ✔ Explain why hosts need two addresses, Ethernet and Internet, and how the Address Resolution Protocol (ARP) is used to determine the Ethernet address of a host given the host's IP address
- ✔ Understand the TCP and UDP protocols, and how they are used to deliver data to a destination host
- ✔ Explain the TCP/IP layered model by comparing it to the layer model that was developed for the postal delivery system
- ✔ Differentiate between classful and classless IP addresses
- ✔ Learn how to subnet and summarize IP networks
- ✔ Learn how routers forward packets using the longest match operation
- ✔ Describe the IP version 6 (IPv6) addressing protocol



CHAPTER 3

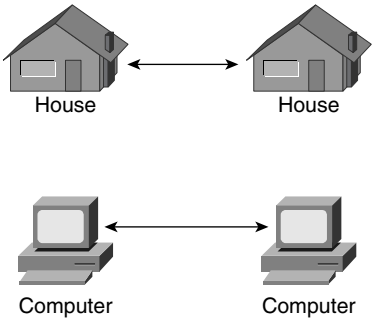
Internet Addressing and Routing

In Chapter 1, you examined systems for delivering the mail, planning a road trip, and making telephone calls. Chapter 2 introduced the binary, octal, and hexadecimal numbering systems. You need to understand how computers represent information, and how you can move between number systems to represent binary numbers in a more readable form. In this chapter, the concepts from the first two chapters will be combined to understand the schemes that are necessary to create a scalable computer communication system — the Internet.

Internet Addressing

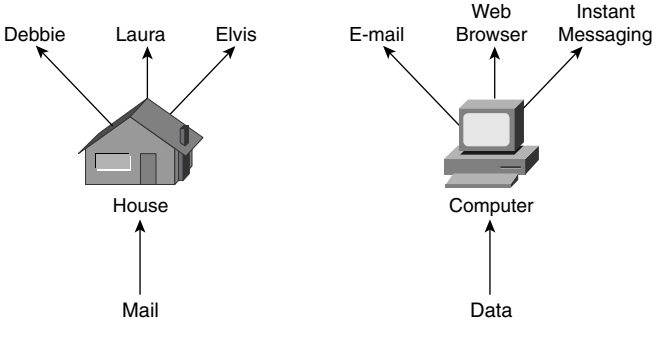
To begin our discussion on computer communication over a network, this section looks at the similarities between mail delivery between houses, and data delivery between computers. The endpoints in mail delivery are houses, and the endpoints between electronic data delivery are computers. Certainly there can be other endpoints in both systems. Letters can be delivered from a house to a business, from a business to a house, between two businesses, and so on. Electronic data delivery can be from a news service to your cell phone or personal data assistant (PDA), from your computer to your friend's pager, from environmental sensors in a building to the heating and cooling control systems for that building, and so on. But to keep the discussion simple, it will suffice to concentrate on mail delivery between houses, and electronic data delivery between computers. The first analogy is that an endpoint in a mail delivery system, a house, is equivalent to the endpoint in a computer communication system, a PC. (See Figure 3-1.)

Figure 3-1 Equivalent Endpoints in the Mail and Data Communication Systems



In the mail delivery system, the function of the post office is to deliver mail to a particular house. In the computer communication system, the function of the Internet is to deliver data to a particular PC. Yet, in both systems, the endpoint is not the ultimate destination. For mail, the ultimate recipient is a person. For data, the ultimate recipient is an application such as an e-mail program, a web browser, an audio or video program, an instant messaging program, or any number of wonderful applications that exist today. (See Figure 3-2.)

Figure 3-2 Final Destinations in the Postal and Electronic Data Delivery Systems



Although the ultimate recipient is a person or a software application, the responsibility of the systems stops when the mail, or data, is delivered to the proper house, or computer. However, as part of the address, you still need the ultimate recipient; either a person or an application, even though this information is not used for delivery to an endpoint. The endpoint uses the name or application to enable delivery to the recipient.

Because the two systems are analogous, it is instructive to revisit the format of an address in the mail delivery system and see if you can use a similar format for electronic data delivery:

Name

Street Number, Street Name

City, State

Although there are five distinct pieces of information in the mail address (name, street number, street name, city, and state), you can consider an address to contain only four pieces of information. For endpoint delivery, you can ignore the name field. You are left with

Street Number

Street Name

City

State

The postal system routers (core, distribution, and access) use the state, city, and street names to deliver the mail from the source access post office to the destination access post office. The street number is not needed until the mail arrives at the access post office that is directly connected to the destination street. So, the address can be broken down into

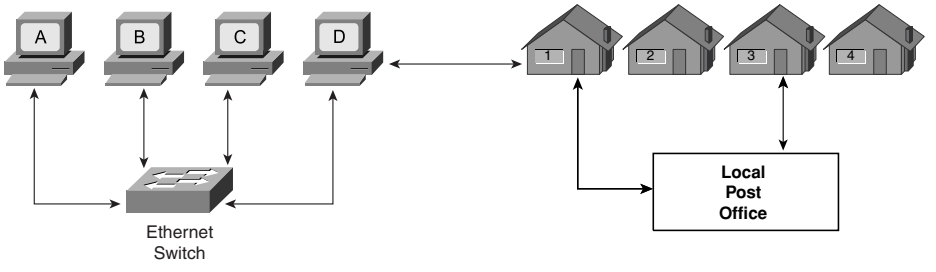
State, City, Street Name

and

Street Number

The state, city, and street name information enables the mail to get close to the destination (a particular street). The street number is used to deliver the mail to the proper house. What is the analogy in the computer world to houses on a street? Recall from Chapter 1 that a group of computers can directly communicate with each other through a switch residing on a local-area network (LAN). So a LAN is the computer equivalent to a street. (See Figure 3-3.)

Figure 3-3 LAN of Computers Is Similar to a Street of Houses



Chapter 1 also mentioned that computers have an address, and the most common technology used for computer communication is Ethernet. The sample Ethernet address that was presented in Chapter 1 was 00-03-47-92-9C-6F.

Before you learn more about Ethernet addresses, take the following quiz to make sure you understand the concepts described so far:

1. What number base is used to represent the Ethernet address?

   Answer: Hexadecimal, because the symbols C and F are not used in the other number bases that we discussed. Computers compute using binary. The hexadecimal representation is for our benefit because it is easier to read and write.

2. How many bytes are in an Ethernet address?

   Answer: Six. One hexadecimal digit contains 4 bits, or 1/2 bytes. Two hexadecimal digits contain 8 bits, or 1 byte. An Ethernet address contains 12 hexadecimal digits or 6 bytes.

3. How many bits are in an Ethernet Address?

   Answer: 48 (8 bits per byte).

4. How many Ethernet addresses are possible?

   Answer: 2^{48} or 281 trillion, 474 billion, 976 million, 710 thousand, 6 hundred fifty-six (281,474,976,710,656).

An Ethernet address is not a property of your PC. An Ethernet address is a property of the Ethernet card, or built in Ethernet port in your PC. If you put a new Ethernet card in your PC, the Ethernet address of your PC changes.

By itself, an Ethernet address cannot deliver data between two endpoints on the Internet. The reason is that there is no structure to an Ethernet address. There are many manufacturers of Ethernet cards for computers, and each manufacturer is assigned a block of Ethernet address to use for their particular brand of card.

An analogy would be to have 281,474,976,710,656 postal addresses that are sold in a local postal address store. Each local postal address store is given a block of numbers from the total range of numbers that are possible. A postal address is just a number between 0 and 281,474,976,710,655. When you build a house, you would go to the local postal address store and your house would be assigned one of the numbers that hasn't yet been assigned. Everyone in your city would need to get a number assigned from the local postal address store. Because people will not be going to the store in any order, numbers will be assigned randomly throughout the city. The only way that these numbers can be used to deliver mail is if every post office at every level (core, distribution, and access) maintained a list of every number, and the route to reach that number. Therefore, every post office would need to maintain a list of 281,474,976,710,656 addresses and the route to get there. Obviously, this is not scalable. So in addition to an Ethernet address, you need another address that has a structure analogous to the structure of the postal address. What you need is an Internet addressing protocol.

Internet Addressing Protocol

The Internet addressing protocol, or Internet Protocol (IP), is an additional addressing scheme. By following this protocol you can have an address with a structure that will allow you to build a scalable data communication system analogous to the postal system. Because this discussion is about computers, you need to decide how many bits you need for an Internet address, and how many Internet addresses you need. Computers can easily work with byte size pieces of data, so the number of bits in the IP should be a multiple of 8 bits or 1 byte. How many IP addresses will you need? That is, and was, a difficult question. When the IP was developed more than 20 years ago, the PC was not common, and it was difficult to imagine the explosion that would take place in the number of computers used throughout the world.

If you placed yourself back in 1980, and had to determine the size, in bits, of an IP address, what would you have picked? You might have started by determining how many addresses are possible based on the number of bytes that are used. And you might have created a table similar to Table 3-1.

Table 3-1 Number of IP Addresses Versus Number of Bytes

Number of Bytes	Number of Bits	Number of Addresses
1	8	$2^8 = 256$
2	16	$2^{16} = 65,536$
3	24	$2^{24} = 16,777,216$
4	32	$2^{32} = 4,294,967,296$
5	40	$2^{40} = 1,099,511,627,776$
6	48	$2^{48} = 281,474,976,710,656$

In 1980, there were more than 256 computers in use, so 1 byte would not be sufficient. Two bytes would give us 65,280 more addresses, but the number is still not sufficient. Although 3 bytes allow more than 16 million addresses, you know that

computers are happier with even numbers of things than odd numbers of things. An ideal size is 4 bytes. Four is an even number and you can have in excess of 4 billion IP addresses, which should be sufficient. Now that you have settled on using a 32-bit address for the Internet address, you next need to determine a structure for those 32 bits.

The postal addressing and delivery schemes worked quite well for mail delivery, so let's try and impose the same type of structure on the Internet addresses. You know that a postal address has two components. The first component consisted of the state, city, and street names. The second component was the street or house number. Although the entire address was needed to identify a particular endpoint, or house, you did not need the street number for delivery until the mail reached the street containing the house. Using the same philosophy for the Internet address, use part of the 32 bits to designate the LAN or local network where the computer resides (the ***network address***); and the remaining bits in the address to identify a particular computer, or host (the ***host address***), on that LAN.

The next step is to determine how many bits to use for the network address and how many bits to use to identify a computer, or host, on that network. The easiest approach is to work with bytes, and then use the dotted decimal notation to represent network and host addresses using decimal numbers. You can't use all the bytes for the network address, and you can't use all the bytes for the computer address so the possibilities that remain are listed in Table 3-2.

Table 3-2 Internet Address Structures

Network Address Size	Number of Possible Networks	Host Address Size	Number of Possible Hosts
1 Byte	$2^8 = 256$	3 Bytes	$2^{24} = 16{,}777{,}216$
2 Bytes	$2^{16} = 65{,}536$	2 Bytes	$2^{16} = 65{,}536$
3 Bytes	$2^{24} = 16{,}777{,}216$	1 Byte	$2^8 = 256$

After some thought, you decide that you want to make some modifications to the range of host addresses. It would be nice if you had a broadcast capability where a message could be sent to every host on a network. Therefore, you need a ***broadcast address*** for the network, and you want the broadcast address to be easy to remember. This can be achieved by using a host address of all 1s for the broadcast address. In addition, you want an address that points to the network itself, or "this" network. This can be achieved by using a host address of all 0s. To accommodate the new broadcast and "this" network addresses, the number of hosts listed in Table 3-2 must now be reduced by two. (See Table 3-3.)

Table 3-3 Number of Hosts Possible

Network Address Size	Number of Possible Networks	Host Address Size	Number of Possible Hosts
1 Byte	$2^8 = 256$	3 Bytes	$2^{24} - 2 = 16{,}777{,}214$
2 Bytes	$2^{16} = 65{,}536$	2 Bytes	$2^{16} - 2 = 65{,}534$
3 Bytes	$2^{24} = 16{,}777{,}216$	1 Byte	$2^8 - 2 = 254$

Try working through the next exercises to reinforce your understanding.

For the first type or class of networks that use 1 byte for the network address and 3 bytes for the host address, the range of addresses in dotted decimal notation are:

0.0.0.1–0.255.255.254 for network 0

1.0.0.1–1.255.255.254 for network 1

2.0.0.1–2.255.255.254 for network 2

...

255.0.0.1–255.255.255.254 for network 255

1. What is the address for host 8 on network 129?

   Answer: 129.0.0.8

2. What is the broadcast address for network 129?

   Answer: 129.255.255.255

For the second class of networks that use 2 bytes for both the network and host addresses, the range of addresses in dotted decimal notation are

0.0.0.1–0.0.255.254 for network 0

0.1.0.1–0.1.255.254 for network 1

...

255.254.0.1–255.254.255.254 for network 65,534

255.255.0.1–255.255.255.254 for network 65,535

3. What is the address for host 8 on the 258th network?

   Answer: Answer.1.1.0.8. The 258th network is network number 257 because network numbering started at 0; 257 in binary = 0000 0001 0000 0001 = 256 + 1 = 257 = 1.1 in dotted decimal notation.

4. What is the broadcast address for the 258th network?

   Answer: 1.1.255.255

   For the third class of networks that use 1 byte for the host address and 3 bytes for the network address, the range of addresses in dotted decimal notation are

   0.0.0.1–0.0.0.254 for network 0

   0.0.1.1–0.0.1.254 for network 1

   ...

   0.1.0.1–0.1.0.254 for network 256

   ...

   1.0.0.1–1.0.0.254 for network 65536

   ...

   255.255.255.1–255.255.255.254 for network 16,777,215

5. What is the address for host 8 on network 25?

   Answer: 0.0.25.8

6. What is the address for host 12 on network 103,481?

   Answer: 103,481 in hexadecimal = $019439.01_{16} = 1_{10}.94_{16} = 148_{10}.39_{16} = 57_{10}$. 103,481 in dotted decimal = 1.148.57 so the address for host 12 on network 103,481 is 1.148.57.12

7. What is the broadcast address for network 103,481?

   Answer: 1.148.57.255

Classful IP Addresses

You must decide which scheme you are going to use for the Internet addresses. You could pick just one, but why not use all three? That way, you would have the flexibility of having networks with a few hosts (254), networks with a moderate number of hosts (65534), and networks with many hosts (16,777,214). This does sound like a good idea, but you need to be able to mix the three address types. A simple way is to use part of the first byte to signal the type of address used. The first byte, like all bytes, contains 8 bits. Using the first few bits to identify the type of network gives you the following rules:

- If the first, or most significant, bit of the first byte is 0, then 1 byte is used for the network address and 3 bytes are used for the host, broadcast, and "this" network address.
- If the most significant bit of the first byte is 1 and the next bit is 0, 2 bytes are used for the network address and 2 bytes are used for the host, broadcast, and "this" network address.
- If the first 2 bits of the first byte are 1, then 3 bytes are used for the network address and 1 byte is used for the host, broadcast, and "this" network address.

Three types of addresses are called Class A, B, and C. In a Class A address, the most significant bit of the first byte is 0. Additionally, you want to reserve two

addresses from the Class A address space. Address 0.0.0.X (where X can be any value from 0–255) is reserved. Address 127.X.X.X is reserved for what is called a *loopback address*. The loopback address is used by a host to send a message to itself without even being connected to a network. This can be used for testing applications without interfering with the network. So the range of values for the first byte in a Class A address is 0000–0001–0111 1110 or 1–126 (0 and 127 are reserved).

The entire range of Class A addresses is – 126.255.255.255.

In a Class B address, the most significant bit of the first byte is 1 and the next bit is 0. So the range of values for the first byte in a Class B address is 1000 0000–1011 1111 or 128–191.

The entire range of Class B addresses is: 128.0.0.0–191.255.255.255.

In a Class C address, the first two most significant bits of the first byte are 1 and the next bit is 0. So the range of values for the first byte in a Class C address is 1100 0000–1101 1111 or 192–223.

The entire range of Class C addresses is: 192.0.0.0–223.255.255.255.

This Internet Protocol addressing scheme is called *classful* because every address falls into one of three classes of addresses as summarized in Table 3-4.

Table 3-4 Classful Internet Protocol Address Ranges

Address Class	First Host Address	Last Host Address
A	1.0.0.1	126.255.255.254
B	128.0.0.1	191.255.255.254
C	192.0.0.1	223.255.255.254

A Class A address is identified by the first bit being a 0, as follows:

0XXX XXXX

A Class B address is identified by the first 2 bits being 1 0, as follows:

10XX XXXX

And a Class C address has the first 3 bits set to 1 1 0, as follows:

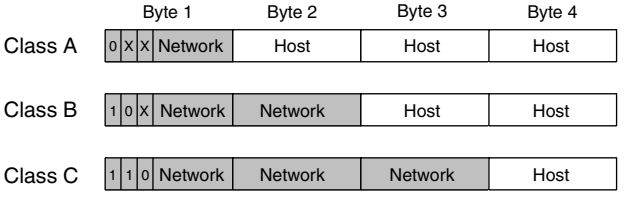
110X XXXX

One question might have popped into your mind at this point: What about addresses that are not Class A, B, or C? In other words, what about addresses where the first four bits are 1 1 1 0 or 1 1 1 1? Addresses beginning with 1110 are a different class of addresses, Class D, which you will learn about in Chapter 9, "Multicast—What the Post Office Can't Do." The Class D, or multicast, address space is in the range **1110** 0000–1110 1111 or 224–239.

Addresses beginning with 1111 are reserved for future use and cover the remaining address space starting at 240.0.0.0.

Figure 3-4 summarizes the structure of the classful IP addresses scheme.

Figure 3-4 Classful IP Addressing Structures



Private IP Addresses

The final addition to the Internet addressing protocol is that of private IP addresses. A public IP address is one that is reachable on the Internet, and therefore must be globally unique (two computers cannot use the same public IP address). You can have LANs that are not connected to the Internet, but the computers on these LANs are using IP for communication. It doesn't make sense to waste public IP addresses on these computers, so a range of addresses has been set

aside for these private networks to use. (See Table 3-5.) Because they are private, the same addresses can be used on more than one LAN with the realization that communication between LANs using the same private IP addresses is not possible.

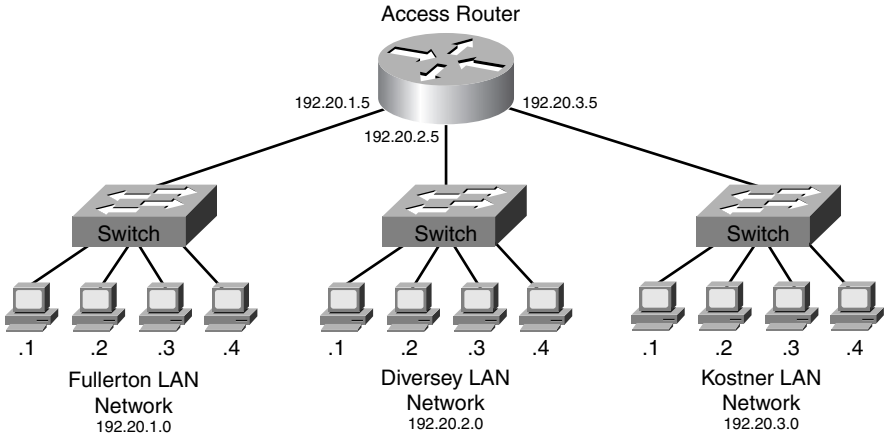
Table 3-5 Private IP Address Ranges

Class	Range
A	10.0.0.0– 10.255.255.255
B	172.16.0.0– 172.31.255.255
B (Used by Microsoft)	169.254.0.0– 169.254.255.255
C	192.168.0.0– 192.168.255.255

Address Resolution

An IP address is routable. Routers can use the network portion of an IP address to make a delivery, or routing decision, to the destination network. Ethernet addresses are not routable (unless every router knows how to reach every Ethernet address). Ultimately, electronic data must be delivered to a host using the host's Ethernet address. To do this, you need a protocol to determine, or resolve, the Ethernet address associated with a host's IP address. There is an analogy for address resolution that you are familiar with. Assume that you want to call your friend Steve and you do not know his telephone number, but you know where he lives. What do you use to resolve Steve's telephone number from his address? A telephone book. With a computer network, you need to do essentially the same thing when resolving between Ethernet and IP addresses. In Figure 3-5, there are three LANs with four hosts each.

Figure 3-5 Address Resolution Matches a Host's Ethernet Address with a Host's IP Address



The Fullerton, Diversey, and Kostner LANs have been assigned networks 192.20.1.0, 192.20.2.0, and 192.20.3.0 (remember the 0 designates "this" network). The host addresses on these LANs are .1, .2, .3, and .4. This is a shorthand notation for IP addresses 192.20.1.1, 192.20.1.2, 192.20.1.3, and 192.20.1.4 on the Fullerton LAN—and the same shorthand notation is used on the other two LANs. Also notice that the three Ethernet interfaces on the access router have also been assigned an IP address taken from the range of addresses associated with each LAN.

This section uses the networks in Figures 3-5 and 3-6 to trace through the steps a host uses to send data to a host on the same LAN and to a host on a different LAN. Tables 3-6, 3-7, and 3-8 contain the IP and Ethernet addresses for the hosts and router on the three LANs.

Table 3-6 Fullerton LAN Address Associations

IP Address	Ethernet Address
192.20.1.1 (Host 1)	00-03-47-92-9C-6F
192.20.1.2 (Host 2)	00-03-47-92-9C-70

Table 3-6 Fullerton LAN Address Associations (continued)

IP Address	Ethernet Address
192.20.1.3 (Host 3)	00-03-47-92-9C-71
192.20.1.4 (Host 4)	00-03-47-92-9C-72
192.20.1.5 (Router)	00-03-47-92-9C-73

Table 3-7 Diversey LAN Address Associations

IP Address	Ethernet Address
192.20.2.1 (Host 1)	00-03-48-AB-61-01
192.20.2.2 (Host 2)	00-03-48-AB-61-02
192.20.2.3 (Host 3)	00-03-48-AB-61-03
192.20.2.4 (Host 4)	00-03-48-AB-61-04
192.20.2.5 (Router)	00-03-48-AB-61-05

Table 3-8 Kostner LAN Address Associations

IP Address	Ethernet Address
192.20.3.1 (Host 1)	00-03-49-C5-12-31
192.20.3.2 (Host 2)	00-03-49-C5-12-32
192.20.3.3 (Host 3)	00-03-49-C5-12-33
192.20.3.4 (Host 4)	00-03-49-C5-12-34
192.20.3.5 (Router)	00-03-49-C5-12-35

Intra-LAN Communication

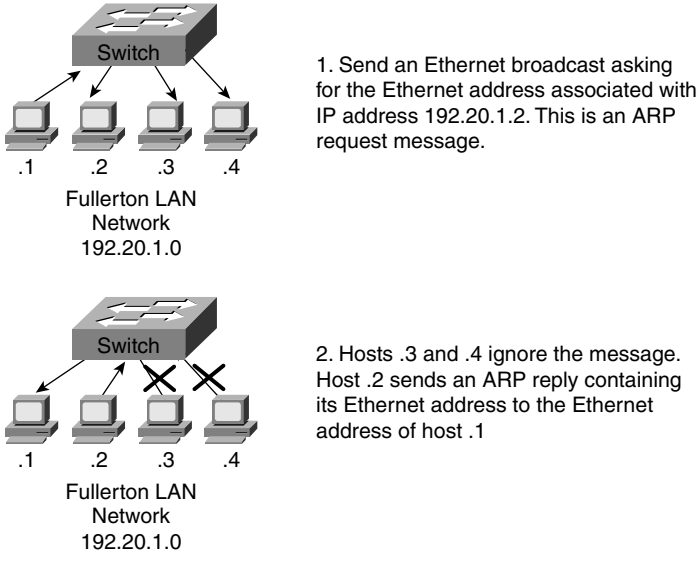
In Figure 3-6, the host with IP address 192.20.1.1 on the Fullerton LAN wants to send data to the host with IP address 192.20.1.2 on the same LAN. The source and destination IP addresses are

- Source: **192.20.1**.1
- Destination: **192.20.1**.2

The source host knows that the destination IP address is on the same network because

- Both source and destination network numbers are Class C.
- Both Class C network numbers are the same; therefore, they both point to the same network.

Figure 3-6 Intra-LAN Address Resolution



The source host knows the destination IP address, but not the destination Ethernet address. The source host needs to resolve the destination Ethernet address from the destination IP address. This is accomplished by using the ***Address Resolution Protocol (ARP)***. The source host sends an Ethernet broadcast to the switch. Like the IP broadcast, an Ethernet broadcast is signified by setting the destination Ethernet address to all 1s or FF:FF:FF:FF:FF:FF. The source Ethernet address is set to the Ethernet address of the host sending the broadcast. The ARP message contains the destination IP address or 192.20.1.2. When the Ethernet switch receives the broadcast message, it is sent to all hosts on the network except for the

host that sent the message. All hosts on the Fullerton LAN will receive the broadcast and inspect the IP address in the message. If the IP address is not the IP address of the host that received the message, the message will be ignored. When the host with IP address 192.20.1.2 receives the ARP message, it will respond back to the sender with its Ethernet address. Now the host at 192.20.1.1 has resolved the Ethernet address for the host with IP address 192.20.1.2.

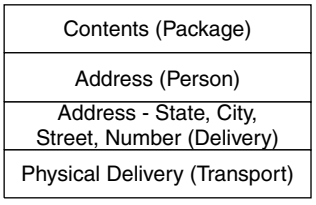
Host .1 on the Fullerton LAN receives the ARP request and stores that association between the Ethernet and IP addresses for host .2 in an ARP table. Storing this information allows host .1 to send additional messages to host .2 without having to send an ARP request each time. An example of a typical ARP table is shown in the following output:

```
Interface: 192.20.1.1
Internet Address       Physical Address         Type
192.20.1.2             00-03-47-92-9C-70        dynamic
```

The physical address is the Ethernet address associated with IP address 192.20.1.2. *Dynamic* means that this association was learned using ARP.

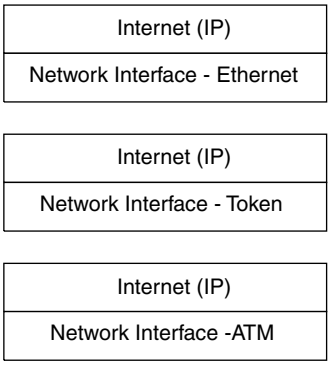
At this point, you might be wondering why we have two addresses. Why not use either the IP address or the Ethernet address. Why use both? The clue is in the ARP table shown earlier. An Ethernet address is a physical address. It is "burned in" to the Ethernet card and is sometimes referred to as a burned-in address (BIA). An IP address is a logical address that was assigned to the host. In this case, the host happens to use Ethernet for sending messages on the LAN. Other technologies exist that can be used by the computers to send messages, such as Token ring or Asynchronous Transfer Mode (ATM). If you use ATM on the Fullerton LAN instead of Ethernet, you should expect that you are still able to send messages between computers. An ATM address is 20 bytes while an Ethernet address is 6 bytes. In other words, the logical addressing (IP) should be independent of the physical addressing (Ethernet, Token Ring, ATM). Does this sound familiar? In Chapter 1, "Routing and Switching in Everyday Life," you learned a layered model for the postal delivery system. (See Figure 3-7.)

Figure 3-7 Layered Postal Delivery Model



For this model, you learned that the address should not be dependent on the contents, and that the physical delivery should not be dependent on the address. The layers in this model are independent. In the same way, you need a layered model for the Internet. With what you've learned, you can start constructing the layer model for the Internet. In Figure 3-8, the lowest layer is the network interface layer.

Figure 3-8 Partial Layered Internet Model



The network interface layer is concerned with the physical, electrical, and addressing requirements for the particular technology used to deliver the messages. The IP layer is a logical layer concerned with being able to route a message between endpoints. The IP layer in the Internet model should be independent from the network interface layer. This independence allows you to change the technology used at the network interface layer without having to modify the IP layer.

Inter-LAN Communication

The host with IP address 192.20.1.1 on the Fullerton LAN wants to send data to the host with IP address 192.20.3.3 on the Kostner LAN. The source and destination IP addresses are

- Source: **192.20.1**.1
- Destination: **192.20.3**.3

The source host knows that the destination IP address is on a different network because

- Both source and destination network numbers are Class C.
- The source and destination Class C network numbers are not the same; therefore, the source and destination computers are on different networks.

The host on the Fullerton LAN doesn't have to know how to get a message to the host on the Kostner LAN. That is the function of the router. Because the source host knows that the destination is on a different LAN or network, the host knows that it must send the message to the router. Each host has been configured with the IP address of the router interface that connects to their LAN. The router is the gateway to the rest of the world, so the IP address of the router is called the *default gateway*. In other words, if a host is sending a message to a different LAN, the message must first be sent to the default gateway, or router, or last resort. The process for inter-LAN communication is

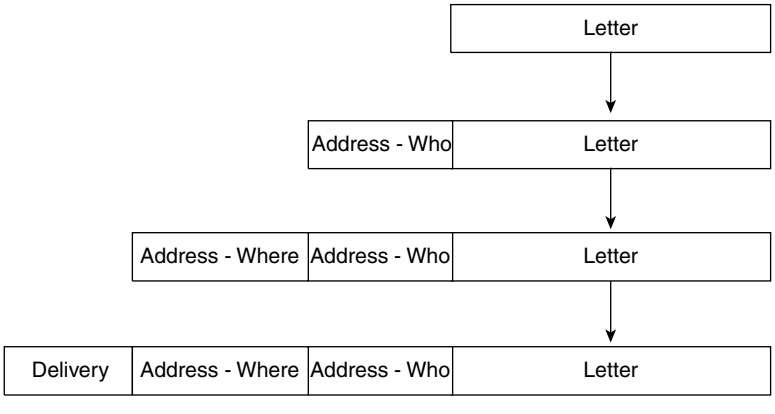
1. Send an ARP broadcast asking for the Ethernet address associated with the default gateway (192.20.1.5).
2. The router responds with the Ethernet address of the interface that is connected to the source LAN (00-03-47-92-9C-73).
3. Host 192.20.1.1 stores the router's IP address, and associated Ethernet address in its local ARP table. The ARP table now contains

```
Interface: 192.20.1.1
Internet Address      Physical Address      Type
192.20.1.2            00-03-47-92-9C-70     dynamic
192.20.1.5            00-03-47-92-9C-73     dynamic
```

4. The source host sends the message to the router.
5. The router removes the source and destination Ethernet addresses from the message and inspects the destination IP address (192.20.3.3).
6. The router determines that the destination LAN is network 192.20.3.0 and the destination host IP address is 192.20.3.3.
7. The router sends an ARP request on the Kostner LAN asking for the Ethernet address associated with IP address 192.20.3.3.
8. Host 192.20.3.3 on the Kostner LAN sends an ARP reply containing its Ethernet address to the router (00-03-49-C5-12-33).
9. The router sends the message to the Ethernet address of host 192.20.3.3.

This process is similar to how mail is delivered. Figure 3-9 shows the flow of a letter down the protocol stack that was developed for the postal system.

Figure 3-9 Flow of a Letter Down the Mail Protocol Stack

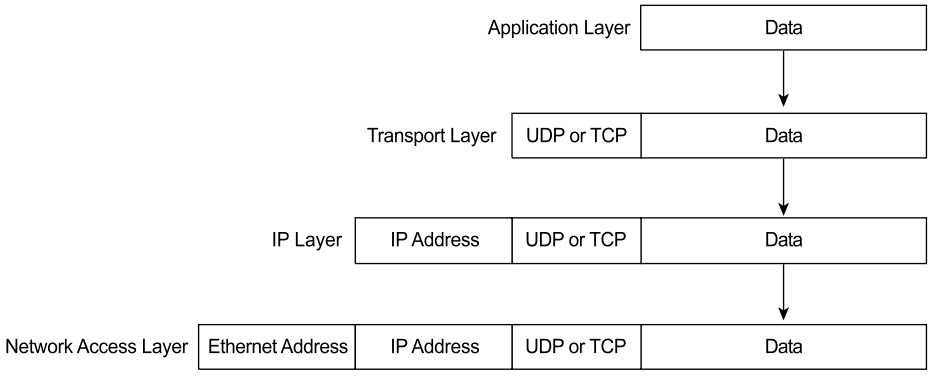


The letter is sent down to the Addressing Person, or Who layer where it is placed, or encapsulated, in an envelope. The envelope is sent to the Addressing Where layer and the state, city, street name, and street number information are added. Remember that you have logically separated the Who from the Where information, because the Who information is not used to deliver the letter. Finally, the

envelope is passed to the Delivery layer where it is encapsulated or placed into whatever delivery means is being used (wagon, horse, truck, and so on).

As the letter makes its way through the postal delivery system, it passes through one or more post offices. At each post office the letter is removed from the delivery layer, and the destination address is inspected. Based on the destination address, the post office makes a routing decision and the letter is again sent back to the delivery layer and encapsulated (placed) in a new means of delivery. Between the source of the letter and the letter's destination, the means of delivery at each post office changes, but the source and destination addresses remain the same. This process can be used to better understand the delivery of an electronic message through a network. (See Figure 3-10.)

Figure 3-10 Flow of Data Down the IP Stack



Your application generates the data to be sent to another host. This data could be an e-mail, an instant message, a request for a web page, and so on. The data is sent to the first addressing layer where an application identifier is placed on the data. Think of this as the Who part of the address. As with the postal system, this information is not used to deliver the data, but to identify which application should receive the data after it arrives at the destination. After the application identifier is placed on the data, the next layer in the protocol stack adds the source and destination IP addresses. Finally, the network interface layer adds the source and

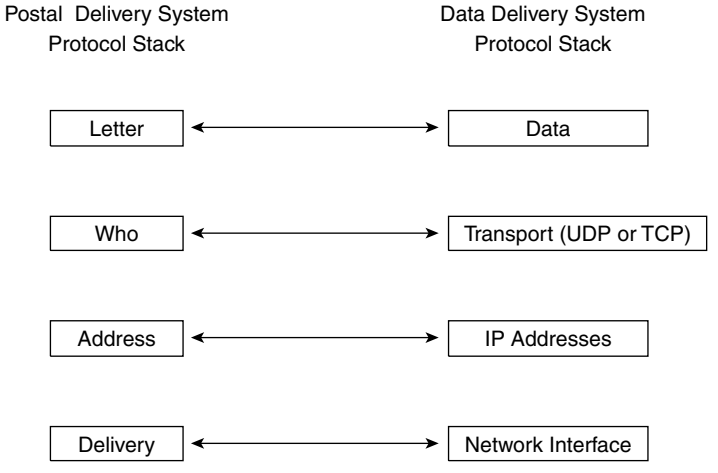
destination Ethernet addresses on the package (assuming the host is using Ethernet), and the package is transmitted toward the destination. In the Internet, the package of data is called a ***packet***.

For intra-LAN communication, the receiving host inspects the destination Ethernet address, and accepts the package if the host sees its own Ethernet address. If it does, the Ethernet addresses (source and destination) will be stripped off, and the remaining package will be sent to the IP layer. The IP layer inspects the destination IP address to verify that the package is meant for this host. If it is, the IP address is stripped off and sent to the application identification layer. After the application has been identified, this information is stripped off and the data is sent to the proper application.

For inter-LAN communication, the package is sent to the router. The router inspects the destination Ethernet address and accepts the package if the router sees its own Ethernet address. If it does, the Ethernet addresses (source and destination) are stripped off and the remaining package is sent to the router's IP layer. The destination IP address is inspected, and the router consults the routing table to determine the interface it needs to use to send the package to the destination. The router looks for the destination IP and Ethernet address association in the ARP table. If the association is not in the ARP table, the router uses ARP to learn the destination Ethernet address associated with the destination IP address. The package is sent back to the network interface layer, and the package is encapsulated using new source and destination Ethernet addresses. Finally, the package is sent to the host, and the host will perform the same functions as mentioned for intra-LAN delivery.

The package might have to travel through more than one router. At each router, the same process takes place. The old Ethernet source and destination addresses are removed, the IP routing table is consulted, and new source and destination Ethernet addresses are applied. But no matter how many routers the package goes through, the source and destination IP addresses do not change. Only the network interface layer addresses change. The analogy between the layers of the mail and data delivery systems is shown in Figure 3-11.

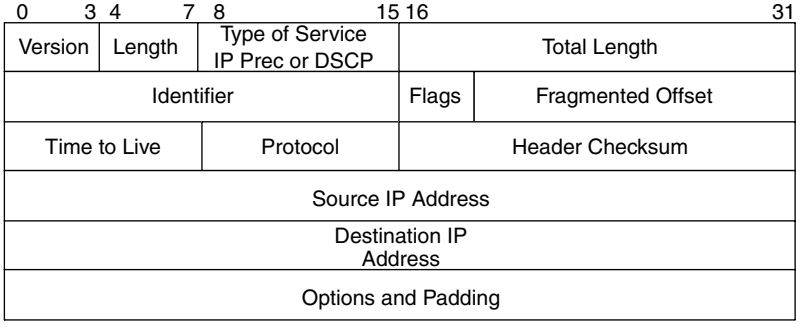
Figure 3-11 Mail and Electronic Data Delivery Protocol Stacks



IP Header Format

Unlike the post office, a router or computer cannot determine the size of a package without additional information. A person can look at a letter or box and determine how big it is, but a router cannot. Therefore, additional information is required at the IP layer, in addition to the source and destination IP addresses. Figure 3-12 is a logical representation of the information that is used at the IP layer to enable the delivery of electronic data. This information is called a *header*, and is analogous to the addressing information on an envelope. A header contains the information required to route data on the Internet, and has the same format regardless of the type of data being sent. This is the same for an envelope where the address format is the same regardless of the type of letter being sent.

Figure 3-12 IP Header Format



The fields in the IP header and their descriptions are

- **Version**—A 4-bit field that identifies the IP version being used. The current version is 4, and this version is referred to as IPv4.
- **Length**—A 4-bit field containing the length of the IP header in 32-bit increments. The minimum length of an IP header is 20 bytes, or five 32-bit increments. The maximum length of an IP header is 24 bytes, or six 32-bit increments. Therefore, the header length field should contain either 5 or 6.
- **Type of Service (ToS)**—The 8-bit ToS uses 3 bits for IP Precedence, 4 bits for ToS with the last bit not being used. The 4-bit ToS field, although defined, has never been used.
- **IP Precedence**— A 3-bit field used to identify the level of service a packet receives in the network.
- **Differentiated Services Code Point (DSCP)**—A 6-bit field used to identify the level of service a packet receives in the network. DSCP is a 3-bit expansion of IP precedence with the elimination of the ToS bits.
- **Total Length**—Specifies the length of the IP packet that includes the IP header and the user data. The length field is 2 bytes, so the maximum size of an IP packet is $2^{16} - 1$ or 65,535 bytes.
- **Identifier, Flags, and Fragment Offset**—As an IP packet moves through the Internet, it might need to cross a route that cannot handle the size of the packet. The packet will be divided, or fragmented, into smaller packets and reassembled later. These fields are used to fragment and reassemble packets.

- **Time to Live (TTL)**—It is possible for an IP packet to roam aimlessly around the Internet. If there is a routing problem or a routing loop, then you don't want packets to be forwarded forever. A routing loop is when a packet is continually routed through the same routers over and over. The TTL field is initially set to a number and decremented by every router that is passed through. When TTL reaches 0 the packet is discarded.
- **Protocol**—In the layered protocol model, the layer that determines which application the data is from or which application the data is for is indicated using the Protocol field. This field does not identify the application, but identifies a protocol that sits above the IP layer that is used for application identification.
- **Header Checksum**—A value calculated based on the contents of the IP header. Used to determine if any errors have been introduced during transmission.
- **Source IP Address**—32-bit IP address of the sender.
- **Destination IP Address**—32-bit IP address of the intended recipient.
- **Options and Padding**—A field that varies in length from 0 to a multiple of 32-bits. If the option values are not a multiple of 32-bits, 0s are added or padded to ensure this field contains a multiple of 32 bits.

The IP Precedence field can be used to prioritize IP traffic. (See Table 3-9.) This is the same as the postal system having different classes of mail such as priority, overnight, and 2-day delivery. Routers can choose to use this field to give preferential treatment to certain types of IP traffic.

Table 3-9 IP Precedence Values

Precedence Value	Meaning
000 (0)	Routine or Best Effort
001 (1)	Priority
010 (2)	Immediate
011 (3)	Flash

continues

Table 3-9 IP Precedence Values (continued)

Precedence Value	Meaning
100 (4)	Flash Override
101 (5)	Critical
110 (6)	Internetwork Control
111 (7)	Network Control

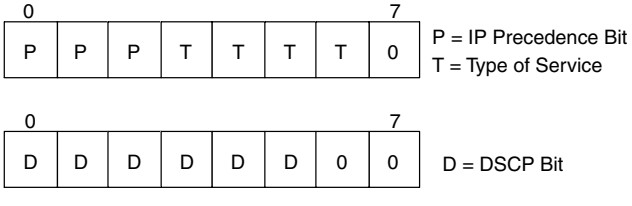
The ToS bits were originally designed to influence the delivery of data based on delay, throughput, reliability and cost. (See Table 3-10.) They are usually not used and are therefore set to zero.

Table 3-10 Type of Service Values

ToS Value	Meaning
0000 (0)	Normal Delivery
0001 (1)	Minimize Cost
0010 (2)	Maximize Reliability
0100 (4)	Maximize Throughput
1000 (8)	Minimize Delay

The IP Precedence field can have 8 or 2^3 possible values. Routers use two of these values, 6 and 7, for routing protocol traffic. That leaves six values that can be used to prioritize user traffic. Because the ToS bits are typically not used, the IP Precedence field can be extended from 3 to 6 bits by using 3 bits from the ToS field. (See Figure 3-13.)

Figure 3-13 IP Header Type of Service (ToS) Field



This new field is called the Differentiated Services Code Point (DSCP). That gives you 64 or 2^6 possible values that can be used to prioritize traffic. Although there are 64 possible DSCP values, only 14 are used typically. (See Table 3-11 and the explanation that follows.)

Table 3-11 Differentiated Services Code Point Values

DSCP Value	Meaning	Drop Probability	Equivalent IP Precedence Value
101 110 (46)	High Priority Expedited Forwarding (EF)	N/A	101 – Critical
000 000 (0)	Best Effort	N/A	000 – Routine
001 010 (10)	AF11	Low	001 – Priority
001 100 (12)	AF12	Medium	001 – Priority
001 110 (14)	AF13	High	001 – Priority
010 010 (18)	AF21	Low	001 – Immediate
010 100 (20)	AF22	Medium	001 – Immediate
010 110 (22)	AF23	High	001 – Immediate
011 010 (26)	AF31	Low	011 – Flash
011 100 (28)	AF32	Medium	011 – Flash
011 110 (30)	AF33	High	011 – Flash
100 010 (34)	AF41	Low	100 – Flash Override
100 100 (36)	AF42	Medium	100 – Flash Override
100 110 (38)	AF43	High	100 – Flash Override

Notice that the first 3 bits of the DSCP value are the 3 bits from the IP precedence. An IP precedence of 000 maps into a DSCP value of 000 000, and both represent best effort delivery. An IP precedence of 101 (Critical) maps into a DSCP value of 101 110 (High Priority or Expedited Forwarding). The remaining 4 IP precedence

values are each mapped into 3 DSCP values. The additional 3-bit portion is used to identify a drop probability within one of the four assured forwarding (AF) classes.

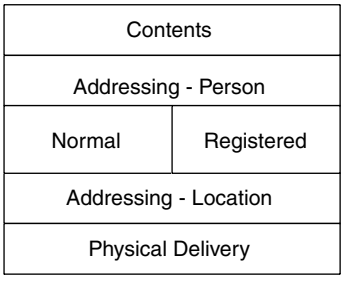
This discussion of the contents of the IP header is meant as an overview. If you are interested in learning more details regarding the IP header, refer to the references at the end of this chapter. The important concept to take away from this discussion is that the IP header contains the source and destination IP addresses. Routers use the destination IP address to determine a route; therefore, the IP layer in the layered model is the routing layer.

At this point, we could stop our discussion of the layered protocol model. This book is about routing, and routing is the second or third layer depending on which model is used. A router does not care what application sent the data, or how the application is going to receive the data. The job of the router is to get the packet to the proper destination. It is then the responsibility of the destination host to deliver the data to the application. The incomplete layered model in Figure 3-8 is sufficient for the remainder of this book. But, to be complete, let's go ahead and finish the model.

TCP/IP Layered Protocol Model

There are different types of service that you can use when delivering a letter. You can use a best effort model. This means that you simply place a stamp on the letter and drop it in a mailbox. How do you know if the letter was delivered? You don't. Not unless the recipient somehow tells you they received the letter. If you want to ensure delivery, you could send a registered letter. After the letter has been delivered, you will receive an acknowledgment from the post office. The layered postal delivery model needs to be modified to include this feature. (See Figure 3-14.) Although IP is used to deliver packets, and TCP and UDP are transport level protocols, the layer model for the Internet is usually referred to as the TCP/IP model.

Figure 3-14 Post Office Layered Protocol Model



At the Addressing – Person layer, two options have been added:

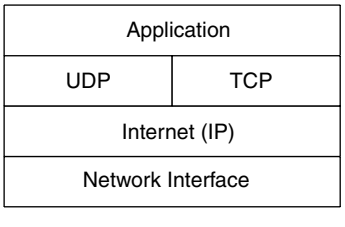
- **Normal delivery**—The sender will not receive any acknowledgment that the letter has been delivered.
- **Registered mail**— The sender will receive an acknowledgment after the letter has been delivered.

The Internet layered model has two additional protocols that are equivalent to normal and registered mail:

- **Transmission Control Protocol (TCP)**—The equivalent of a registered letter. When data is received at the destination host, an acknowledgment is sent back to the sender.
- **User Datagram Protocol (UDP)**—The equivalent of normal mail. An acknowledgment is not sent back to the sender.

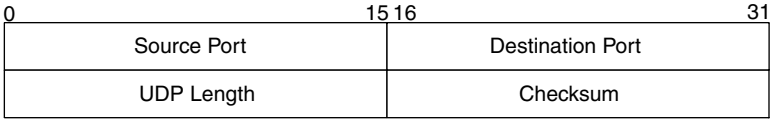
Integrating TCP and UDP into the Internet model gives you the complete TCP/IP layered protocol model. (See Figure 3-15.)

Figure 3-15 TCP/IP Layered Model or Protocol Stack



The port information in the UDP header is used to identify the sending and receiving applications. (See Figure 3-16.)

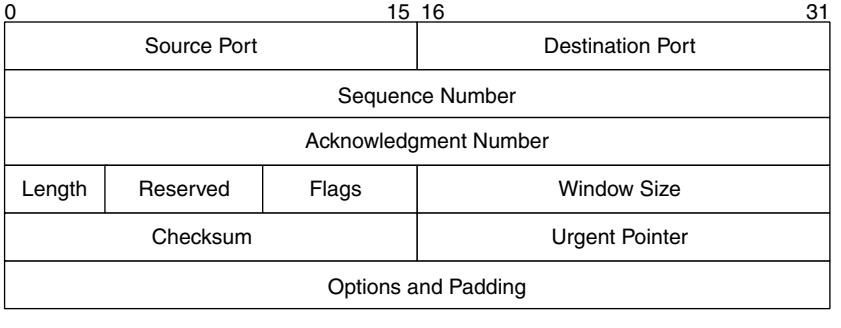
Figure 3-16 UDP Header Format



The source port is equivalent to the person who is sending the letter. The destination port is equivalent to the person who is to receive the letter. Applications initially will use a well-known port number. For example, if you are using a web browser to contact the Cisco website, your application will send a message to port 80 at IP address 198.133.219.25. The source port is usually assigned dynamically for the application and is included in the UDP header. When the webserver for Cisco.com sends a reply, it will send it back to the dynamic port number that was assigned for the sender's application. The destination IP address is used to reach the host running the web server, and the destination port number is used to reach the proper application. The combination of a port number and an IP address is called a ***socket***. A socket is sufficient to identify a particular application on a specific host.

The TCP header is similar to the UDP header with additional fields to enable acknowledgments. (See Figure 3-17.)

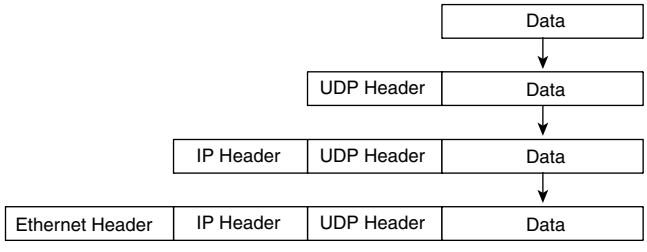
Figure 3-17 TCP Header Format



The source and destination port numbers serve the same function as they did in the UDP header. The remaining fields are used to send the equivalent of registered mail. The operation of the TCP protocol can be complex. If you are interested in learning more, consult the reference list at the end of the chapter.

Before moving on to the next exciting topic, let's trace the flow through the TCP/IP protocol stack. (See Figure 3-18.)

Figure 3-18 Data Flow Through the TCP/IP Protocol Stack



UDP Header - Source and Destination Port Numbers
IP Header - Source and Destination IP Addresses
Ethernet Header - Source and Destination Ethernet Addresses

The application sends the data, such as the text of an e-mail, you are sending to UDP or TCP where the destination and source port numbers are recorded. The IP layer adds the source and destination IP addresses, and sets the protocol field to UDP or TCP depending on what is being used. The IP layer then hands the packet off to the network interface layer. This example uses Ethernet, so the network interface layer adds the source and destination Ethernet addresses. Finally, the entire thing is sent to the network to be sent to the destination.

Upon receipt of an IP packet by a host, the destination Ethernet address is first inspected. If it matches the host's Ethernet address, the Ethernet header is stripped off, and the remaining part of the packet is sent up the protocol stack to the IP layer. The IP layer looks at the destination IP address. If it is the correct IP address, the IP layer strips off the IP header, and sends what is left either to TCP or UDP—depending on the setting of the protocol field. UDP or TCP then uses the destination port number to send the data to the correct application.

Classless Internet Addressing

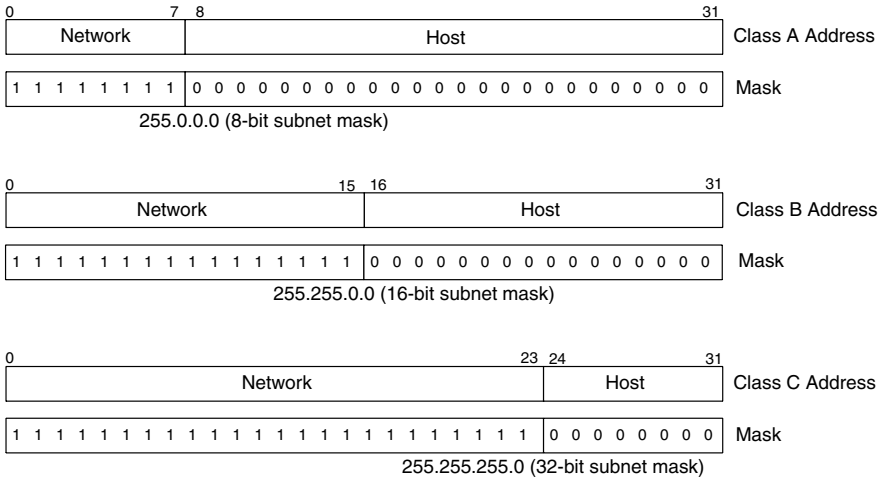
Using a classful IP addressing format worked well when the Internet was relatively small. But as the number of networks on the Internet grew, the limitations of classful addresses became apparent. The Class A address space contains only 125 usable networks in the range 0–127 because networks 0 and 127 are reserved, and network 10 is used for private addressing. Each of these 125 Class A networks could theoretically contain $2^{24} - 2$ or 16,777,214 hosts, but it's not realistic to have more than 16 million hosts on the same network. Therefore, in the early 1990s, the Internet moved away from a classful address space to a classless address space. In other words, the number of bits used for the network portion of an IP address became variable instead of fixed.

The network portion of classful IP addresses is fixed. For the network portion of an IP address, Class A addresses use 8 bits, Class B addresses use 16 bits, and Class C addresses use 24 bits. A router could determine the address class by inspecting the first byte of the address. A value of 1–126 is Class A, 128–191 is Class B, and 192–223 is Class C.

For classless IP addressing, there is no longer a relationship between the number of bits used in the network portion and the value of the first byte of the address. A different method has to be used to determine the size of the network portion of an IP address. This new method allows you to borrow bits that are normally used for the host portion of an IP address, and use them to extend the network portion of an IP address.

A router is a computer of sorts, and can therefore manipulate binary numbers quite well. It would seem natural to use a 1 to identify a bit in an IP address that is part of the network address, and a 0 to identify a bit that is used as the host address. These bits can be thought of as masking off the network address from the host address. An IP address is 32 bits, so a 32-bit mask is needed to determine the network and host components of an IP address. Figure 3-19 contains the natural mask values for Class A, B, and C addresses.

Figure 3-19 Classful IP Address Masks



There are two common ways to refer to the mask that is used to determine the number of bits used for the network component of an IP address. The first is to use the number of 1 bits in the mask. A Class A mask is an 8-bit mask, Class B is a 16-bit mask, and Class C is a 24-bit mask. The other way is to represent the mask as / (slash) and then the number of 1 bits in the mask. Class A is /8 (slash 8), Class B is /16 (slash 16), and Class C is /24 (slash 24). An important rule is that the number of 1s and 0s in a mask must be contiguous (all the 1s must be together and all the 0s must be together). For example

11111111 11111111 00000000 00000000 is a valid mask.

11111111 00111111 00000000 00000111 is not a valid mask.

Using a mask to determine the network component of an IP address is called a *bitwise logical AND operation*. Bitwise AND is equivalent to bitwise multiplication:

A * 1 = A

A * 0 = 0 where A = 0 or 1

A router can determine the network component of the classful IP address 156.26.32.1 by using a mask as shown:

156.26.32.1

AND

255.255.0.0

Equals

156.26.0.0

This might seem like a trivial operation. For classful addresses, this is a fair statement because the network component is on an easy-to-use byte boundary. But you want to be able to switch from classful to classless addressing, and you will need a mask to do that.

As an introduction to classless addressing, assume that your company has been assigned the Class B address 156.26.0.0. If you use this as a classful address, you can have one network with 2^{16}-2 or 65,534 hosts. You would like to have more than one network with fewer hosts on each network. This means you will have to create subnets from the assigned Class B address space. Instead of using a 16-bit mask, or /16, see what happens if you use a 17-bit subnet mask:

IP Address = 156.26.0.0

Subnet Mask = 255.255.128.0

The Class B part, or 156.26, is fixed and cannot be changed. But your company owns the following 16 bits, so they can be any value you want. The seventeenth bit of your network address can either be a 0 or a 1. If it is 0, that identifies network 156.26.0.0. If the seventeenth bit is a 1, that identifies network 156.26.128.0. By borrowing 1 bit from the standard host portion of the IP address and assigning it to the network portion, you have created two subnets of the Class B address space 156.26. The first subnetwork has host addresses in the range 156.26.0.1–156.26.127.254.

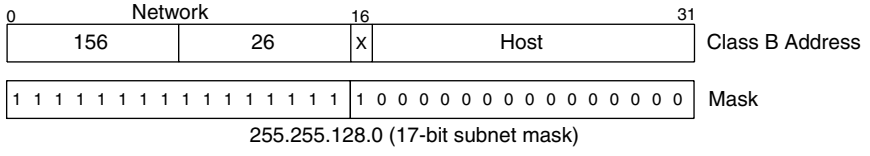
The broadcast address is 156.26.127.255.

The second subnetwork has host addresses in the range 156.26.128.1–156.26.255.254.

The broadcast address is 156.26.255.255.

This operation is shown in Figure 3-20.

Figure 3-20 Subnetting a Class B Address with a 17-bit Mask



If you use 2 additional bits, or a /18 bit mask, you will have four subnets. These four subnets are identified by the four values possible with 2 bits:

0 0

0 1

1 0

1 1

Remember, the network is identified by setting the host portion of the IP address to 0. So, the first subnet using an 18-bit mask is 156.26.0.0.

The second subnet is determined by calculating the value of the third byte when the most significant bits are 0 1:

0 1 0 0 0 0 0 0 = 64

Subnet 2 has a network address of 156.26.64.0.

The third subnet is determined by calculating the value of the third byte when the most significant bits are 1 0:

1 0 0 0 0 0 0 0 = 128

Subnet 2 has a network address of 156.26.128.0.

And the fourth subnet is determined by calculating the value of the third byte when the most significant bits are 1 1:

1 1 0 0 0 0 0 0 = 192

Subnet 4 has a network address of 156.26.192.0.

If you continue this logic, you obtain the information in Table 3-12.

Table 3-12 Number of Networks and Hosts for a Subnetted Class B Network

Subnet Mask in Bits	Number of Networks	Number of Hosts
16 (Class B)	1	65,534
17	2	32,766
18	4	16,382
19	8	8190
20	16	4096
21	32	2048
22	64	1022
23	128	510
24	256	254
25	512	126
26	1,024	62
27	2,048	30
28	4,096	14
29	8,192	6
30	16,384	2

The first entry is for a Class B network. Increase the subnet mask by 1 bit, and calculate the number of subnetworks and hosts to find the remaining entries. The number of possible subnetworks is 2 raised to the power of the number of extra bits used for the network. The number of hosts is 2 to the power of the bits left over for the host portion of the address – 2 (broadcast and "this" network addresses).

It seems that there are two entries missing in Table 3-12. One for a 31-bit subnet mask and one for a 32-bit subnet mask. You can't have a 32-bit subnet mask because there would be no bits left over for host addresses. A 31-bit mask would leave only 1 bit for the host addresses, either 0 or 1. The broadcast address is obtained by setting all the host bits to 1. The "this" network address is found by setting all the host bits to 0. So, if you used 31-bits, the addresses you would have available are only the broadcast and "this" network addresses.

You do not have to use only one subnet mask to divide the 156.26.0.0 address space in subnetworks. You can use different masks on different networks. For example, assume you have the following requirements for your company's network:

- A maximum of 60 Class C size networks (1–254 hosts)
- A maximum of 14 networks having a maximum of 10 hosts
- Four point-to-point networks

You need to satisfy these requirements, and you want to have addresses in reserve that you can use if your company expands. Where do you start? There is not just one correct way of doing this. You have a Class B address space assigned to you, and you shall see that this will not be that difficult. First, let's subnet the Class B address space into four equal size pieces. For four subnets, you will need to use 2 bits from the host address or a /18 subnet mask. The third byte of the IP address is divided as

N N H H H H H H (2 bits for the network and 6 bits for the host)

To determine the network numbers, first set the host bits to 0:

N N 0 0 0 0 0 0

The possible network values for the third byte are

0 0 0 0 0 0 0 0 = 0

0 1 0 0 0 0 0 0 = 64

1 0 0 0 0 0 0 0 = 128

1 1 0 0 0 0 0 0 = 192

The 156.26 address space is now divided into the following networks:

156.26.0.0

156.26.64.0

156.26.128.0

156.26.192.0

To satisfy the first requirement of a maximum of 60 Class C size networks, subnet the 156.26.0.0/18 address into Class C size or /24 subnets. How many Class C size subnets will this provide? We are using an additional 6 bits to subnet the 156.26.0.0/18 network, and $2^6 = 64$ subnets. This will be sufficient to satisfy the first requirement. The Class C networks will have the following addresses:

156.26.0.0/24

156.26.1.0/24

156.26.2.0/24

...

156.26.62.0/24

156.26.63.0/24

How were these network numbers determined? The 156.26.0.0/18 network was derived from the 156.26.0.0/16 network. The first 16 bits are fixed and equal to

156.26. The next 2 bits are fixed and equal to 0 because this is the subnet used for the Class C size networks. Therefore, the possible range of values for the third byte are

0 0 0 0 0 0 0 0 = 0

0 0 0 0 0 0 0 1 = 1

0 0 0 0 0 0 1 0 = 2

...

0 0 1 1 1 1 1 0 = 62

0 0 1 1 1 1 1 1 = 63

For the first requirement, use networks 156.26.0.0/24 through 156.26.59.0/24.

To satisfy the second requirement, use the last Class C size network, 156.26.63.0, and subnet it to the proper size. For a maximum of 10 hosts, you will need 4 bits for the host address. With 4 bits, a network can support 14 hosts (16 – 2). Because a Class C size network is being subnetted, there are only 8 bits to work with (the last byte). Four bits are needed for the hosts, which leaves 4 bits for the network. The requirement is 14 networks, and 4 network bits can support 16 networks. The last byte is divided, so 4 bits are used for the network and 4 bits for the host:

N N N N H H H H

The networks addresses are

0 0 0 0 0 0 0 0 = 0 156.26.63.0/28

0 0 0 1 0 0 0 0 = 16 156.26.63.16/28

0 0 1 0 0 0 0 0 = 32 156.26.63.32/28

.....

1 1 1 0 0 0 0 0 = 224 156.26.63.224/28

1 1 1 1 0 0 0 0 = 240 156.26.63.240/28

The host addresses are

156.26.63.1–156.26.63.14

156.26.63.17–156.26.63.30

156.26.63.33–156.26.63.46

...

156.26.63.225–156.26.63.238

156.26.63.241–156.26.63.254

For the final requirement of four point-to-point networks, the 156.26.63.240 network will be subnetted using a 30-bit mask. A point-to-point network requires only two host addresses.

There are 4 bits available on the 156.26.63.240/28 subnet. Two are needed for the host bits. The two remaining bits are sufficient for the four point-to-point networks that are required. The last byte of the 156.26.63.240 is used for the final subnetting operation:

1 1 1 1 N N H H

The network numbers using a 30-bit mask are

1 1 1 1 0 0 0 0 = 240 156.26.63.240

1 1 1 1 0 1 0 0 = 244 156.26.63.244

1 1 1 1 1 0 0 0 = 248 156.26.63.248

1 1 1 1 1 1 0 0 = 252 156.26.63.252

The host addresses are

156.26.63.241 and 242

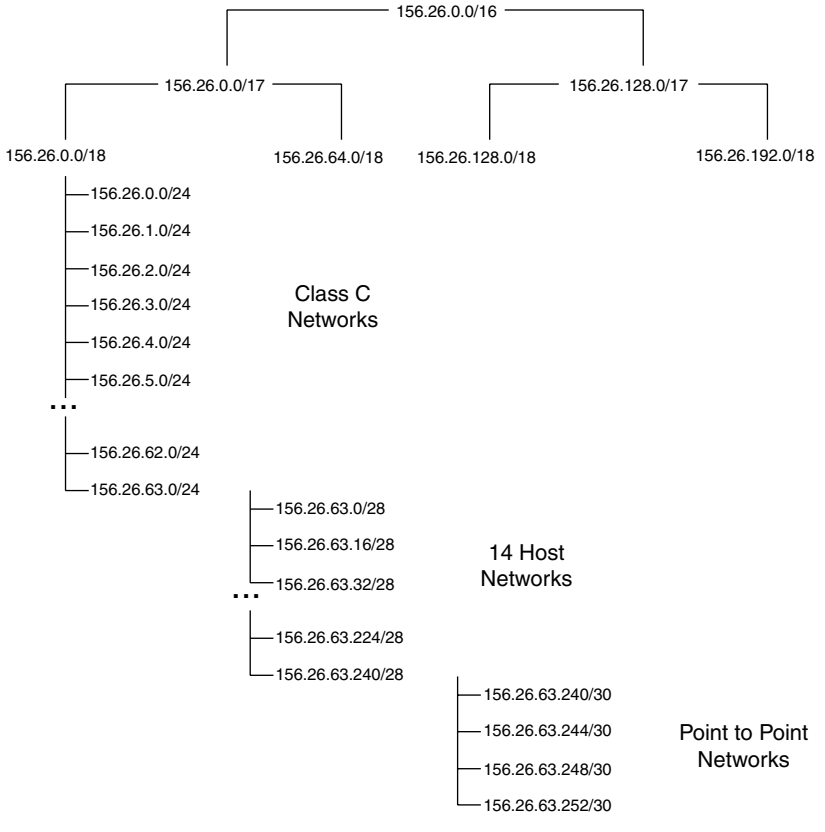
156.26.63.245 and 246

156.26.63.249 and 250

156.26.63.253 and 254

The final plan is shown in Figure 3-21.

Figure 3-21 Subnetting a Class B Address



If this is your first experience dealing with subnet masks and you find it a bit confusing, take comfort in the fact that this is normal. Subnets and subnet masks take time to master. Get some paper and a pencil and practice, practice, practice. To aid in your understanding, try the following problems:

1. What is the broadcast address for network 156.26.0.0/16?

   Answer: Set the 16 host bits to 1 to obtain 156.26.255.255.

2. What is the broadcast address for network 156.26.0.0/24?

   Answer: Set the 8 host bits to 1 to obtain 156.26.0.255.

3. What is the broadcast address for network 156.26.0.0/28?

   Answer: Set the 4 host bits to 1 to obtain 156.26.0.15.

4. The Class C address 195.14.22.0 is subnetted using a 27-bit subnet mask. How many subnets are there and what are the network numbers?

   Answer: The natural mask for a Class C address is /24. Therefore, 33 additional bits are used for the subnet, $2^3 = 8$, so there are eight subnets. The 3 additional network bits are taken from the fourth byte so the network numbers are

   0 0 0 0 0 0 0 0 = 0 195.14.22.0/27

   0 0 1 0 0 0 0 0 = 32 195.14.22.32/27

   0 1 0 0 0 0 0 0 = 64 195.14.22.64/27

   0 1 1 0 0 0 0 0 = 96 195.14.22.96/27

   1 0 0 0 0 0 0 0 = 128 195.14.22.128/27

   1 0 1 0 0 0 0 0 = 160 195.14.22.160/27

   1 1 0 0 0 0 0 0 = 192 195.14.22.192/27

   1 1 1 0 0 0 0 0 = 224 195.14.22.224/27

5. What is the range of host addresses for the network 195.14.22.64/27?

   Answer: 195.14.22.65 – 195.14.22.94

6. What is the broadcast address for network 195.14.22.64/27?

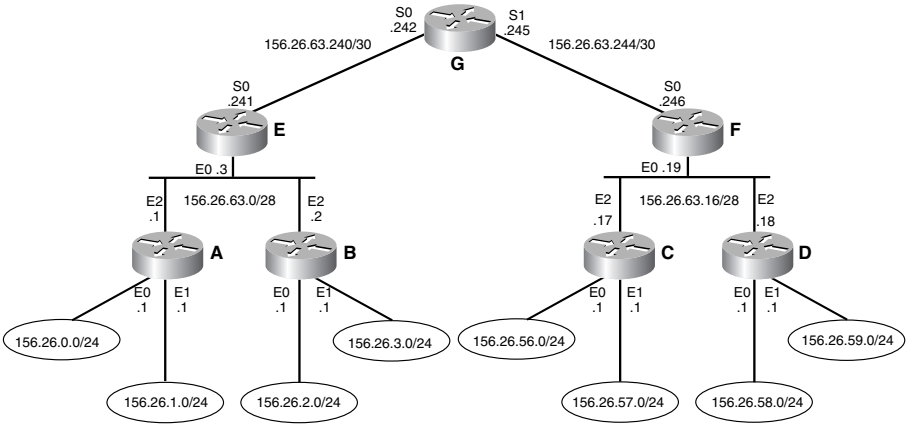
   Answer: 64 = 0 1 0 0 0 0 0 0, so the broadcast address is:

   0 1 0 1 1 1 1 1 = 95 or 195.14.22.95

IP Routing and Route Summarization

The network in Figure 3-22 is a partial implementation of the addressing plan developed for the 156.26.0.0 network.

Figure 3-22 Example Network for Route Summarization



Routers A, B, C, and D are access routers and each one connects to two Class C size networks. Routers E and F are the distribution routers, and Router G is the core router. The terminology used in Figure 3-22 is explained in Figure 3-23.

The network in Figure 3-22 has 12 subnets, so each router will have 12 entries in its IP routing table. The routing table for Router G is listed in Table 3-13. Initially, the only routes in the IP routing table are the directly connected networks. The other subnets need to be learned either statically or dynamically. *Statically* means that every route has to be manually entered on every router. The network has 7 routers so 7 * 12, or 84, routes would need to be entered for IP routing to work. Certainly this can be done, but it would take some time and would be prone to error. Imagine entering all routes statically for a network with hundreds of routers and thousands of routes. This is not a scalable solution. A better solution is to use a dynamic IP routing protocol that will dynamically advertise routes throughout your network. The later chapters will discuss IP routing protocols. For now, assume that all the routes have been entered statically.

Figure 3-23 Network Terminology

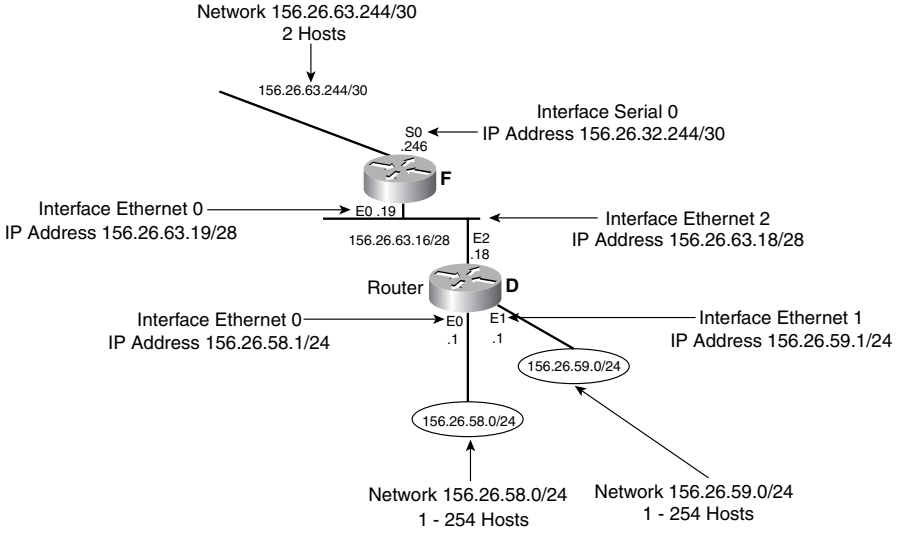


Table 3-13 IP Routing Table for Router G

Route	Output Interface
156.26.63.240/30	Directly connected, Serial 0
156.26.63.244/30	Directly connected, Serial 1
156.26.63.0/28	Serial 0
156.26.63.16/28	Serial 1
156.26.0.0/24	Serial 0
156.26.1.0/24	Serial 0
156.26.2.0/24	Serial 0
156.26.3.0/24	Serial 0
156.26.56.0/24	Serial 1
156.26.57.0/24	Serial 1
156.26.58.0/24	Serial 1
156.26.59.0/24	Serial 1

The network in Figure 3-22 is similar to the network that was developed in Chapter 1 for the statewide delivery of mail. Router G is equivalent to the core post office that routed mail between states, and between cities in a state. Routers E and F are equivalent to the distribution post offices that routed mail between the access post offices and the state post office. Routers A, B, C, and D are equivalent to the access post offices that routed mail between streets (networks) and the distribution post offices. For the statewide postal network, the core post office did not need to know about every street. It was sufficient to route mail based on the city name. For routing between states, the core post office did not need to know the route to every city and every street in another state. It was sufficient to route interstate mail based on the state name alone. This process of information hiding, or route reduction, was called *route summarization* or aggregation. It would be nice if IP routes could be aggregated to reduce the size of the routing tables.

Routes are summarized, or aggregated, by reversing the subnetting process. For example, in Figure 3-21, the 156.26.63.240/28 network was subnetted into 4 /30 networks:

156.26.63.240/30

156.26.63.244/30

156.26.63.248/30

156.26.63.252/30

A router can have these four specific routes in the routing table. Or, a router can have one route, or ***IP prefix***, that summarizes these four specific networks. The summary prefix 156.26.63.240/28 contains every possible subnet of 156.26.63.240/28 in the same way that a state contains every possible city and street name contained within that state. The state name summarizes all the city and street names into one prefix. A summary address allowed the core post office to maintain one route to another state and not a route for every possible destination in the other state.

A summary prefix should only summarize those subnets that are actually being used. The prefix 156.26.0.0/16 summarizes all the subnets of the Class B address

space 156.26.0.0. So the prefix 156.26.0.0/16 does summarize the four specific /30 subnets of 156.26.63.240/28, but it also summarizes all other subnets of 156.26.0.0/16. This summary tells a router that all subnets of 156.26.0.0/16 are reachable even though many of the subnets might not be in use.

For the network in Figure 3-22 and subnets in Table 3-13, the subnets can be summarized into one route advertisement.

For Router G, 156.26.0.0/24 through 156.26.3.0/24 can be reached through interface serial 0. If you look at the bit patterns of these four subnets, you can determine the subnet mask to use to summarize these routes. It is sufficient, in this case, to examine only the third byte of the subnets:

0 = 0 0 0 0 0 0 0 0
1 = 0 0 0 0 0 0 0 1
2 = 0 0 0 0 0 0 1 0
3 = 0 0 0 0 0 0 1 1

The subnet mask that needs to be used should include only those bits that do not change. For these four routes, the upper 6 bits do not change. These 6 bits need to be included in the summary subnet mask. The value of the mask for the third byte is 1 1 1 1 1 1 0 0 = 252, so the required subnet mask is 255.255.252.0.

Applying the same process to the subnets 156.26.56.0/24 through 156.26.59.0/24, the values of the third byte are

56 = 0 0 1 1 1 0 0 0
57 = 0 0 1 1 1 0 0 1
58 = 0 0 1 1 1 0 1 0
59 = 0 0 1 1 1 0 1 1

As with the previous example, the upper 6 bits need to be included in the subnet mask and the required mask is again 255.255.252.0. The new routing table for Router G is listed in Table 3-14.

Table 3-14 IP Routing Table for Router G Using Summary Prefixes

Route	Output Interface
156.26.63.240/30	Directly connected, Serial 0
156.26.63.244/30	Directly connected, Serial 1
156.26.63.0/28	Serial 0
156.26.63.16/28	Serial 1
156.26.0.0/22	Serial 0
156.26.56.0/22	Serial 1

The routing table on Router G has been reduced from 12 to 6 routes, a significant reduction. Notice that the two new summary prefixes have a 22-bit subnet mask instead of a 24-bit subnet mask. To see how this works, assume Router G receives a packet for the host at IP address 156.26.2.37. There is no subnet mask information in a destination IP address. The router will find the best match for this route from the routing table. An address with /32 is a host address:

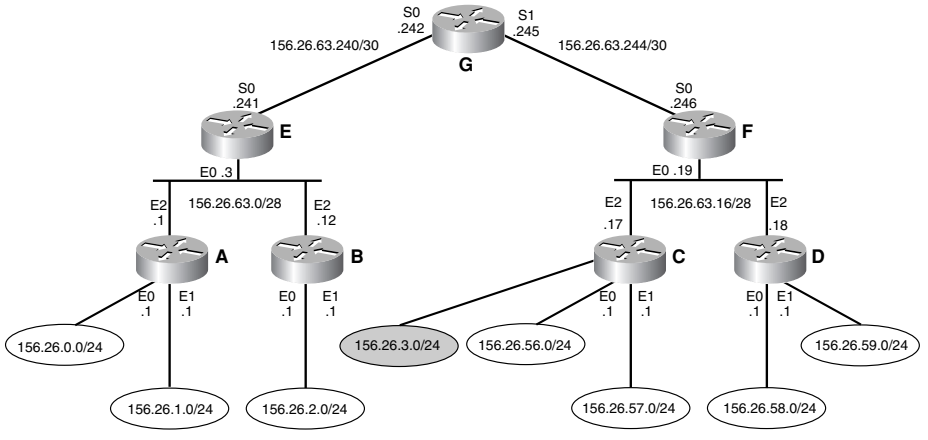
156.26.2.37/32 = 10011100 00011010 00000010 00100101

156.26.0.0/22= 10011100 00011010 00000000 00000000

There is a 22-bit match between the host address and the prefix 156.25.0.0/22, so this packet will be forward using interface serial 0.

What if subnet 156.26.3.0/24 was moved to Router C? (See Figure 3-24.)

Figure 3-24 Summary and Specific IP Prefixes



Can we still summarize the networks attached to Routers A and B? Yes. The summary prefix 156.26.0.0/22 contains 156.26.0.0/24 through 156.26.3.0/24, so Router G thinks it can reach the 156.26.3.0/24 network through Router E. You can keep this summary prefix as long as a more specific prefix for network 152.26.3.0/24 is added to the routing table on Router G. (See Table 3-15.)

Table 3-15 IP Routing Table for Router G Using Summary Prefixes and a More Specific Prefix

Route	Output Interface
156.26.63.240/30	Directly connected, Serial 0
156.26.63.244/30	Directly connected, Serial 1
156.26.63.0/28	Serial 0
156.26.63.16/28	Serial 1
156.26.0.0/22	Serial 0
156.26.56.0/22	Serial 1
156.26.3.0/24	Serial 1

Router G now has two routes to subnet 156.26.3.0/24. Which one will it use? Assume Router G receives a packet for host 156.26.3.12/32. Router G will compare this route with the entries in the routing table and there are two that match.

This matches 22 bits in the host address:

156.26.0.0/22 = 10011100 00011010 00000000 00000000

156.26.3.12/32 = 10011100 00011010 00000011 00001100

This matches 24 bits and the longest match wins. Router G will forward the packet to Router F:

156.26.3.0/24 = 10011100 00011010 00000011 00000000

156.26.3.12/32 = 10011100 00011010 00000011 00001100

Try reinforcing the key points with the following questions:

1. How many subnets of the Class C address 197.45.120.0/24 are there that can support at least 12 hosts?

   Answer: Four bits are required for 12 hosts ($2^4 - 2 = 14$). This is a Class C address, so there are 4 bits left for the network. Therefore, there are 16 subnets that can support at least 12 hosts.

2. What are the network numbers for the subnets in the previous question?

   Answer: The first 4 bits of the last byte are included in the network number.

   0 0 0 0 0 0 0 0 = 0 197.45.120.0

   0 0 0 1 0 0 0 0 = 16 197.45.120.16

   0 0 1 0 0 0 0 0 = 32 197.45.120.32

   0 0 1 1 0 0 0 0 = 48 197.45.120.48

   0 1 0 0 0 0 0 0 = 64 197.45.120.64

   0 1 0 1 0 0 0 0 = 80 197.45.120.80

   0 1 1 0 0 0 0 0 = 96 197.45.120.96

0 1 1 1 0 0 0 0 = 112 197.45.120.112

1 0 0 0 0 0 0 0 = 128 197.45.120.128

1 0 0 1 0 0 0 0 = 144 197.45.120.144

1 0 1 0 0 0 0 0 = 160 197.45.120.160

1 0 1 1 0 0 0 0 = 176 197.45.120.176

1 1 0 0 0 0 0 0 = 192 197.45.120.192

1 1 0 1 0 0 0 0 = 208 197.45.120.208

1 1 1 0 0 0 0 0 = 224 197.45.120.224

1 1 1 1 0 0 0 0 = 240 197.45.120.240

3. Summarize the 16 networks from the previous example into two equal size prefixes.

Answer: Examine the bit patterns of the fourth byte of the first 8 subnets.

0 0 0 0 0 0 0 0 = 0 197.45.120.0

0 0 0 1 0 0 0 0 = 16 197.45.120.16

0 0 1 0 0 0 0 0 = 32 197.45.120.32

0 0 1 1 0 0 0 0 = 48 197.45.120.48

0 1 0 0 0 0 0 0 = 64 197.45.120.64

0 1 0 1 0 0 0 0 = 80 197.45.120.80

0 1 1 0 0 0 0 0 = 96 197.45.120.96

0 1 1 1 0 0 0 0 = 112 197.45.120.112

The only bit that is constant is the first bit, so a 25-bit mask is needed. The summary for the first eight subnets is

197.45.120.0/25

The only bit that is constant for the second set of eight subnets is again the first bit and it is always 1. The summary for the second set of eight subnets is

197.45.120.128/25.

Supernets

When more bits are used than the natural mask length for the network portion of a Class A, B, or C address, this process was called ***subnetting***. The natural mask for a Class A address is 8 bits. If more than 8 bits are used for the network portion of the IP address, we say that the Class A address has been subnetted.

You can also use fewer bits than the natural mask for the network portion. This process is called ***supernetting***. For example, assume your company owns the following four Class C addresses:

200.10.4.0/24

200.10.5.0/24

200.10.6.0/24

200.10.7.0/24

You can aggregate the addresses using a 22-bit mask, which is 2 bits less than the natural 24-bit mask. The process is the same as subnetting, but the term that is used depends on whether more or fewer bits than the natural mask are being used. The supernet for these networks is 200.10.4.0/22.

IP Version 4 and IP Version 6

IP version 4 (IPv4) has not changed much since it was defined in 1981. For the last two decades, IPv4 has proven to be a robust and scalable protocol for Internet routing. Unfortunately, the designers of IPv4 did not anticipate the explosive growth of the Internet, or the need for more IP addresses than version 4 could supply.

IPv4 uses 32-bit IP address, and with 32 bits the maximum number of IP addresses is 2^{32}—or 4,294,967,296. This provides a little more than four billion IPv4 addresses (in theory). The number of IPv4 available addresses is actually less than the theoretical maximum number. The reason the actual number of usable IP addresses is less than the maximum is because the broadcast and "this" network addresses cannot be assigned to hosts. A usable IPv4 address is one that can be assigned to a host, implying a unicast IP address. The only unicast IP addresses available are Class A, B, and C addresses. How many unicast IPv4 addresses are there? There are $2^7 - 3$, or 126, possible Class A networks with numbers ranging from 1 to 126. (0 and 127 are not used, and 10 is the Class A private address space.) Each Class A network can have $2^{24} - 2$, or 16,777,216 hosts. (A host address of all 0s signifies the network address, and a host address of all 1s signifies the broadcast address.) The number of Class A hosts is 126 * 16,777,216 or 2,113,929,216. There are $2^{14} - 1$, or 16,383 Class B networks. (172.16.0.0 is the private Class B address space.) Each Class B network can have $2^{16} - 2$, or 65,534 hosts. The number of Class B hosts is 16,383 * 65,534, or 1,073,643,522. There are $2^{21} - 1$, or 2,097,151 possible Class C networks. (192.168.0.0 is the private Class C address space.) Each Class C network can have $2^8 - 2$, or 254, hosts. The number of Class C hosts is 2,097,151 * 254 or 532,676,354. The total number of IPv4 unicast addresses is 3,720,249,092. A Class A, B, or C address identifies one specific host, and these addresses are called unicast addresses. The private addresses can be used in a network, but cannot be advertised on the Internet. This allows many networks to use the same private address as long as the hosts using these addresses do not need to be connected to the Internet.

The actual number of usable IPv4 unicast addresses is less than four billion. But there are usable addresses that will never be used. When IPv4 addresses were first allocated to government agencies, universities, and businesses, the addresses were allocated as classful addresses. If a university received a Class A address, the university had 16,777,216 host addresses that could be used. I cannot imagine any university, business, or government agency using every possible address assigned to them. It is difficult to determine how many IPv4 unicast addresses will never be used, but I'm sure it is more than 1. So the actual number of usable IPv4 addresses is less than 3.7 billion.

At first glance, even 3.7 billion addresses seems like enough. One reason it is not enough is the majority of the IPv4 address space has been allocated to countries that were early implementers of the Internet. The United States and Europe own the majority of the IP address space. Emerging countries like China need more IP addresses than what is available, driving the need for a larger address space.

Also, in the twenty-first century, devices other than computers need an Internet address. Cell phones, PDAs, vehicles, and appliances are all becoming part of the Internet. There simply are not enough IPv4 addresses to go around. So the big question is, how much is enough?

The current world population is more than six billion people, so there are more people than there are IPv4 addresses. If you assume everyone will eventually need at least one IP address, it is easy to see IPv4 does not have enough addresses. For every bit added to an IP address, the size of the address space doubles. A 33-bit IP address has around 8.5 billion addresses. A 34-bit IP address has about 17 billion possible addresses, and so on. IP version 6 (IPv6) uses 128 bits and it is interesting to investigate if 128 bits satisfies the need for more IP addresses.

Using 128 bits gives a theoretical address space of $3.4 * 10^{38}$ addresses. This is 3.4 followed by 38 zeros, or 3,400,000,000,000,000,000,000,000,000,000,000,000. Wow! That looks like a BIG number. But how big is it? To put this number in perspective, we need something to compare it to.

There are approximately 100 billion nerve cells in your brain or $1 * 10^{11}$. If you divide the number of possible IPv6 addresses by the number of nerve cells in your brain you get

$3.4 * 10^{38} / 1.0 * 10^{11} = 3.4 * 10^{27}$ IPv6 address for every nerve cell in your brain.

There are approximately $7 * 10^{27}$ atoms in your body. $3.4 * 10^{38} / 7.0 * 10^{27} = 4.86 * 10^{10}$ IPv6 address for every atom in your body. This is more than 48 billion! Of course, you have to share these addresses with 6 billion plus people, so every atom in your body can only have 8 billion IPv6 addresses. By now you should be convinced that the number of possible addresses using 128 bits should last us for quite awhile.

IPv6 Address Format

IPv4 addresses are typically represented using the dotted decimal notation. For example, the 32-bit IPv4 address $10011100000110100010000000000001_2$ can be represented as the dotted decimal number 156.26.21.1.

IPv6 uses eight 16-bit hexadecimal numbers (8 * 16 = 128 bits) separated by a colon to represent a 128-bit IPv6 address using the following rules:

- Leading zeros in each 16-bit field are optional.

  Example: The IPv6 address

  1A23:120B:0000:0000:0000:7634:AD01:004D can be represented by

  1A23:120B:0:0:0:7634:AD01:004D

- Successive fields with the value 0 can be represented by a pair of colons (::).

  Example: The IPv6 address

  1A23:120B:0000:0000:0000:7634:AD01:004D can be represented by

  1A23:120B::7634:AD01:4D

  The double colon :: represents the number of 0s needed to produce eight 16-bit hexadecimal numbers.

- The double colon :: can be used only once to represent an IPv6 address.

  Example: The IPv6 address

  1A23:120B:0000:0000:1234:0000:0000:4D can be represented by

  1A23:120B::1234:0:0:004D or

  1A23:120B:0:0:124::4D, but not by

  1A23:120B::1234::4D because there is no way to determine how many zeros each :: represents.

IPv6 Address Types

IPv4 uses two types of addresses: unicast and multicast. Unicast addresses are the Class A, B, and C addresses and are used to identify a single host on the Internet. Multicast addresses are used to identify multiple hosts for the delivery of multicast traffic (discussed in more detail in Chapter 9, "Multicast—What the Post Office Can't Do"). IPv6 has three major address types: unicast, multicast, and anycast.

IPv6 unicast addresses are divided into five groups:

- Global unicast addresses—Equivalent in function to an IPv4 unicast address using 64 bits for the network ID and 64 bits for the host ID.
- Site-local unicast addresses—Equivalent to the IPv4 private addresses such as 10.0.0.0 and 172.16.0.0.
- Link-local unicast addresses—An IPv6 address that is automatically configured on an interface allowing hosts on the same subnet to communicate with each other without the need for a router.
- IPv4-compatible IPv6 addresses—Used to transport IPv6 messages over an IPv4 network. An IPv4 address is placed in the low-order 32 bits of an IPv6 address. For example, the IPv4-compatible IPv6 address for the IPv4 address 156.26.32.1 is

  0:0:0:0:0:0:156.26.32.1 = ::156.26.32.1 = ::9C1A2001

- IPv4-mapped IPv6 addresses—Similar to an IPv4-compatible address, and used to represent an IPv4 interface as an IPv6 interface using 16 ones before the IPv4 address. For example, the IPv4-mapped IPv6 address for the IPv4 address 156.26.32.1 is

  0:0:0:0:0:FF:156.26.32.1 = ::FFFF:9C1A:2001

IPv6 multicast addresses serve the same function as IPv4 mulitcast addresses (again, more on this in Chapter 9). The anycast address type is a unicast address assigned to a set of interfaces, and a packet is sent to the nearest interface.

IPv6 provides enough addresses to last for a very long time. Eventually, the Internet will move to the use of IPv6; and, for a time, IPv4 and IPv6 will both be used. Routing protocols must be able to handle both address formats. For an in-depth discussion of IPv6, refer to the references at the end of the chapter.

Summary

The delivery of an IP packet is similar in concept to the delivery of a letter in the postal system. The destination address on a letter consists of two parts used to deliver the letter to the final destination. The state, city, and street name are used to route the letter to the proper street. This is analogous to the network portion of an IP address used to route an IP packet to the proper network. The street number is used to determine the proper house while the host portion of an IP address is used to determine the destination host. The name on the letter determines the recipient of the letter, and on the Internet, the UDP or TCP port number serves a similar function. The port number is used to deliver the data to the proper application.

Although the concepts of mail and IP packet delivery are similar, the addressing details of IP are much more complicated and difficult to master. If you are planning on becoming a network professional, it is imperative that you master the details of IP addressing, subnetting, and summarization.

Chapter Review Questions

You can find the answers to these questions in Appendix A.

1. What is the broadcast address for network 142.16.72.0/23?
2. Subnet 198.4.81.0/24 into the maximum number of networks that can support 28 hosts each.
3. What is the broadcast address for network 198.4.81.96/27?

4. What is the prefix and subnet mask that summarizes the following networks:

   162.8.0.0/22

   162.8.4.0/22

   162.8.8.0/22

   162.8.12.0/22

5. Using the following routing table, determine the best route to reach the host at address 132.19.237.5.

   Network Output Interface

   132.0.0.0/8 Serial 0

   132.16.0.0/11 Ethernet 1

   132.16.233.0/22 Ethernet 2

6. What is the range of host addresses for network 172.16.53.96/27?

7. How many two-host subnets can be made from a /24 network?

8. What is the full IPv6 address represented by FF02::130F:5?

References

Two of these references are RFCs or Requests for Comments. RFC is a misleading name because RFCs are Internet standards that specify every protocol used on the Internet. There are thousands of RFCs, and a searchable index of RFCs can be found at http://www.ietf.org/rfc.html. RFCs are technical documents that are sometimes difficult to follow, so you might want to read the books listed here first.

- Comer, Douglas E. 2000. *Internetworking with TCP/IP, Volume 1: Principles, Protocols, and Architecture*, Fourth Edition. Upper Saddle River, NJ: Prentice Hall.
- Hinden, R. and S. Deering. July 1998. IP Version 6 Addressing Architecture. RFC 2373.

- Postel, J. September 1981, Internet Protocol. RFC 0791.
- Stevens, W. Richard. 1993. *TCP/IP Illustrated*, Volume 1. Upper Saddle River, NJ: Addison-Wesley.