ADDRESSING

Internet Address

The identifier used in the network layer of the Internet model to identify each device connected to the Internet is called the **Internet address** or **IP address.** An IP address, in the current version of the protocol, is a 32-bit binary address that *uniquely* and *universally* defines the connection of a host or a router to the Internet.

An IP address is a 32-bit address.

IP addresses are unique. They are unique in the sense that each address defines one, and only one, connection to the Internet. Two devices on the Internet can never have the same address at the same time. However, if a device has two connections to the Internet, via two networks, it has two IP addresses.

The IP addresses are universal in the sense that the addressing system must be accepted by any host that wants to be connected to the Internet.

Example 2

Change the following IP addresses from dotted-decimal notation to binary notation.

a. 111.56.45.78

b. 75.45.34.78

Solution

We replace each decimal number with its binary equivalent (see Appendix B):

a. 01101111 00111000 00101101 01001110

b. 01001011 00101101 00100010 01001110

Classful Addressing

IP addresses, when started a few decades ago, used the concept of classes. This architecture is called **classful addressing.** In the mid-1990s, a new architecture, called **classless addressing,** was introduced which will eventually supersede the original architecture. However, most of the Internet is still using classful addressing, and the migration is slow. We first discuss classful addressing.

In classful addressing, the IP address space is divided into five classes: **classes A, B, C, D,** and **E.** Each class occupies some part of the whole **address space.**

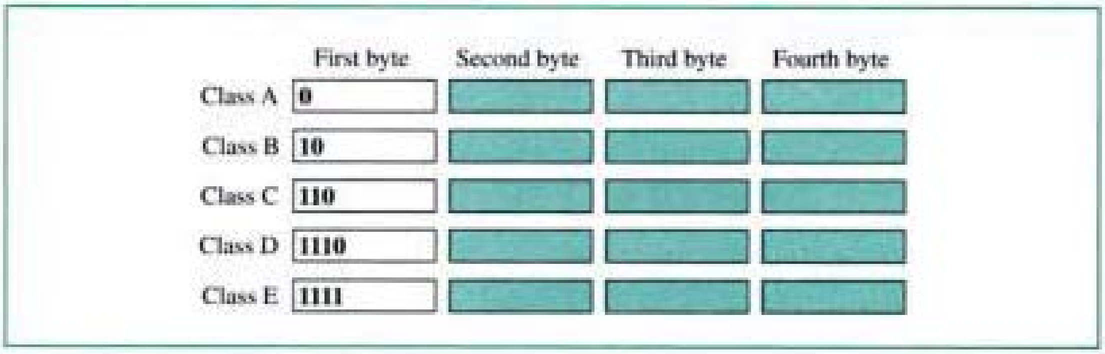
In classful addressing, the address space is divided into five classes: A, B, C, D, and E.

We can find the class of an address when the address is given in binary notation or dotted-decimal notation.

Finding the Class in Binary Notation

If the address is given in binary notation, the first few bits can immediately tell us the class of the address, as shown in Figure 19.10.

Figure 19.10 *Finding the class in binary notation*



One can follow the procedure shown in Figure 19.11 to systematically check the bits and find the class.

Solution

a. The first byte is 227 (between 224 and 239); the class is D.
b. The first byte is 252 (between 240 and 255); the class is E.
c. The first byte is 134 (between 128 and 191); the class is B.

Unicast, Multicast, and Reserved Addresses

Addresses in classes A, B, and C are for unicast communication, from one source to one destination. A host needs to have at least one **unicast address** to be able to send or receive packets.

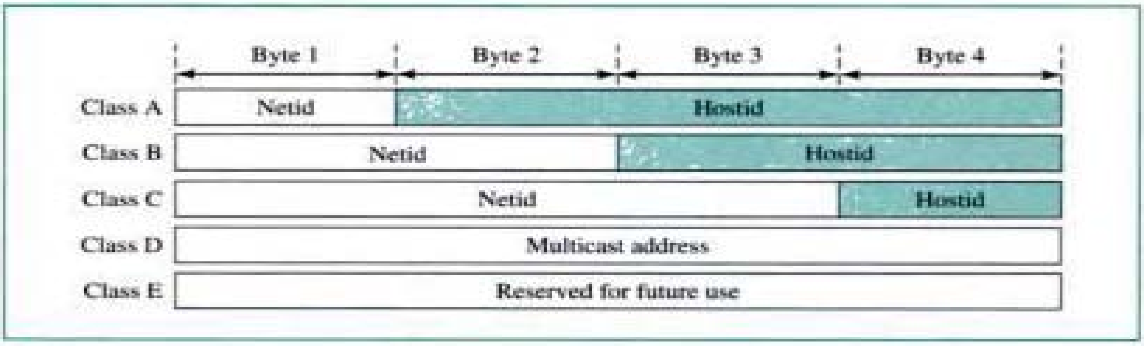
Addresses in class D are for multicast communication, from one source to a group of destinations. If a host belongs to a group or groups, it may have one or more multicast addresses. A **multicast address** can be used only as a destination address, but never as a source address.

Addresses in class E are reserved. The original idea was to use them for special purposes. They have been used only in a few cases.

Netid and Hostid

In classful addressing, an IP address in classes A, B, and C is divided into **netid** and **hostid.** These parts are of varying lengths, depending on the class of the address. Figure 19.13 shows the netid and hostid bytes. Note that classes D and E are not divided into netid and hostid for reasons that we will discuss later.

Figure 19.13 *Netid and hostid*



In class A, one byte defines the netid and three bytes define the hostid. In class B, two bytes define the netid and two bytes define the hostid. In class C, three bytes define the netid and one byte defines the hostid.

Classes and Blocks

One problem with classful addressing is that each class is divided into a fixed number of blocks with each block having a fixed size. Let us look at each class.

Class A Class A is divided into 128 blocks with each block having a different netid. The first block covers addresses from **0**.0.0.0 to **0**.255.255.255 (netid **0**). The second block covers addresses from **1**.0.0.0 to **1**.255.255.255 (netid **1**). The last block covers addresses from **127**.0.0.0 to **127**.255.255.255 (netid **127**). Note that for each block of addresses the first byte (netid) is the same, but the other 3 bytes (hostid) can take any value in the given range. Figure 19.14 shows the blocks in class A.

Figure 19.14 *Blocks in class A*

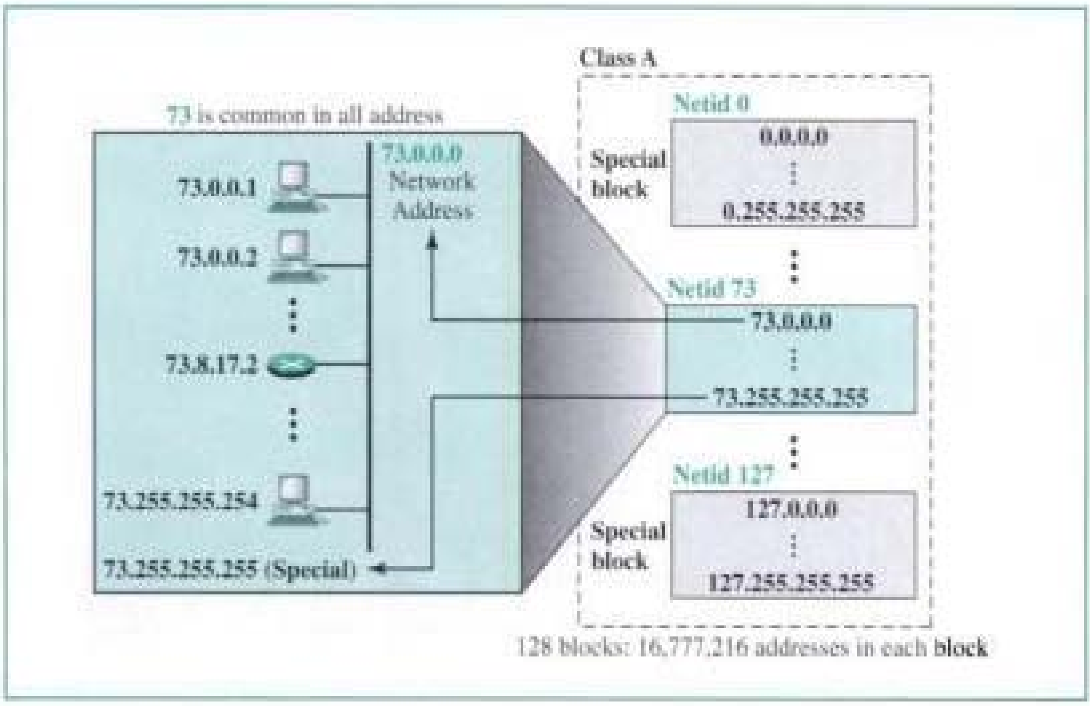


Figure 19.14 also shows how an organization that is granted a block with netid 73 uses its addresses. The first address in the block is used to identify the organization to the rest of the Internet. This address is called the **network address;** it defines the network of the organization, not individual hosts. The organization is not allowed to use the last address; it is reserved for a special purpose, as we will see shortly.

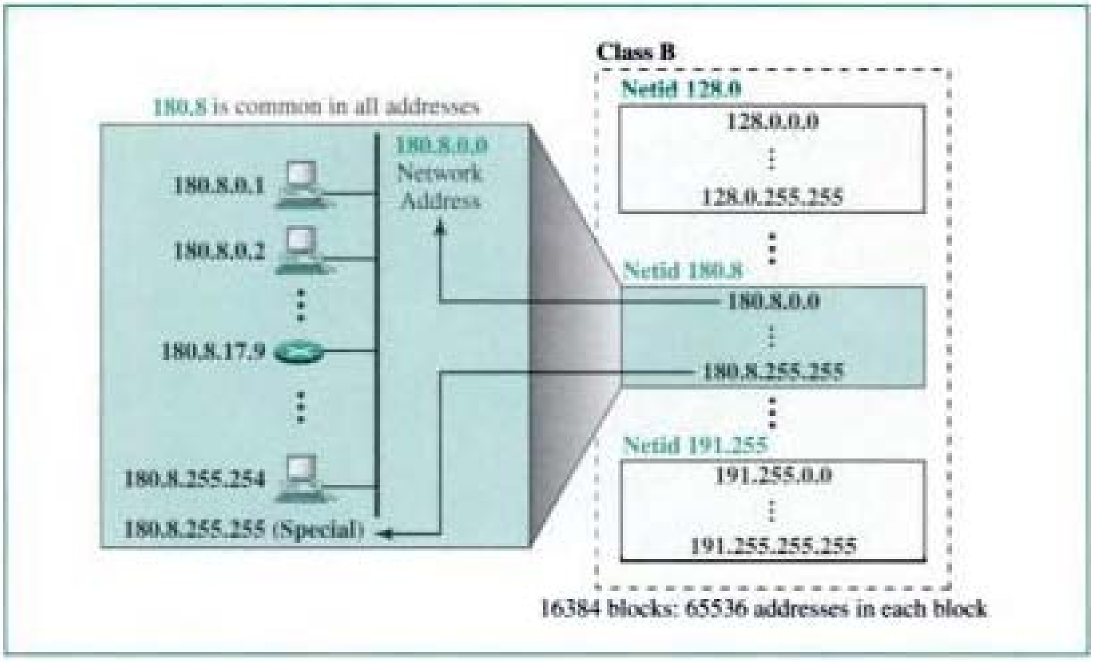
Class A addresses were designed for large organizations with a large number of hosts or routers attached to their network. However, the number of addresses in each block, 16,777,216, is probably larger than the needs of almost all organizations. Many addresses are wasted in this class.

Millions of class A addresses are wasted.

Class B Class B is divided into 16,384 blocks with each block having a different netid. Sixteen blocks are reserved for private addresses, leaving 16,368 blocks for assignment to organizations. The first block covers addresses from **128.0**.0.0 to **128.0**.255.255 (netid **128.0**). The last block covers addresses from **191.255**.0.0 to **191.255**.255.255 (netid **191.255**). Note that for each block of addresses the first 2 bytes (netid) is the same, but the other 2 bytes (hostid) can take any value in the given range.

There are 16,368 blocks that can be assigned. This means that the total number of organizations that can have a class B address is 16,368. However, since each block in

Figure 19.15 *Blocks in class B*



this class contains 65,536 addresses, the organization should be large enough to use all these addresses. Figure 19.15 shows the blocks in class B.

Class B addresses were designed for midsize organizations that may have tens of thousands of hosts or routers attached to their networks. However, the number of addresses in each block, 65,536, is larger than the needs of most midsize organizations. Many addresses are also wasted in this class.

Many class B addresses are wasted.

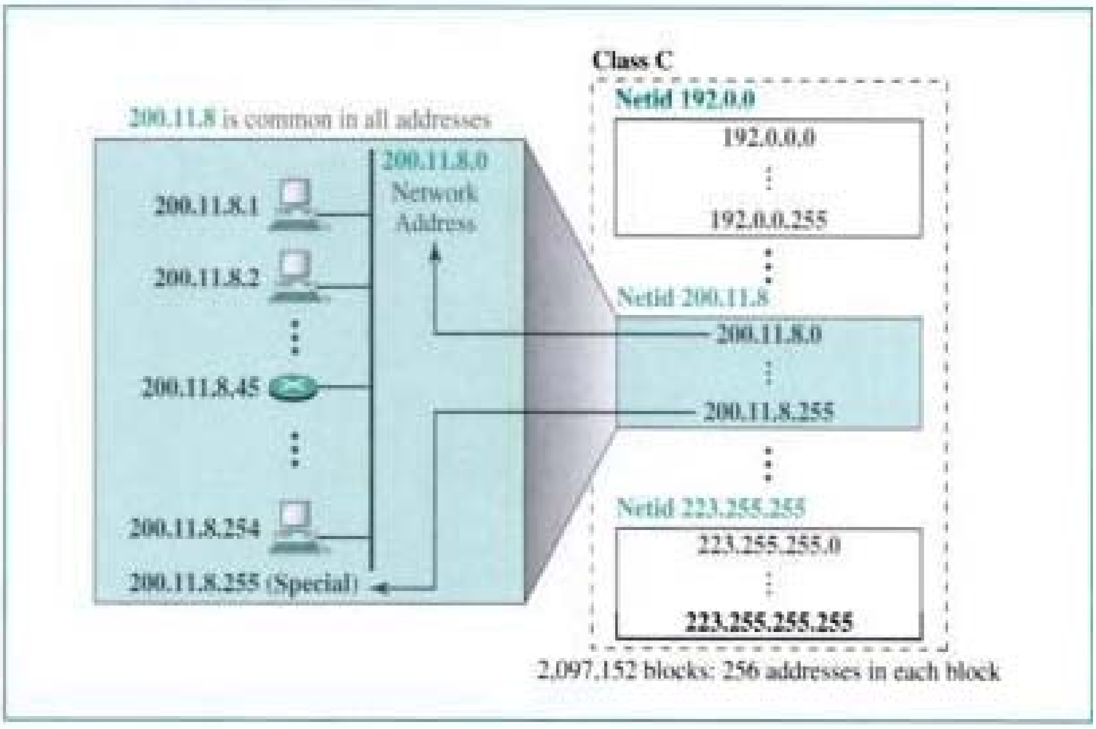
Class C Class C is divided into 2,097,152 blocks with each block having a different netid. Two hundred fifty-six blocks are used for private addresses, leaving 2,096,896 blocks for assignment to organizations. The first block covers addresses from 192.0.0.0 to 192.0.0.255 (netid 192.0.0). The last block covers addresses from 223.255.255.0. to 223.255.255.255 (netid 223.255.255). Note that for each block of addresses the first 3 bytes (netid) are the same, but the remaining byte (hostid) can take any value in the given range.

There are 2,096,902 blocks that can be assigned. This means that the total number of organizations that can have a class C address is 2,096,902. However, each block in this class contains 256 addresses, which means the organization should be small enough to need less than 256 addresses. Figure 19.16 shows the blocks in class C.

Class C addresses were designed for small organizations with a small number of hosts or routers attached to their networks. The number of addresses in each block is so limited that most organizations do not want a block in this class.

The number of addresses in class C is smaller than the needs of most organizations.

Figure 19.16 *Blocks in class C*



Class D There is just one block of class D addresses. It is designed for multicasting.

Class E There is just one block of class E addresses. It was designed for use as reserved addresses.

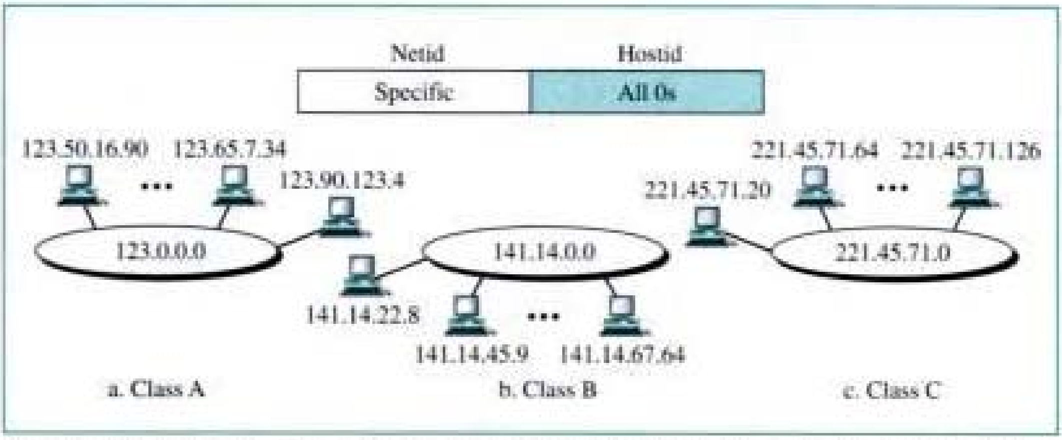
Network Address

The network address is an address that defines the network itself; it cannot be assigned to a host. Figure 19.17 shows three examples of network addresses, one for each class.

Network Address

The network address is an address that defines the network itself; it cannot be assigned to a host. Figure 19.17 shows three examples of network addresses, one for each class.

Figure 19.17 *Network address*



Network addresses play a very important role in classful addressing. A network address has several properties:

1. All hostid bytes are 0s.
2. The network address defines the network to the rest of the Internet. Later, we learn that routers can route a packet based on the network address.
3. The network address is the first address in the block.
4. Given the network address, we can find the class of the address.

In classful addressing, the network address is the one that is assigned to the organization.

Example 5

Given the address 23.56.7.91, find the network address.

Solution

The class is A. Only the first byte defines the netid. We can find the network address by replacing the hostid bytes (56.7.91) with 0s. Therefore, the network address is 23.0.0.0.

Example 6

Given the address 132.6.17.85, find the network address.

Solution

The class is B. The first 2 bytes defines the netid. We can find the network address by replacing the hostid bytes (17.85) with 0s. Therefore, the network address is 132.6.0.0.

Example 7

Given the network address 17.0.0.0, find the class.

Solution

The class is A because the netid is only 1 byte.

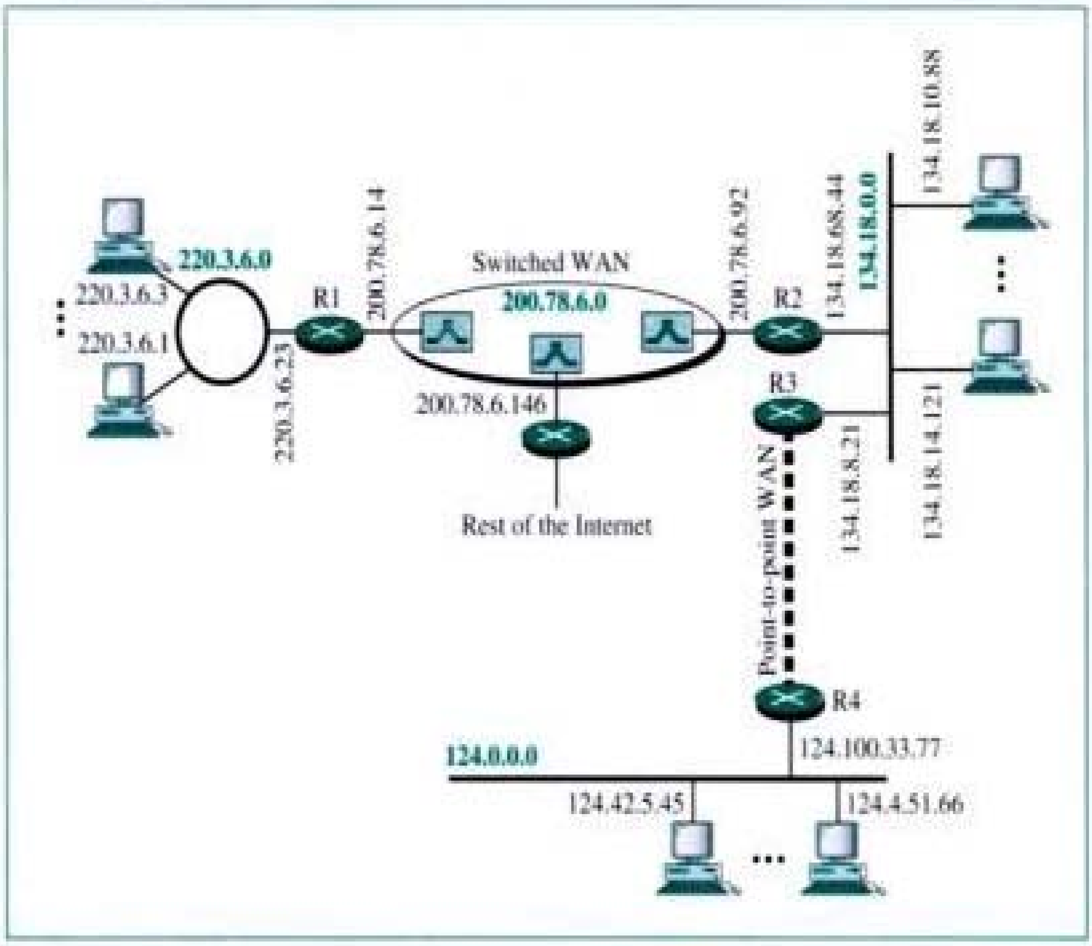
A network address is different from a netid. A network address has both netid and hostid, with 0s for the hostid.

A Sample Internet with Classful Addresses

Figure 19.18 shows a part of an internet with five networks.

1. A Token Ring LAN with network address 220.3.6.0 (class C).
2. An Ethernet LAN with network address 134.18.0.0 (class B).
3. An Ethernet LAN with network address 124.0.0.0 (class A).
4. A point-to-point WAN (broken line). This network (a T-1 line, for example) just connects two routers; there are no hosts. In this case, to save addresses, no network address is assigned to this type of WAN. A switched WAN (such as Frame Relay or ATM) can be connected to many routers. We have shown three. One router connects the WAN to the Token Ring network. One connects the WAN to one of the Ethernet networks, and one router connects the WAN to the rest of the Internet.

Figure 19.18 *Sample internet*



Calculation of IP address.

First Address The first address in the block can be found by setting the 32 - n rightmost bits in the binary notation of the address to Os. The first address in the block can be found by setting the rightmost 32 - *n* bits to 0s.

***Example*19.6**

A block of addresses is granted to a small organization. We know that one of the addresses is 205.16.37.39/28. What is the first address in the block?

Solution

The binary representation of the given address is 11001101 00010000 00100101 00100 I 11. If we set 32 - 28 rightmost bits to 0, we get 11001101 000100000100101 0010000 or 205.16.37.32.

This is actually the block shown in Figure 19.3.

Last Address The last address in the block can be found by setting the 32 - *n* rightmost bits in the binary notation of the address to Is.

The last address in the block can be found by setting the rightmost 32 - *n* bits to Is.

Example 19.7

Find the last address for the block in Example 19.6.

Solution

The binary representation of the given address is 11001101 000100000010010100100111. If we set 32 - 28 rightmost bits to 1, we get 11001101 00010000 001001010010 1111 or 205.16.37.47.

This is actually the block shown in Figure 19.3.

Number of Addresses The number of addresses in the block is the difference between the last and first address. It can easily be found using the formula *232- n.*

The number of addresses in the block can be found by using the formula *232- n*•

Example 19.8

Find the number of addresses in Example 19.6.

Solution

The value of *n* is 28, which means that number of addresses is 232- 28 or 16.

Example 19.9

Another way to find the first address, the last address, and the number of addresses is to represent the mask as a 32-bit binary (or 8-digit hexadecimal) number. This is particularly useful when we

are writing a program to find these pieces of information. In Example 19.5 the /28 can be represented as 11111111 11111111 11111111 11110000 (twenty-eight Is and four Os). Find
a. The first address
h. The last address
c. The number of addresses
Solution
a. The first address can be found by ANDing the given addresses with the mask. ANDing here is
done bit by bit. The result of ANDing 2 bits is 1 if both bits are Is; the result is 0 otherwise.
Address:
Mask:
First address:
11001101 00010000 00100101 00100111
11111111 11111111 11111111 11110000
11001101 00010000 00100101 00100000
b. The last address can be found by ORing the given addresses with the complement of the mask. ORing here is done bit by bit. The result of ORing 2 bits is 0 if both bits are Os; the result is 1 otherwise. The complement of a number is found by changing each 1 to 0 and each 0 to 1.
Address:
Mask complement:
Last address:
11001101 00010000 00100101 00100111
00000000 00000000 00000000 00001111
11001101 00010000 *00100101 00101111*
c. The number of addresses can be found by complementing the mask, interpreting it as a decimal number, and adding 1 to it.
Mask complement:
Number of addresses:
000000000 00000000 00000000 00001111
15 + 1 =16

Network Address Translation (NAT)

The number of home users and small businesses that want to use the Internet is ever increasing. In the beginning, a user was connected to the Internet with a dial-up line, which means that she was connected for a specific period of time. An ISP with a block of addresses could dynamically assign an address to this user. An address was given to a user when it was needed. But the situation is different today. Home users and small businesses can be connected by an ADSL line or cable

modem. In addition, many are not happy with one address; many have created small networks with several hosts and need an IP address for each host. With the shortage of addresses, this is a serious problem.

A quick solution to this problem is called network address translation (NAT). NAT enables a user to have a large set of addresses internally and one address, or a small set of addresses, externally. The traffic inside can use the large set; the traffic outside, the small set.

To separate the addresses used inside the home or business and the ones used for the Internet, the Internet authorities have reserved three sets of addresses as private addresses, shown in Table 19.3.

Table 19.3 *Addresses for private networks*

Range			*Total*
10.0.0.0	to	10.255.255.255	2^{24}
172.16.0.0	to	172.31.255.255	2^{20}
192.168.0.0	to	192.168.255.255	2^{16}