5.1 Introduction

5.2 Classful Addressing

5.3 Classless Addressing

5.4 Special Addresses

5.5 NAT

5-1 INTRODUCTION

The identifier used in the IP layer of the TCP/IP protocol suite to identify each device connected to the Internet is called the Internet address or IP address. An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a host or a router to the Internet; an IP address is the address of the interface.

Topics Discussed in the Section

- ✓ **Notation**
- ✓ **Range of Addresses**
- ✓ **Operations**

Note

An IPv4 address is 32 bits long.

Note

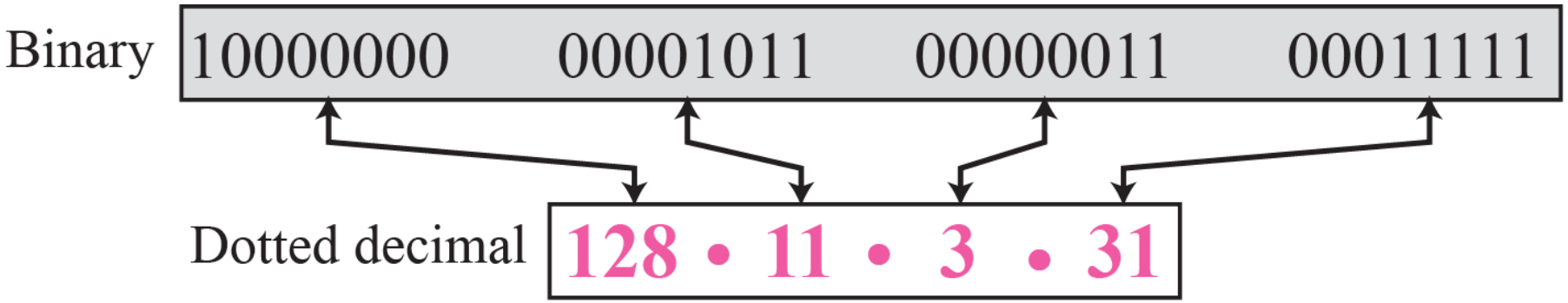
The IPv4 addresses are unique and universal.

Note

The address space of IPv4 is 2^{32} or 4,294,967,296.

4 billion 294 million 967 thousand 296

Figure 5.1 *Dotted-decimal notation*



Example 5.1

Change the following IPv4 addresses from binary notation to dotted-decimal notation.

a. 10000001 00001011 00001011 11101111
b. 11000001 10000011 00011011 11111111
c. 11100111 11011011 10001011 01101111
d. 11111001 10011011 11111011 00001111

Solution

We replace each group of 8 bits with its equivalent decimal number and add dots for separation:

a. 129.11.11.239
b. 193.131.27.255
c. 231.219.139.111
d. 249.155.251.15

Example 5.2

Change the following IPv4 addresses from dotted-decimal notation to binary notation.

a. 111.56.45.78
b. 221.34.7.82
c. *241.8.56.12*
d. *75.45.34.78*

Solution

We replace each decimal number with its binary equivalent:

a. 01101111 00111000 00101101 01001110
b. 11011101 00100010 00000111 01010010
c. 11110001 00001000 00111000 00001100
d. 01001011 00101101 00100010 01001110

Example 5.3

Find the error, if any, in the following IPv4 addresses:

a. 111.56.045.78
b. 221.34.7.8.20
c. 75.45.301.14
d. 11100010.23.14.67

Solution

a. There should be no leading zeroes (045).
b. We may not have more than 4 bytes in an IPv4 address.
c. Each byte should be less than or equal to 255.
d. A mixture of binary notation and dotted-decimal notation.

Example 5.4

Change the following IPv4 addresses from binary notation to hexadecimal notation.

a. 10000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

Solution

We replace each group of 4 bits with its hexadecimal equivalent. Note that 0X (or 0x) is added at the beginning or the subscript 16 at the end.

a. 0X810B0BEF or $810B0BEF_{16}$

b. 0XC1831BFF or $C1831BFF_{16}$

Example 5.5

Find the number of addresses in a range if the first address is 146.102.29.0 and the last address is 146.102.32.255.

Solution

We can subtract the first address from the last address in base 256 (see Appendix B). The result is 0.0.3.255 in this base. To find the number of addresses in the range (in decimal), we convert this number to base 10 and add 1 to the result..

$$\text{Number of addresses} = (0 \times 256^3 + 0 \times 256^2 + 3 \times 256^1 + 255 \times 256^0) + 1 = 1024$$

Example 5.6

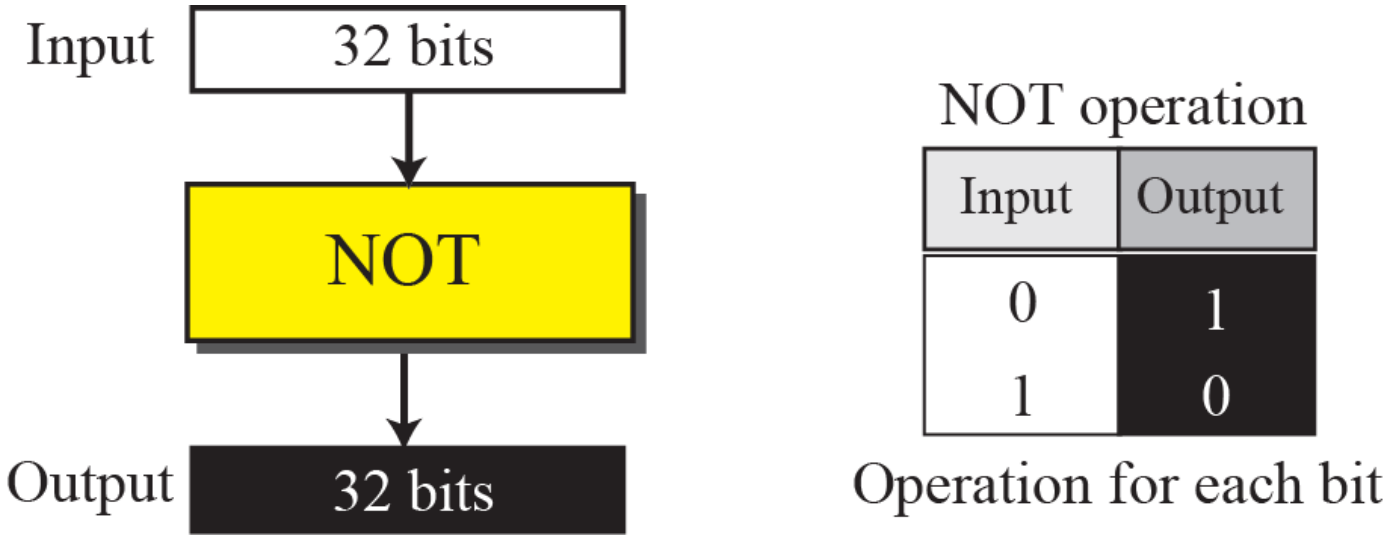
The first address in a range of addresses is 14.11.45.96. If the number of addresses in the range is 32, what is the last address?

Solution

We convert the number of addresses minus 1 to base 256, which is 0.0.0.31. We then add it to the first address to get the last address. Addition is in base 256.

$$\text{Last address} = (14.11.45.96 + 0.0.0.31)_{256} = 14.11.45.127$$

Figure 5.2 *Bitwise NOT operation*



Example 5.7

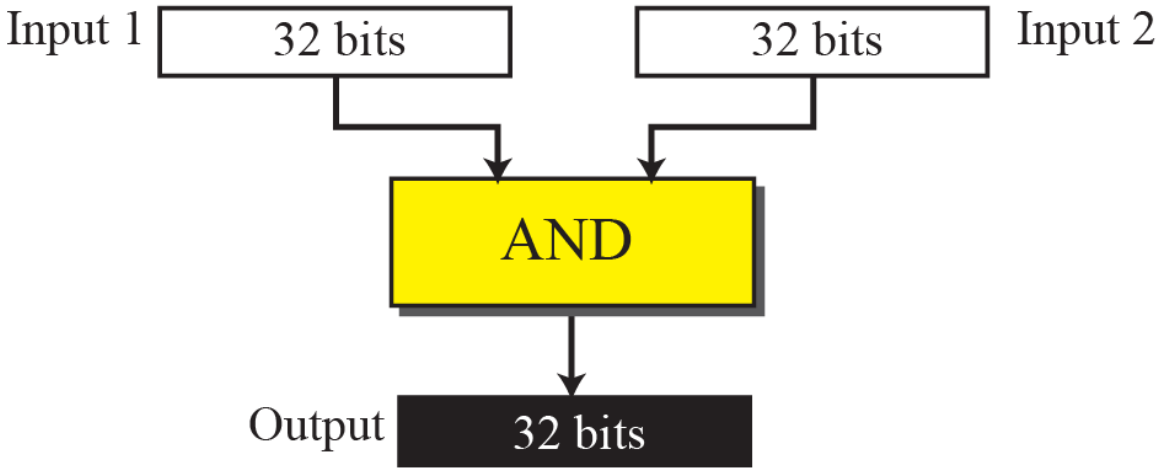
The following shows how we can apply the NOT operation on a 32-bit number in binary.

Original number:	00010001	01111001	00001110	00100011
Complement:	11101110	10000110	11110001	11011100

We can use the same operation using the dotted-decimal representation and the short cut.

Original number:	17	.	121	.	14	.	35
Complement:	238	.	134	.	241	.	220

Figure 5.3 *Bitwise AND operation*



AND

Input 1	Input 2	Output
0	0	0
0	1	0
1	0	0
1	1	1

Operation for each bit

Example 5.8

First number:	00010001	01111001	00001110	00100011
Second number:	11111111	11111111	10001100	00000000
Result	00010001	01111001	00001100	00000000

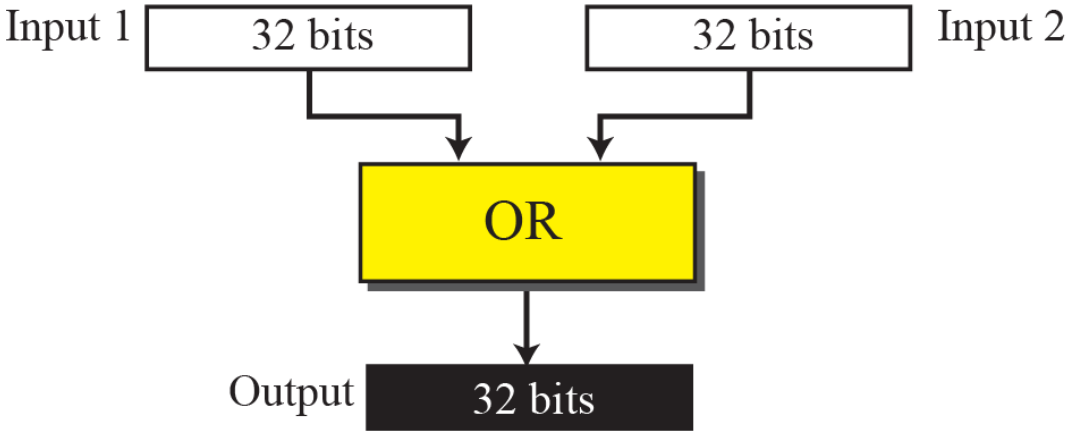
We can use the same operation using the dotted-decimal representation and the short cut.

First number:	17	.	121	.	14	.	35
Second number:	255	.	255	.	140	.	0
Result:	17	.	121	.	12	.	0

We have applied the first short cut on the first, second, and the fourth byte; we have applied the second short cut on the third byte. We have written 14 and 140 as the sum of terms and selected the smaller term in each pair as shown below.

Powers	2^7		2^6		2^5		2^4		2^3		2^2		2^1		2^0
Byte (14)	0	+	0	+	0	+	0	+	8	+	4	+	2	+	0
Byte (140)	128	+	0	+	0	+	0	+	8	+	4	+	0	+	0
Result (12)	0	+	0	+	0	+	0	+	8	+	4	+	0	+	0

Figure 5.4 *Bitwise OR operation*



OR

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	1

Operation for each bit

Example 5.9

The following shows how we can apply the OR operation on two 32-bit numbers in binary.

First number:	00010001	01111001	00001110	00100011
Second number:	11111111	11111111	10001100	00000000
Result	11111111	11111111	10001110	00100011

We can use the same operation using the dotted-decimal representation and the short cut.

First number:	17	.	121	.	14	.	35
Second number:	255	.	255	.	140	.	0
Result:	255	.	255	.	142	.	35

We have used the first short cut for the first and second bytes and the second short cut for the third byte.

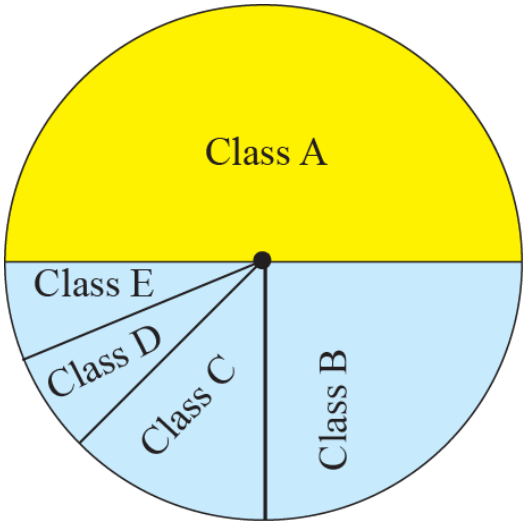
5-2 CLASSFUL ADDRESSING

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 that supersedes the original architecture. In this section, we introduce classful addressing because it paves the way for understanding classless addressing and justifies the rationale for moving to the new architecture. Classless addressing is discussed in the next section.

Topics Discussed in the Section

- ✓ Classes
- ✓ Classes and Blocks
- ✓ Two-Level Addressing
- ✓ Three-Level Addressing: Subnetting
- ✓ Supernetting

Figure 5.5 *Occupation of address space*



Class A: 2^{31} = 2,147,483,648 addresses, 50%

Class B: 2^{30} = 1,073,741,824 addresses, 25%

Class C: 2^{29} = 536,870,912 addresses, 12.5%

Class D: 2^{28} = 268,435,456 addresses, 6.25%

Class E: 2^{28} = 268,435,456 addresses, 6.25%

Figure 5.6 *Finding the class of address*

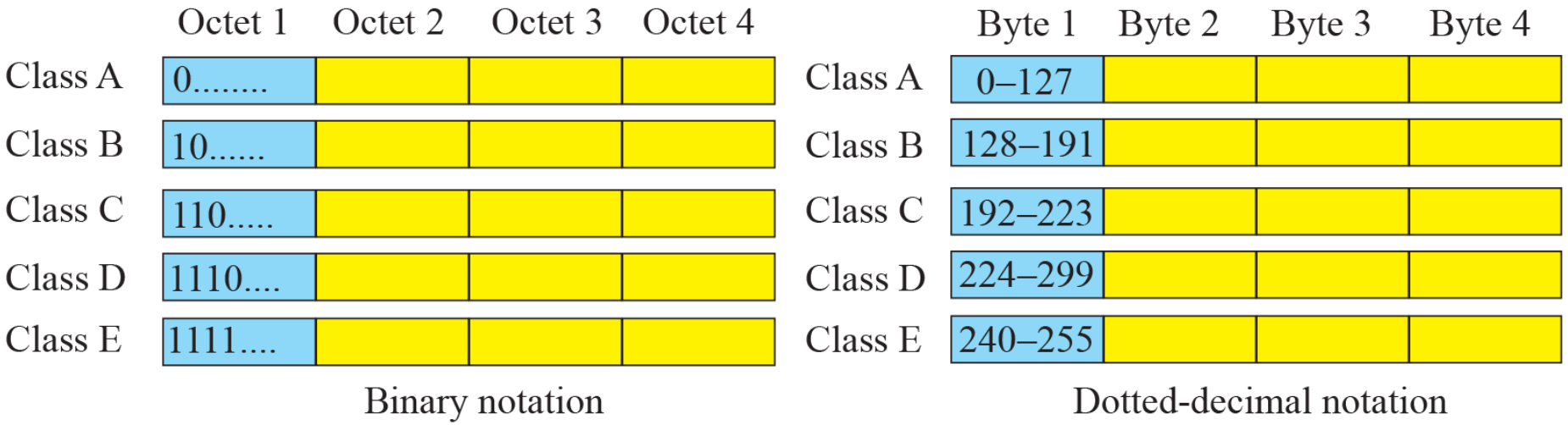
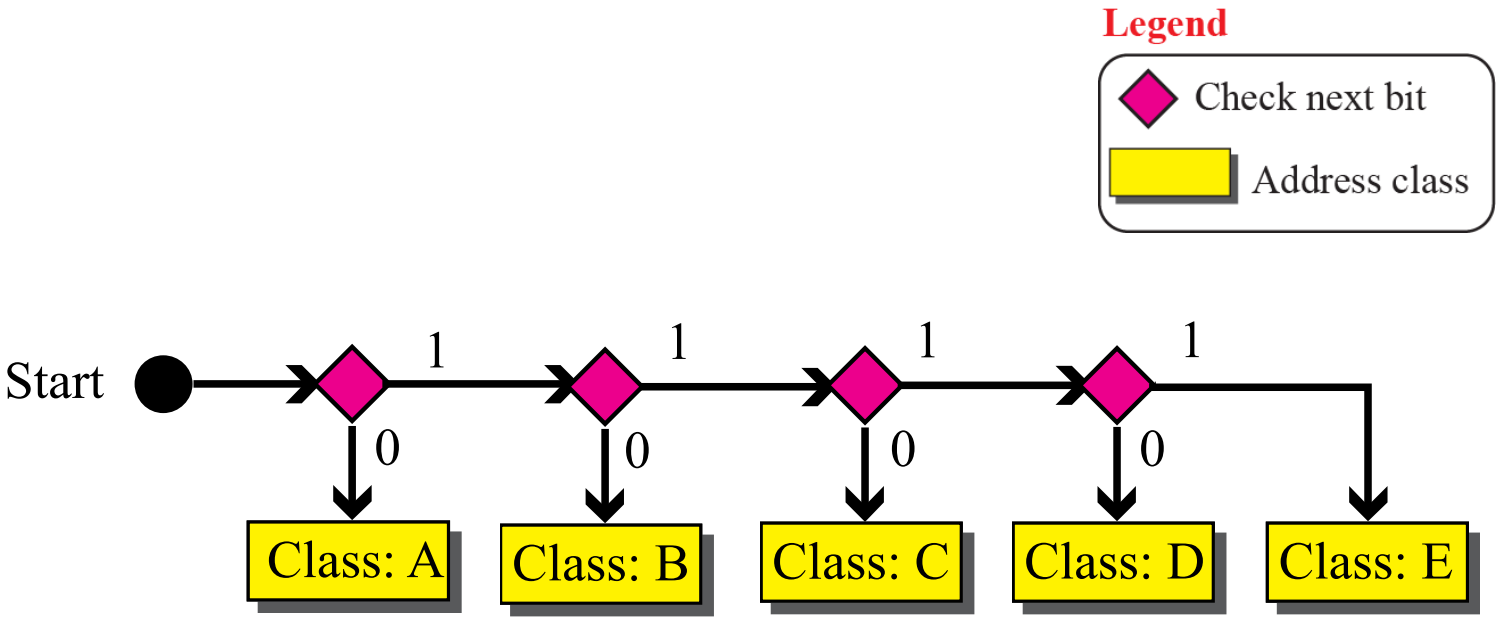


Figure 5.7 *Finding the class of an address using continuous checking*



Example 5.10

Find the class of each address:

a. 00000001 00001011 00001011 11101111
b. 11000001 10000011 00011011 11111111
c. 10100111 11011011 10001011 01101111
d. 11110011 10011011 11111011 00001111

Solution

See the procedure in Figure 5.7.

a. The first bit is 0. This is a class A address.
b. The first 2 bits are 1; the third bit is 0. This is a class C address.
c. The first bit is 1; the second bit is 0. This is a class B address.
d. The first 4 bits are 1s. This is a class E address.

Example 5.11

Find the class of each address:

a. 227.12.14.87

b. 193.14.56.22

c. 14.23.120.8

d. 252.5.15.111

Solution

a. The first byte is 227 (between 224 and 239); the class is D.

b. The first byte is 193 (between 192 and 223); the class is C.

c. The first byte is 14 (between 0 and 127); the class is A.

d. The first byte is 252 (between 240 and 255); the class is E.

Figure 5.8 *Netid and hostid*

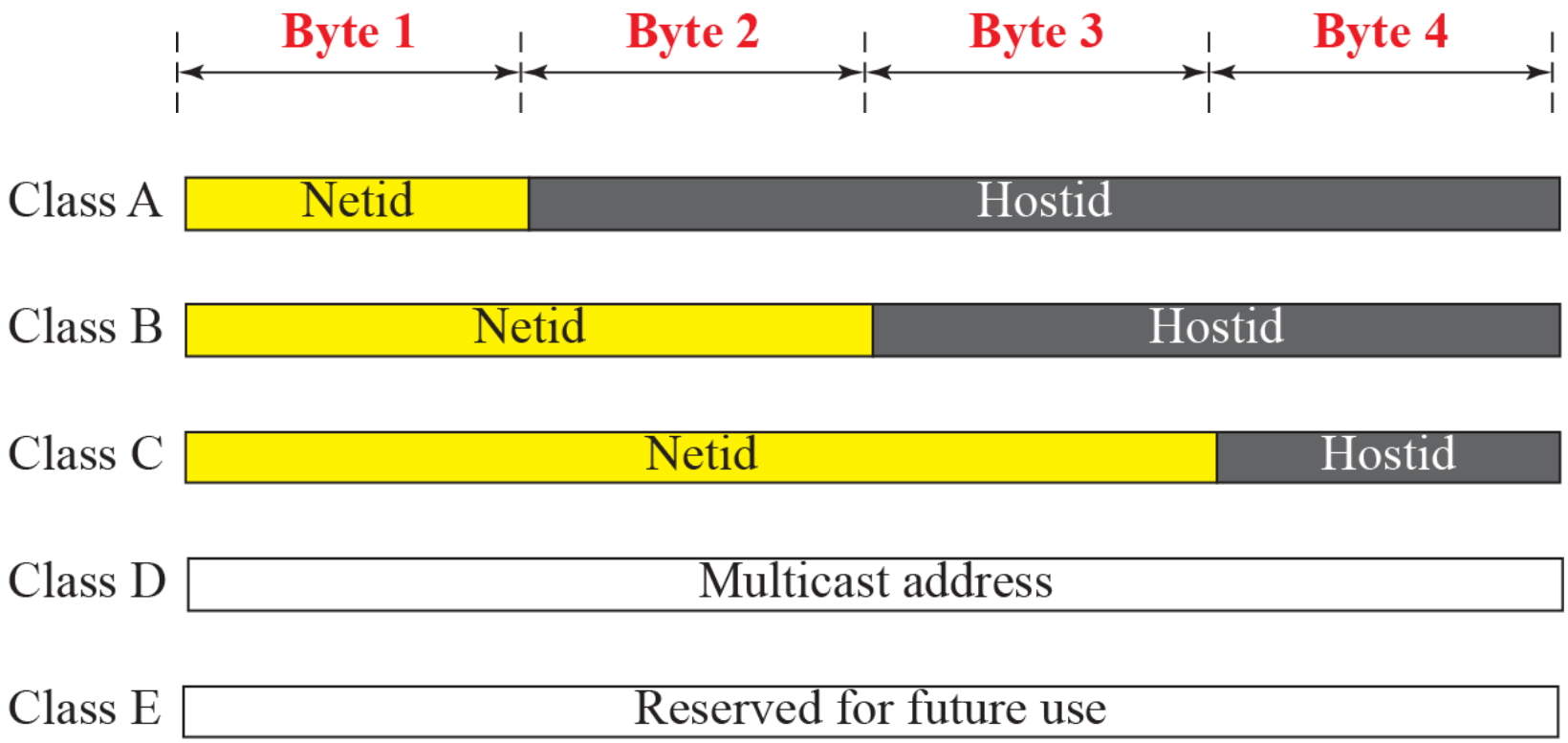
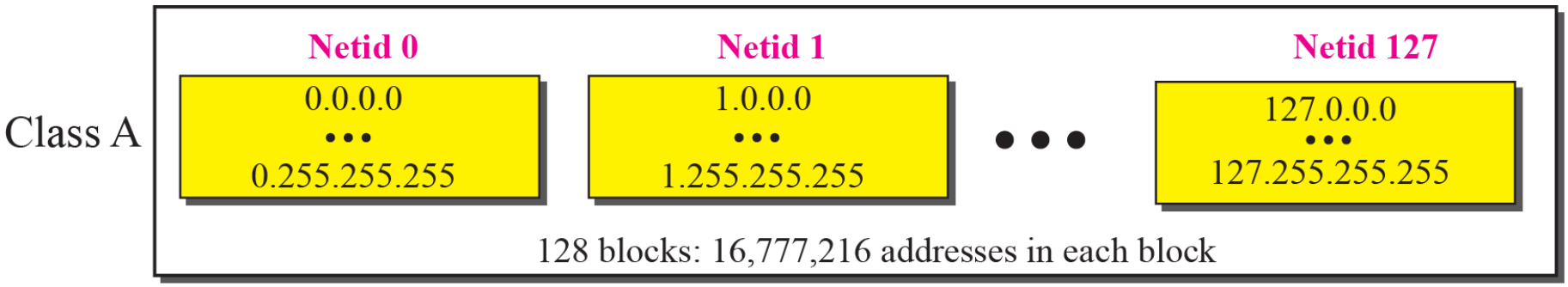
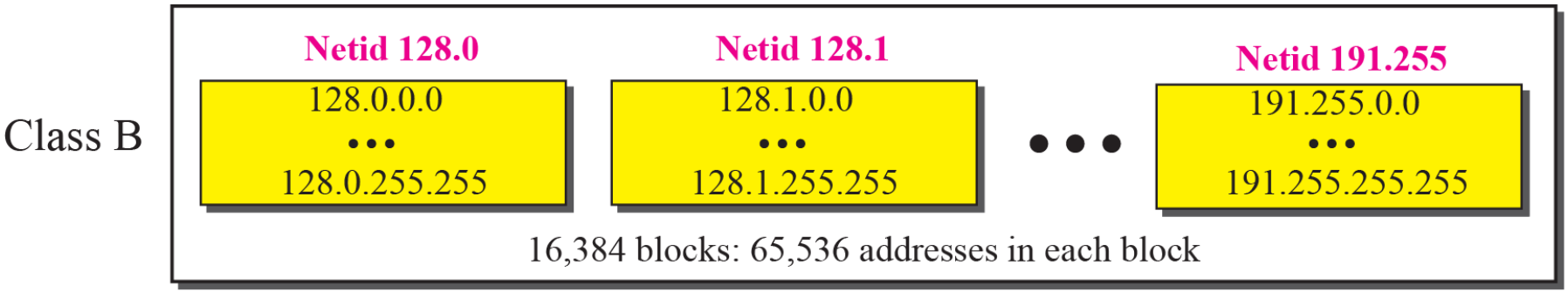


Figure 5.9 *Blocks in Class A*



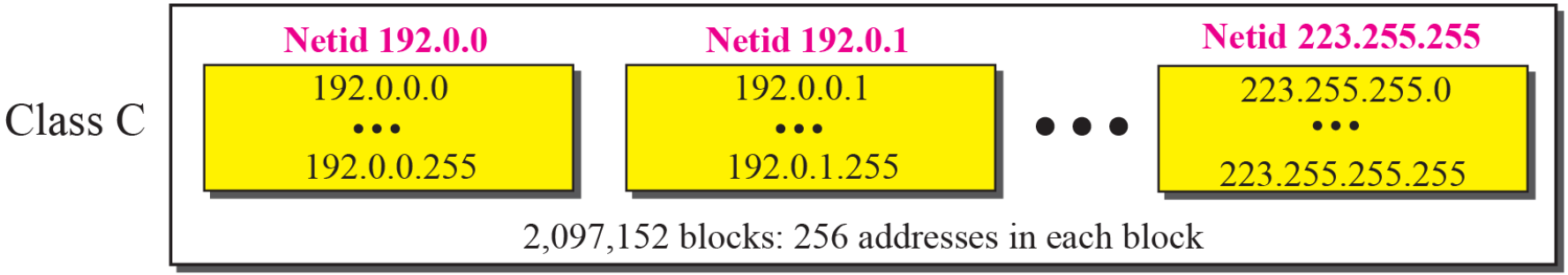
Millions of class A addresses are wasted.

Figure 5.10 *Blocks in Class B*



Many class B addresses are wasted.

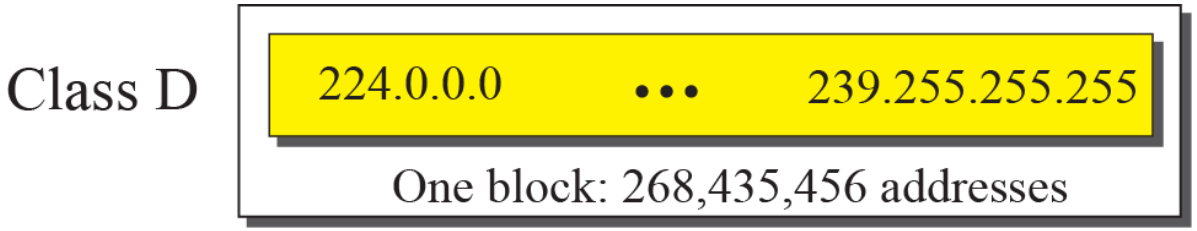
Figure 5.11 *Blocks in Class C*



Note

Not so many organizations are so small to have a class C block.

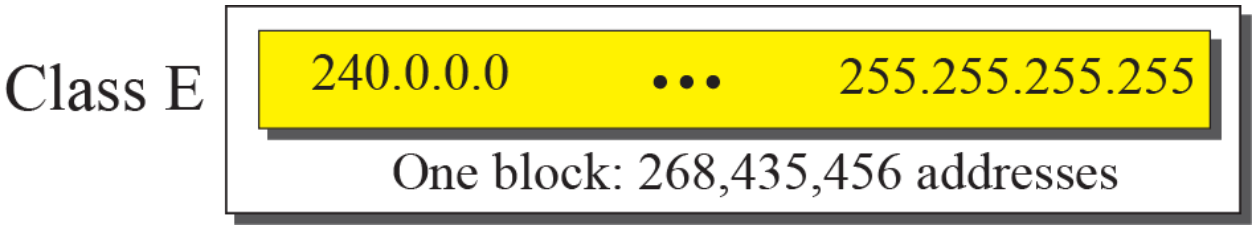
Figure 5.12 *The single block in Class D*



Note

Class D addresses are made of one block, used for multicasting.

Figure 5.13 *The single block in Class E*



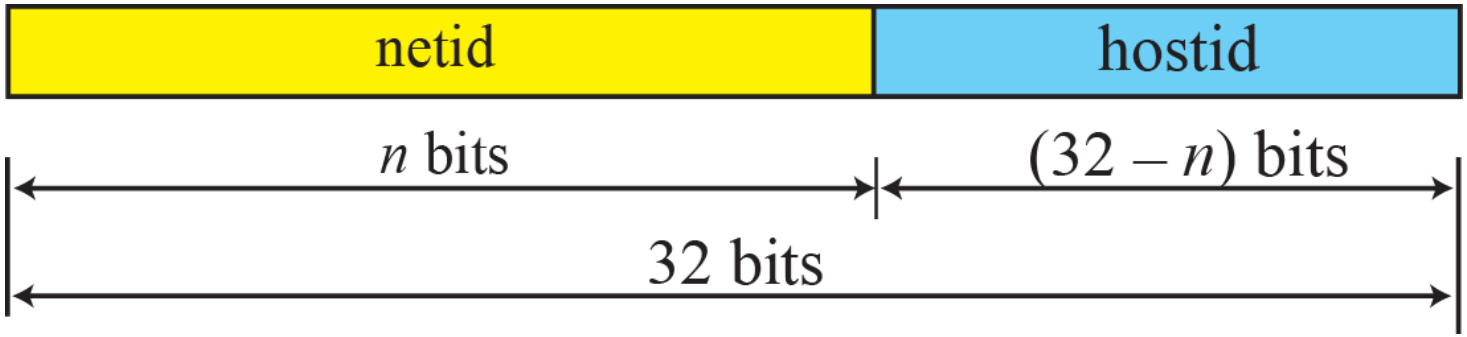
Note

The only block of class E addresses was reserved for future purposes.

Note

The range of addresses allocated to an organization in classful addressing was a block of addresses in Class A, B, or C.

Figure 5.14 *Two-level addressing in classful addressing*



Class A: $n = 8$

Class B: $n = 16$

Class C: $n = 24$

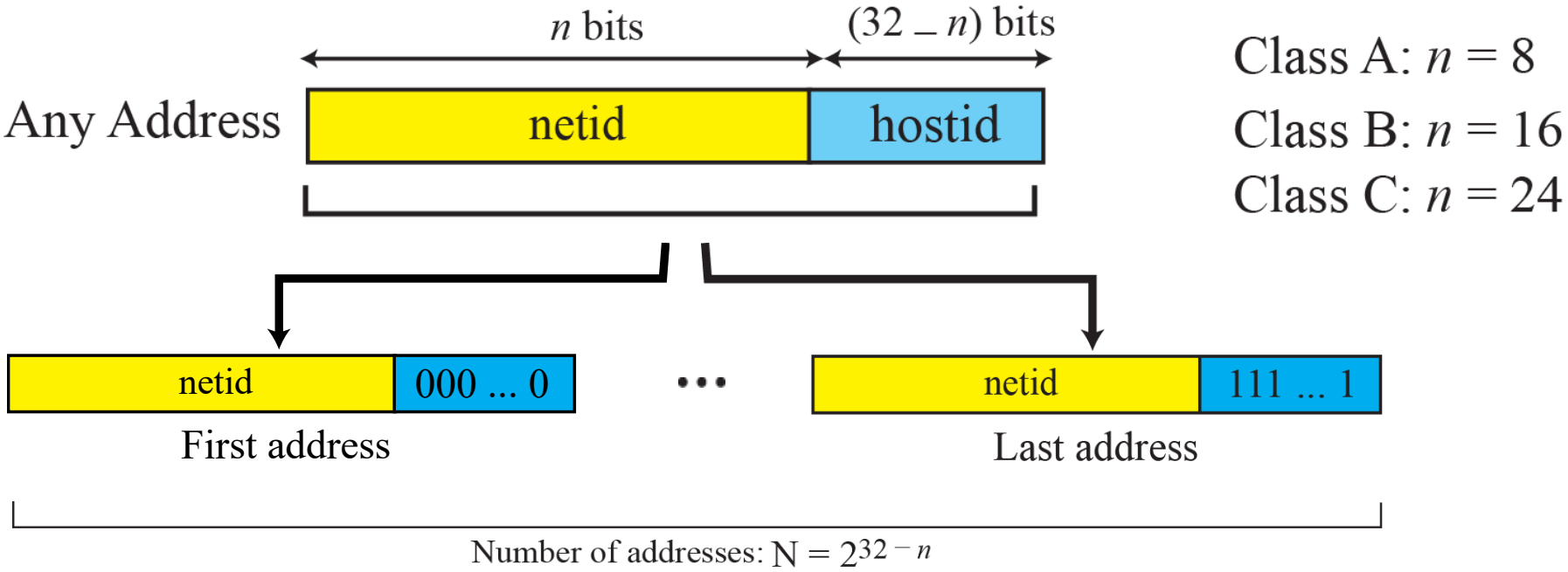
Example 5.12

Two-level addressing can be found in other communication systems. For example, a telephone system inside the United States can be thought of as two parts: area code and local part. The area code defines the area, the local part defines a particular telephone subscriber in that area.

(626) 3581301

The area code, 626, can be compared with the netid, the local part, 3581301, can be compared to the hostid.

Figure 5.15 *Information extraction in classful addressing*



Example 5.13

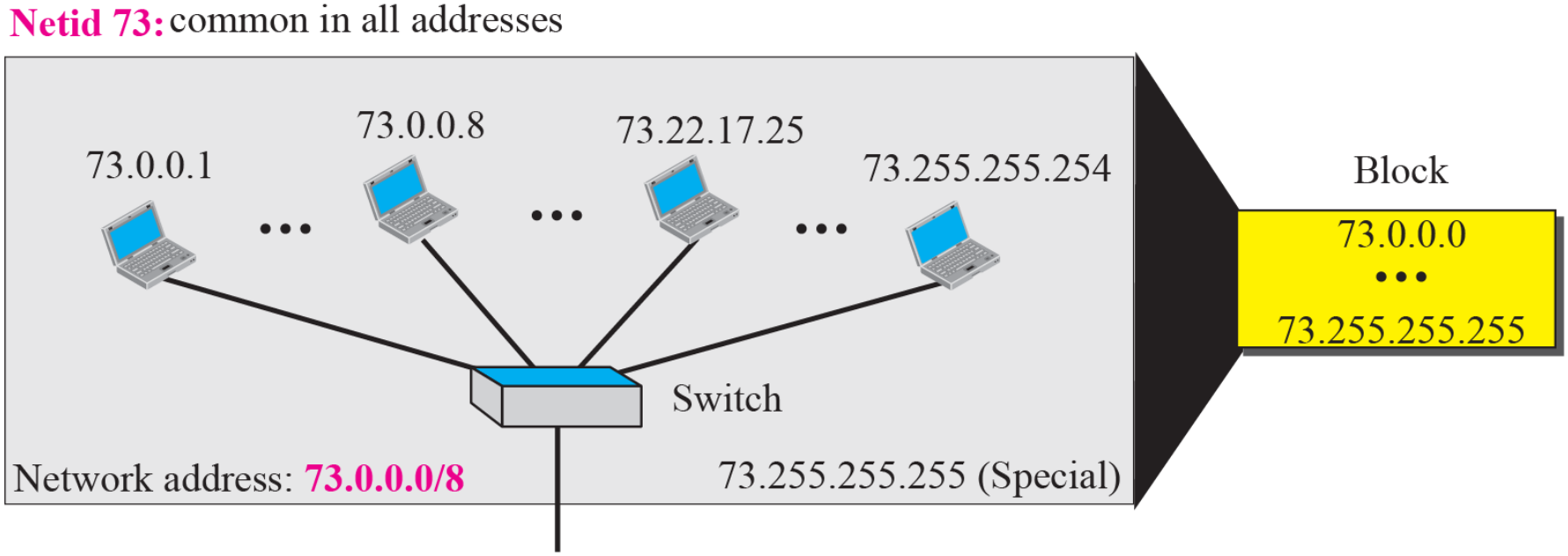
An address in a block is given as 73.22.17.25. Find the number of addresses in the block, the first address, and the last address.

Solution

Figure 5.16 shows a possible configuration of the network that uses this block.

1. The number of addresses in this block is $N = 2^{32-n} = 16{,}777{,}216$.
2. To find the first address, we keep the leftmost 8 bits and set the rightmost 24 bits all to 0s. The first address is 73.0.0.0/8, in which 8 is the value of *n*.
3. To find the last address, we keep the leftmost 8 bits and set the rightmost 24 bits all to 1s. The last address is 73.255.255.255.

Figure 5.16 *Solution to Example 5.13*



Example 5.14

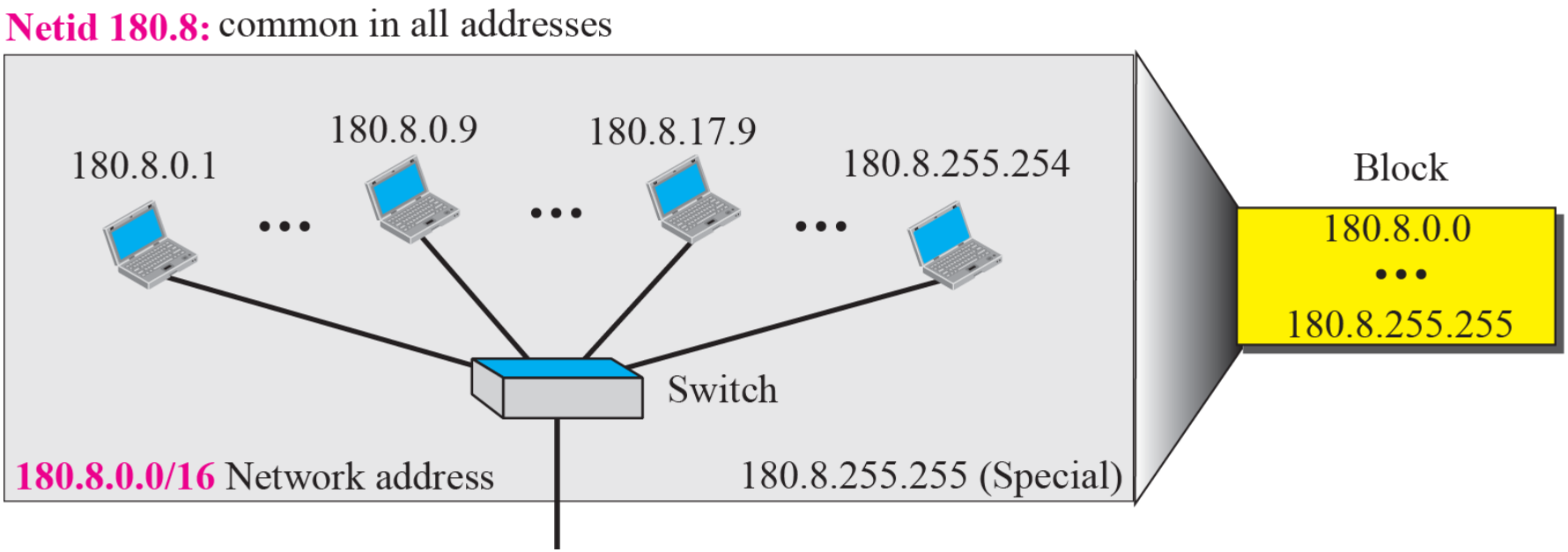
An address in a block is given as 180.8.17.9. Find the number of addresses in the block, the first address, and the last address.

Solution

Figure 5.17 shows a possible configuration of the network that uses this block.

1. The number of addresses in this block is $N = 2^{32-n} = 65,536$.
2. To find the first address, we keep the leftmost 16 bits and set the rightmost 16 bits all to 0s. The first address is 180.8.0.0/16, in which 16 is the value of *n*.
3. To find the last address, we keep the leftmost 16 bits and set the rightmost 16 bits all to 1s. The last address is 180.8.255.255.

Figure 5.17 *Solution to Example 5.14*



Example 5.15

An address in a block is given as 200.11.8.45. Find the number of addresses in the block, the first address, and the last address.

Solution

Figure 5.17 shows a possible configuration of the network that uses this block.

1. The number of addresses in this block is $N = 2^{32-n} = 256$.
2. To find the first address, we keep the leftmost 24 bits and set the rightmost 8 bits all to 0s. The first address is 200.11.8.0/16, in which 24 is the value of *n*.
3. To find the last address, we keep the leftmost 24 bits and set the rightmost 8 bits all to 1s. The last address is 200.11.8.255/16.

Figure 5.18 *Solution to Example 5.15*

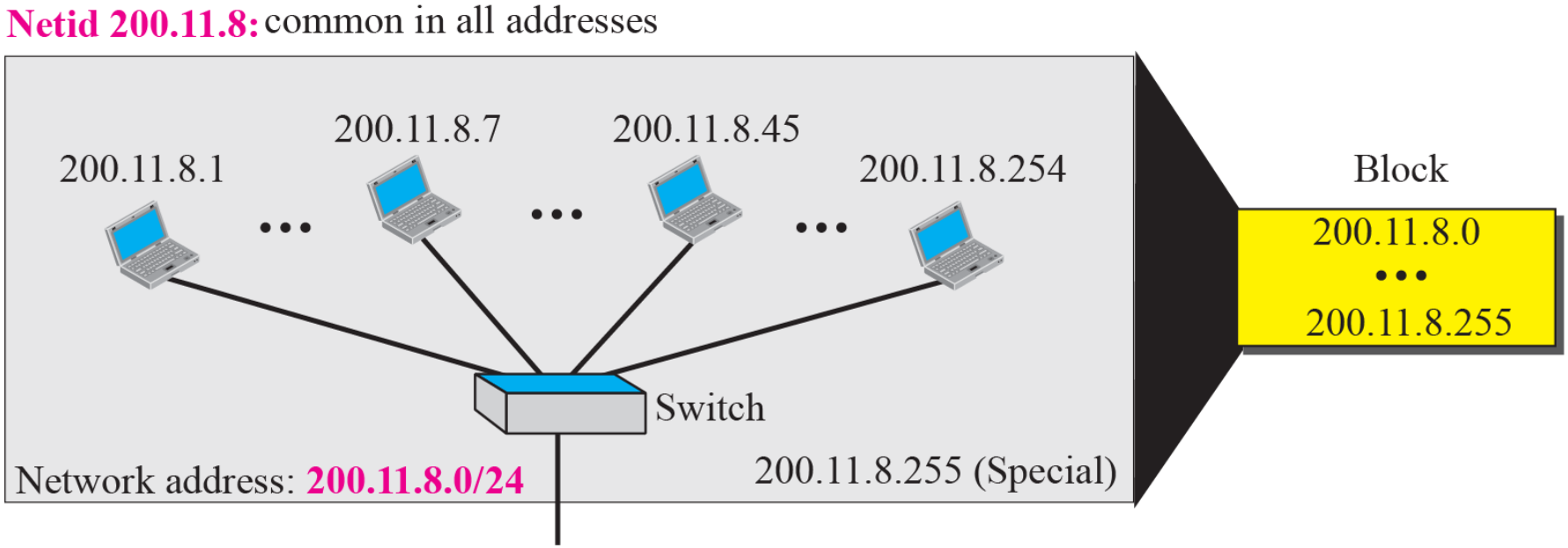
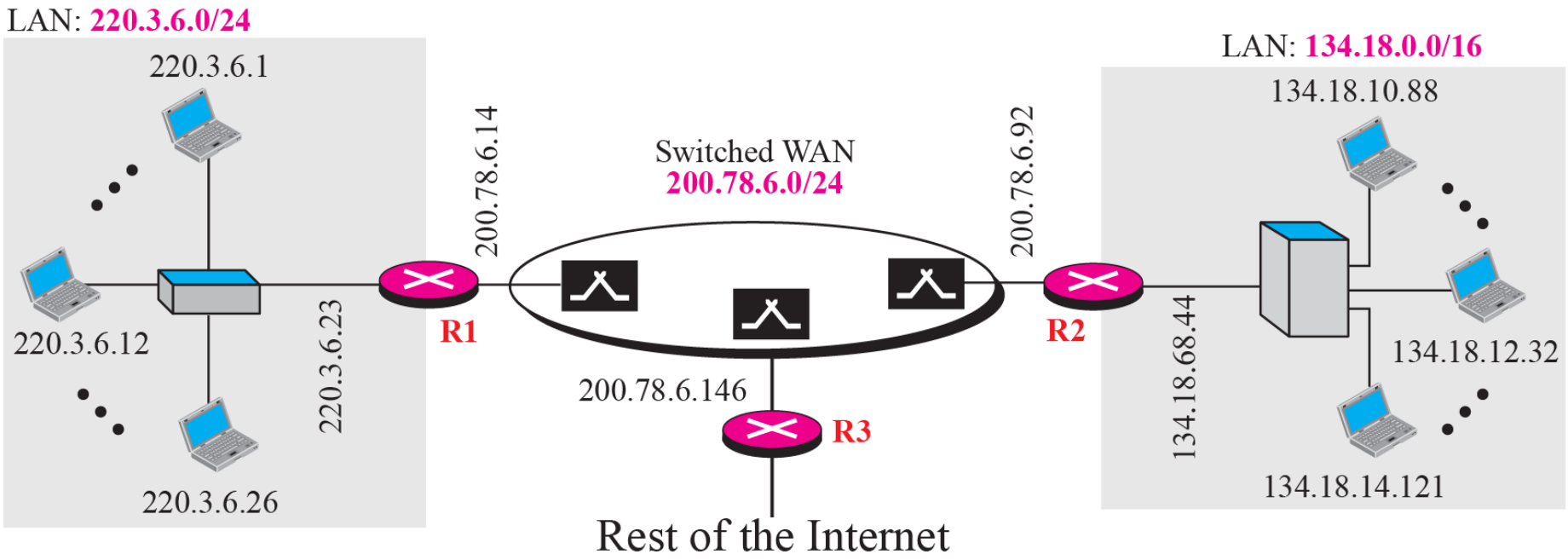


Figure 5.19 *Sample Internet*



Note

The network address is the identifier of a network.

Figure 5.20 *Network addresses*

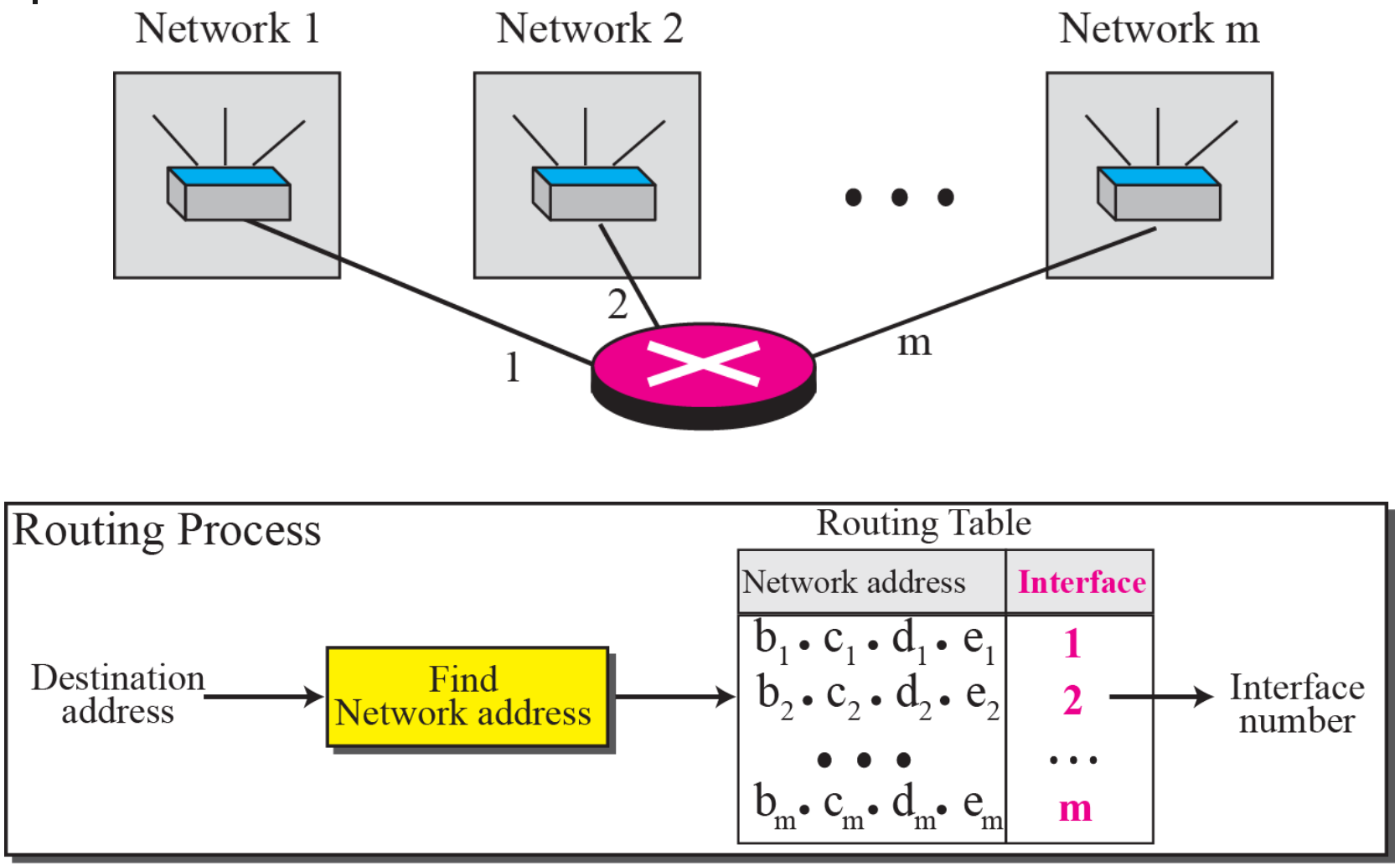


Figure 5.21 *Network mask*

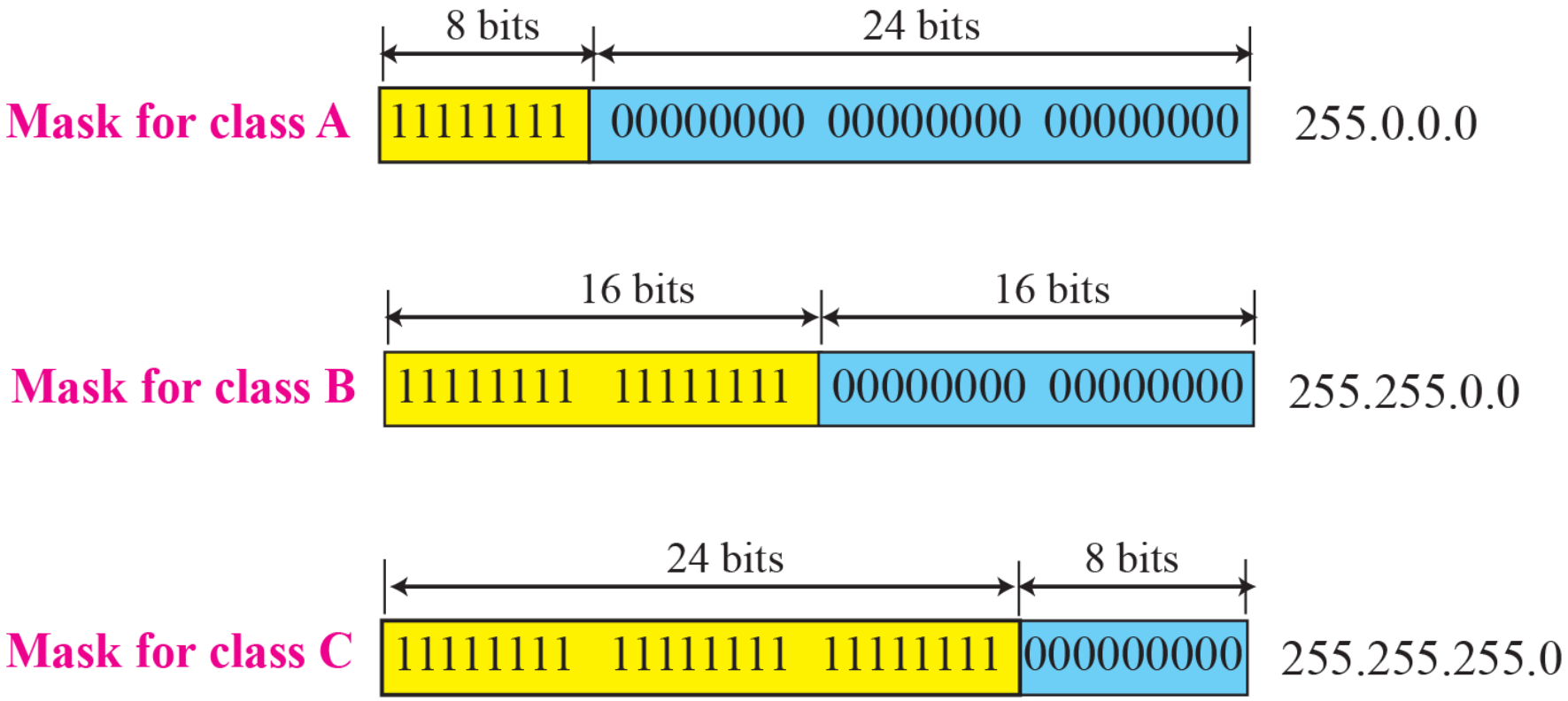
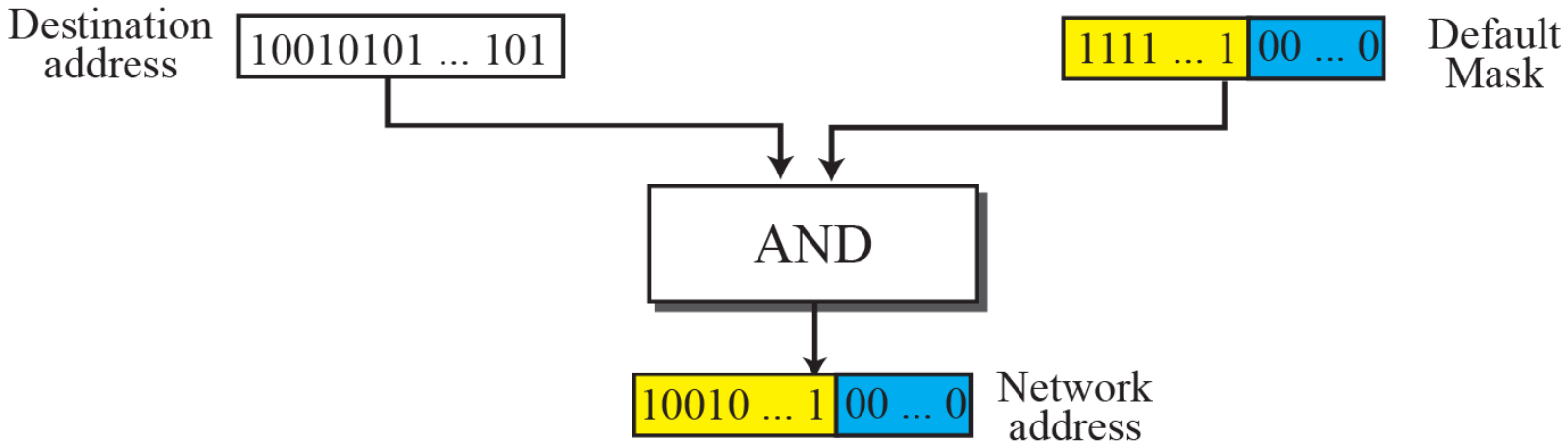


Figure 5.22 *Finding a network address using the default mask*



Example 5.16

A router receives a packet with the destination address 201.24.67.32. Show how the router finds the network address of the packet.

Solution

Since the class of the address is B, we assume that the router applies the default mask for class B, 255.255.0.0 to find the network address.

Destination address	→	201	.	24	.	67	.	32
Default mask	→	255	.	255	.	0	.	0
Network address	→	201	.	24	.	0	.	0

Example 5.17

Three-level addressing can be found in the telephone system if we think about the local part of a telephone number as an exchange and a subscriber connection:

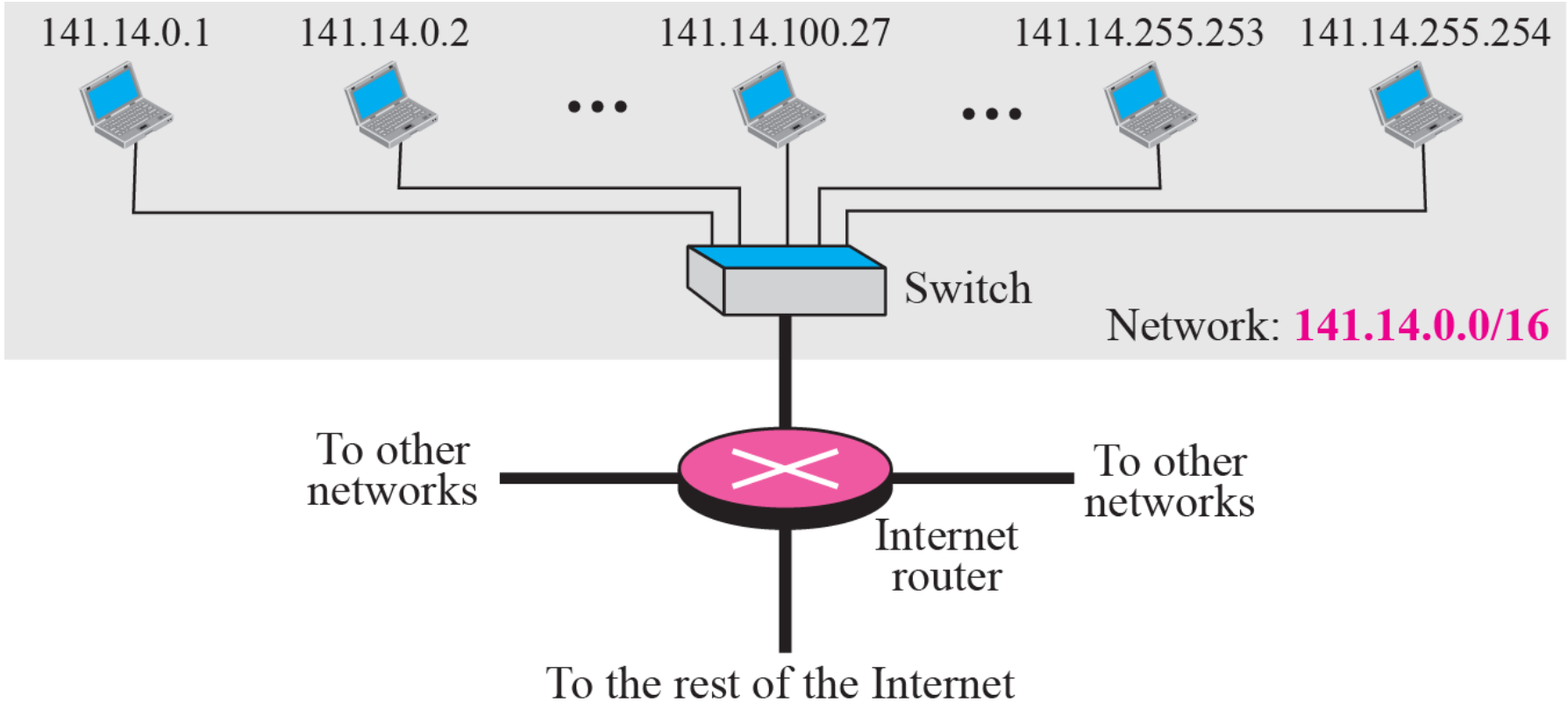
(626) 358 - 1301

in which 626 is the area code, 358 is the exchange, and 1301 is the subscriber connection.

Example 5.18

Figure 5.23 shows a network using class B addresses before subnetting. We have just one network with almost 2^{16} hosts. The whole network is connected, through one single connection, to one of the routers in the Internet. Note that we have shown /16 to show the length of the netid (class B).

Figure 5.23 *Example 5.18*



What is Supernetting? (Route Summarization)

- Supernetting is the opposite of Subnetting.
- In subnetting, a single big network is divided into multiple smaller subnetworks.
- In Supernetting, multiple networks are combined into a bigger network termed as a Supernetwork or Supernet.
- Supernetting is mainly used in Route Summarization, where routes to multiple networks with similar network prefixes are combined into a single routing entry, with the routing entry pointing to a Super network, encompassing all the networks.
- This in turn significantly reduces the size of routing tables and also the size of routing updates exchanged by routing protocols.

- 192.168.0.0/24
- 192.168.1.0/24
- 192.168.2.0/24
- 192.168.3.0/24

- Write all the IP Addresses in binary like so:
- 192.168.0.0/24
  11000000.10101000.00000000.00000000
- 192.168.1.0/24
  11000000.10101000.00000001.00000000
- 192.168.2.0/24
  11000000.10101000.00000010.00000000
- •192.168.3.0/24
  11000000.10101000.00000011.00000000

Step 2: Find matching bits from left to right

- 11000000.10101000.00000000.00000000
- 11000000.10101000.00000001.00000000
- 11000000.10101000.00000010.00000000
- 11000000.10101000.00000011.00000000

Step 3: Re write the matching numbers and add the remaining zeros, because you are converting network bits into host bits. This will be your NEW NETWORK ID, the route that you will be advertising. (A summarized route)
11000000.10101000.00000000.00000000 = 192.168.0.0

- Step 4: Find the new subnet mask. Put “1s” in the matching networking part, and all zeros in the host part.
- 11111111.11111111.11111100.00000000
- This your new subnet mask 255.255.252.0
- •Your new summarized route is 192.168.0.0/22

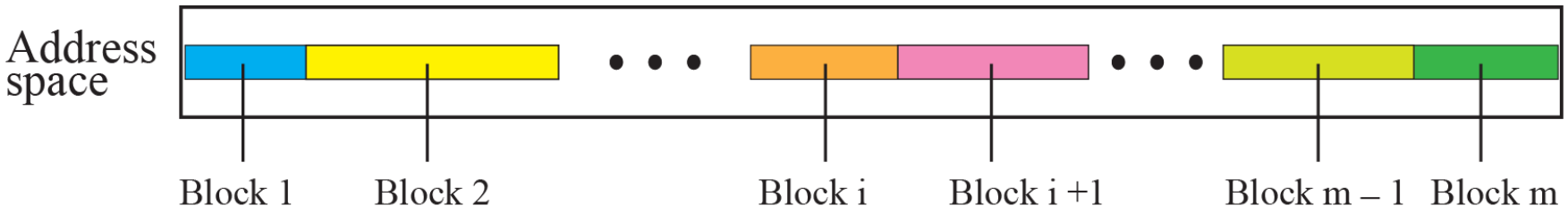
5-3 CLASSLESS ADDRESSING

Subnetting and supernetting in classful addressing did not really solve the address depletion problem. With the growth of the Internet, it was clear that a larger address space was needed as a long-term solution. Although the long-range solution has already been devised and is called IPv6, a short-term solution was also devised to use the same address space but to change the distribution of addresses to provide a fair share to each organization. The short-term solution still uses IPv4 addresses, but it is called *classless addressing*.

Topics Discussed in the Section

- ✓ **Variable –Length Blocks**
- ✓ **Two-Level Addressing**
- ✓ **Block Allocation**
- ✓ **Subnetting**

Figure 5.27 *Variable-length blocks in classless addressing*



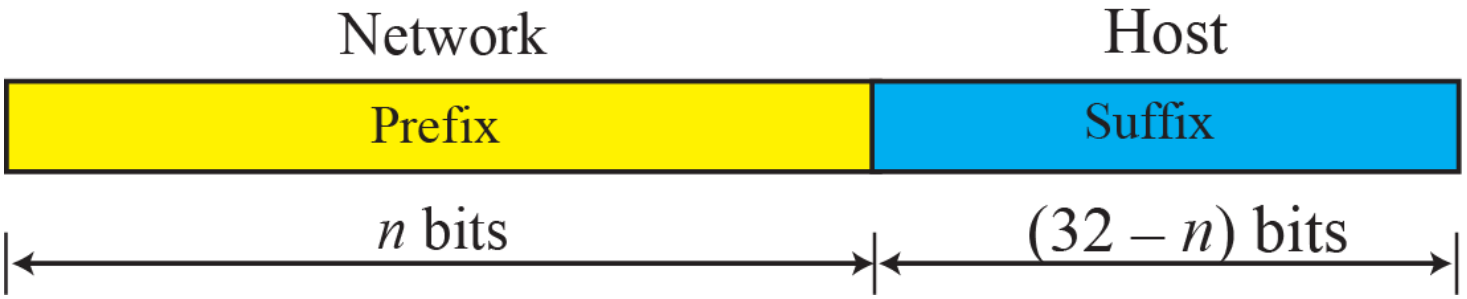
Note

In classless addressing, the prefix defines the network and the suffix defines the host.

Two-Level Addressing

In classful addressing, two-level addressing was provided by dividing an address into *netid* and *hostid*. The netid defined the network; the hostid defined the host in the network.

The same idea can be applied in classless addressing. When an organization is granted a block of addresses, the block is actually divided into two parts, the **prefix** and the **suffix.** The prefix plays the same role as the netid; the suffix plays the same role as the hostid. All addresses in the block have the same prefix; each address has a different suffix. Figure 5.28 shows the prefix and suffix in a classless block.



Note

The prefix length in classless addressing can be 1 to 32.

Example 5.22

What is the prefix length and suffix length if the whole Internet is considered as one single block with 4,294,967,296 addresses?

Solution

In this case, the prefix length is 0 and the suffix length is 32. All 32 bits vary to define 2^{32} = 4,294,967,296 hosts in this single block.

Example 5.23

What is the prefix length and suffix length if the Internet is divided into 4,294,967,296 blocks and each block has one single address?

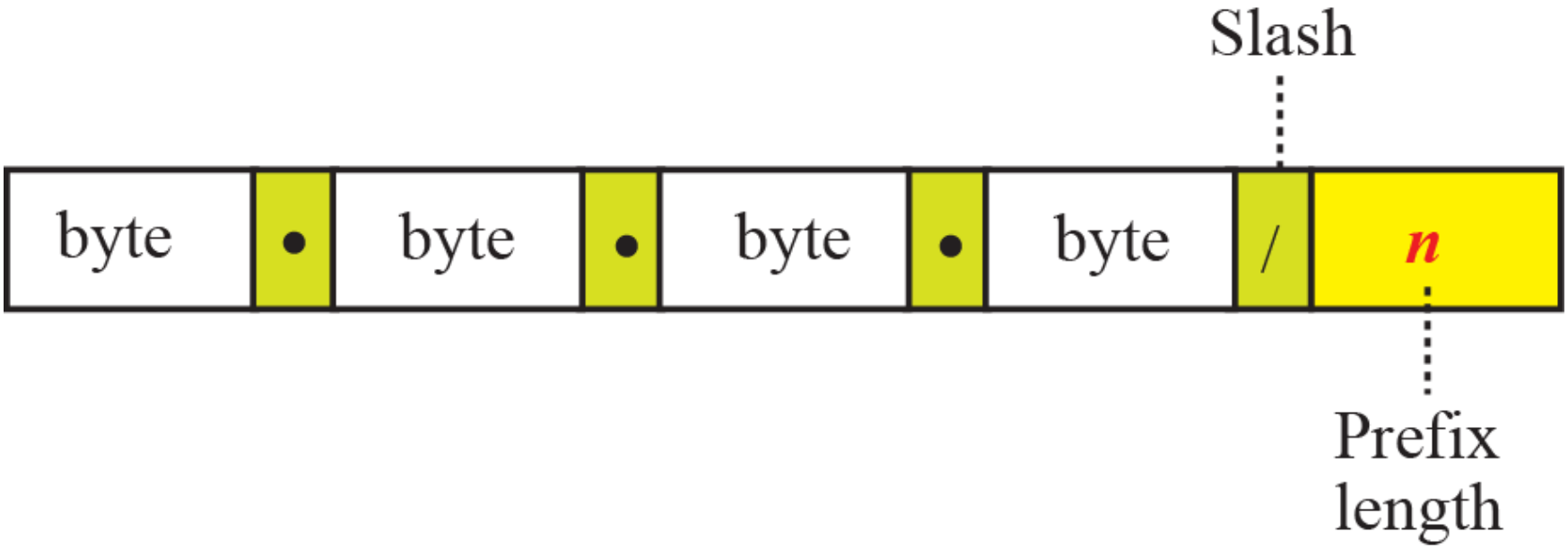
Solution

In this case, the prefix length for each block is 32 and the suffix length is 0. All 32 bits are needed to define 2^{32} = 4,294,967,296 blocks. The only address in each block is defined by the block itself.

Example 5.24

The number of addresses in a block is inversely related to the value of the prefix length, *n*. A small *n* means a larger block; a large *n* means a small block.

Figure 5.29 *Slash notation*



Note

In classless addressing, we need to know one of the addresses in the block and the prefix length to define the block.

Example 5.25

In classless addressing, an address cannot per se define the block the address belongs to. For example, the address 230.8.24.56 can belong to many blocks some of them are shown below with the value of the prefix associated with that block:

Prefix length:16	→	Block:	230.8.0.0	to	230.8.255.255
Prefix length:20	→	Block:	230.8.16.0	to	230.8.31.255
Prefix length:26	→	Block:	230.8.24.0	to	230.8.24.63
Prefix length:27	→	Block:	230.8.24.32	to	230.8.24.63
Prefix length:29	→	Block:	230.8.24.56	to	230.8.24.63
Prefix length:31	→	Block:	230.8.24.56	to	230.8.24.57

Example 5.26

The following addresses are defined using slash notations.

a. In the address 12.23.24.78/8, the network mask is 255.0.0.0. The mask has eight 1s and twenty-four 0s. The prefix length is 8; the suffix length is 24.

b. In the address 130.11.232.156/16, the network mask is 255.255.0.0. The mask has sixteen 1s and sixteen 0s.The prefix length is 16; the suffix length is 16.

c. In the address 167.199.170.82/27, the network mask is 255.255.255.224. The mask has twenty-seven 1s and five 0s. The prefix length is 27; the suffix length is 5.

Example 5.27

One of the addresses in a block is 167.199.170.82/27. Find the number of addresses in the network, the first address, and the last address.

Solution

The value of n is 27. The network mask has twenty-seven 1s and five 0s. It is 255.255.255.240.

a. The number of addresses in the network is $2^{32-n} = 32$.

b. We use the AND operation to find the first address (network address). The first address is 167.199.170.64/27.

Address in binary:	10100111	11000111	10101010	01010010
Network mask:	11111111	11111111	11111111	11100000
First address:	10100111	11000111	10101010	01000000

Example 5.27 *Continued*

c. To find the last address, we first find the complement of the network mask and then OR it with the given address: The last address is 167.199.170.95/27.

Address in binary:	10100111	11000111	10101010	01010010
Complement of network mask:	00000000	00000000	00000000	00011111
Last address:	10100111	11000111	10101010	01011111

Example 5.28

One of the addresses in a block is 17.63.110.114/24. Find the number of addresses, the first address, and the last address in the block.

Solution

The network mask is 255.255.255.0.

a. The number of addresses in the network is $2^{32-24} = 256$.

b. To find the first address, we use the short cut methods discussed early in the chapter. The first address is 17.63.110.0/24.

Address:	17	.	63	.	110	.	114
Network mask:	255	.	255	.	255	.	0
First address (AND):	17	.	63	.	110	.	0

Example 5.28 *Continued*

c. To find the last address, we use the complement of the network mask and the first short cut method we discussed before. The last address is 17.63.110.255/24.

Address in binary:	10100111 11000111 10101010 01010010
Complement of network mask:	00000000 00000000 00000000 00011111
Last address:	10100111 11000111 10101010 01011111

Example 5.29

One of the addresses in a block is 110.23.120.14/20. Find the number of addresses, the first address, and the last address in the block.

Solution

The network mask is 255.255.240.0.

a. The number of addresses in the network is $2^{32-20} = 4096$.

b. To find the first address, we apply the first short cut to bytes 1, 2, and 4 and the second short cut to byte 3. The first address is 110.23.112.0/20.

Address:	110	.	23	.	120	.	14
Network mask:	255	.	255	.	240	.	0
First address (AND):	110	.	23	.	112	.	0

Example 5.29 *Continued*

c. To find the last address, we apply the first short cut to bytes 1, 2, and 4 and the second short cut to byte 3. The OR operation is applied to the complement of the mask. The last address is 110.23.127.255/20.

Address:	110	.	23	.	120	.	14
Network mask:	0	.	0	.	15	.	255
Last address (OR):	110	.	23	.	127	.	255

Example 5.30

An ISP has requested a block of 1000 addresses. The following block is granted.

a. Since 1000 is not a power of 2, 1024 addresses are granted (1024 = 210).
b. The prefix length for the block is calculated as n = 32 − log21024 = 22.
c. The beginning address is chosen as 18.14.12.0 (which is divisible by 1024).

The granted block is 18.14.12.0/22. The first address is 18.14.12.0/22 and the last address is 18.14.15.255/22.

Relation to Classful Addressing

All issues discussed for classless addressing can be applied to classful addressing. As a matter of fact, classful addressing is a special case of the classless addressing in which the blocks in class A, B, and C have the prefix length $n_A = 8$, $n_B = 16$, and $n_C = 24$. A block in classful addressing can be easily changed to a block in class addressing if we use the prefix length defined in Table 5.1.

Table 5.1 *Prefix length for classful addressing*

Class	*Prefix length*	*Class*	*Prefix length*
A	/8	D	/4
B	/16	E	/4
C	/24		

Example 5.31

Assume an organization has given a class A block as 73.0.0.0 in the past. If the block is not revoked by the authority, the classless architecture assumes that the organization has a block 73.0.0.0/8 in classless addressing.

Note

The restrictions applied in allocating addresses for a subnetwork are parallel to the ones used to allocate addresses for a network.

Subnetting

Three levels of hierarchy can be created using subnetting. An organization (or an ISP) that is granted a range of addresses may divide the range into several subranges and assign each subrange to a **subnetwork** (or **subnet**). The concept is the same as we discussed for classful addressing. Note that nothing stops the organization from creating more levels. A subnetwork can be divided into several sub-subnetworks. A sub-subnetwork can be divided into several sub-sub-subnetworks. And so on.

Designing Subnets

The subnetworks in a network should be carefully designed to enable the routing of packets. We assume the total number of addresses granted to the organization is N, the prefix length is n, the assigned number of addresses to each subnetwork is N_{sub}, the prefix length for each subnetwork is n_{sub}, and the total number of subnetworks is s. Then, the following steps need to be carefully followed to guarantee the proper operation of the subnetworks.

1. The number of addresses in each subnetwork should be a power of 2.

2. The prefix length for each subnetwork should be found using the following formula:

3. The starting address in each subnetwork should be divisible by the number of addresses in that subnetwork. This can be achieved if we first assign addresses to larger networks.

Example 5.32

An organization is granted the block 130.34.12.64/26. The organization needs four subnetworks, each with an equal number of hosts. Design the subnetworks and find the information about each network.

Solution

The number of addresses for the whole network can be found as $N = 2^{32 - 26} = 64$. The first address in the network is 130.34.12.64/26 and the last address is 130.34.12.127/26. We now design the subnetworks:

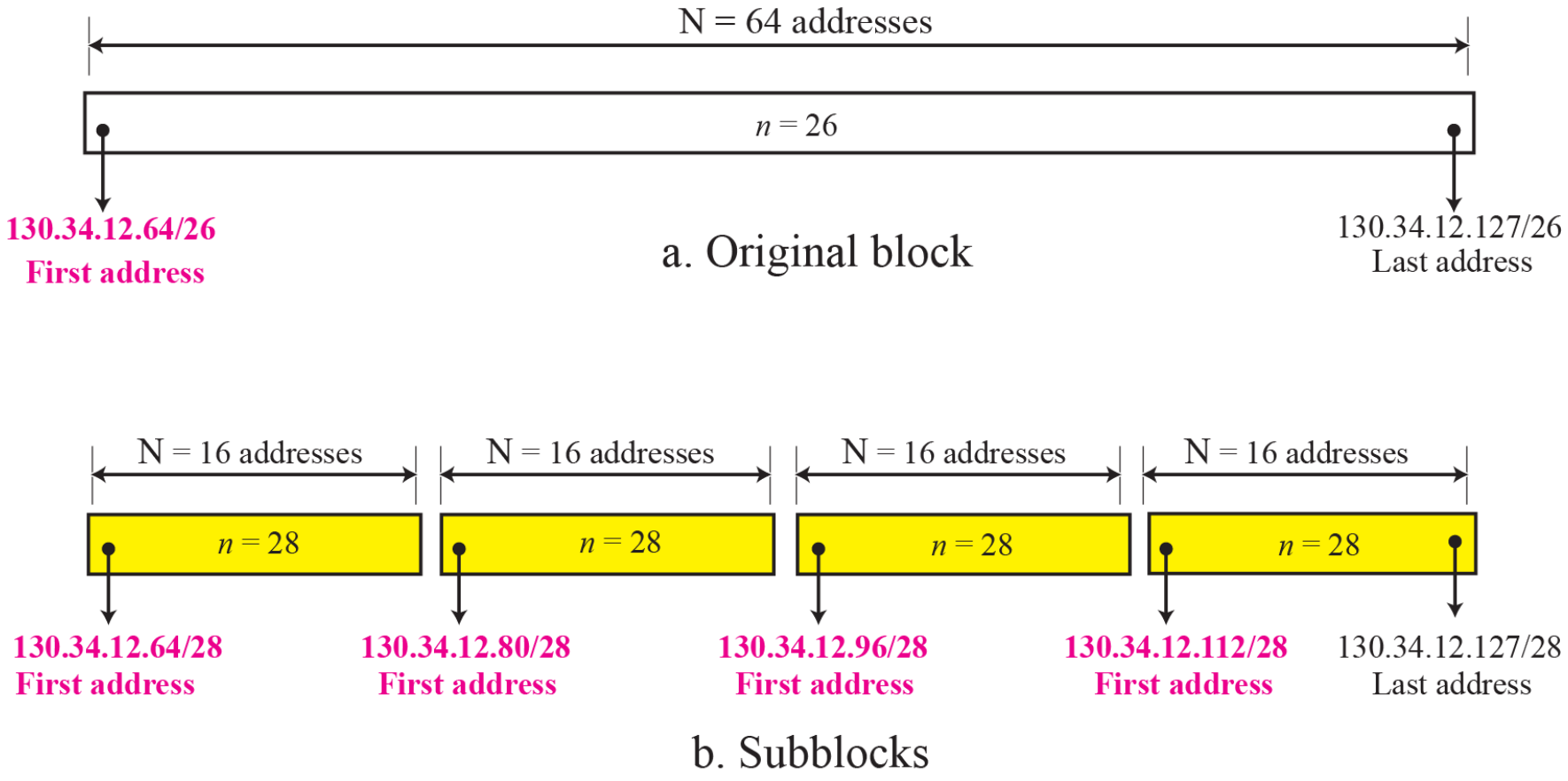
1. We grant 16 addresses for each subnetwork to meet the first requirement (64/16 is a power of 2).
2. The subnetwork mask for each subnetwork is:

$$n_1 = n_2 = n_3 = n_4 = n + \log_2 (N/N_i) = 26 + \log_2 4 = 28$$

Example 5.32 *Continued*

3. We grant 16 addresses to each subnet starting from the first available address. Figure 5.30 shows the subblock for each subnet. Note that the starting address in each subnetwork is divisible by the number of addresses in that subnetwork.

Figure 5.30 *Solution to Example 5.32*



Example 5.33

An organization is granted a block of addresses with the beginning address 14.24.74.0/24. The organization needs to have 3 subblocks of addresses to use in its three subnets as shown below:

- One subblock of 120 addresses.
- One subblock of 60 addresses.
- One subblock of 10 addresses.

Solution

There are $2^{32-24} = 256$ addresses in this block. The first address is 14.24.74.0/24; the last address is 14.24.74.255/24.

a. The number of addresses in the first subblock is not a power of 2. We allocate 128 addresses. The subnet mask is 25. The first address is 14.24.74.0/25; the last address is 14.24.74.127/25.

Example 5.33 Continued

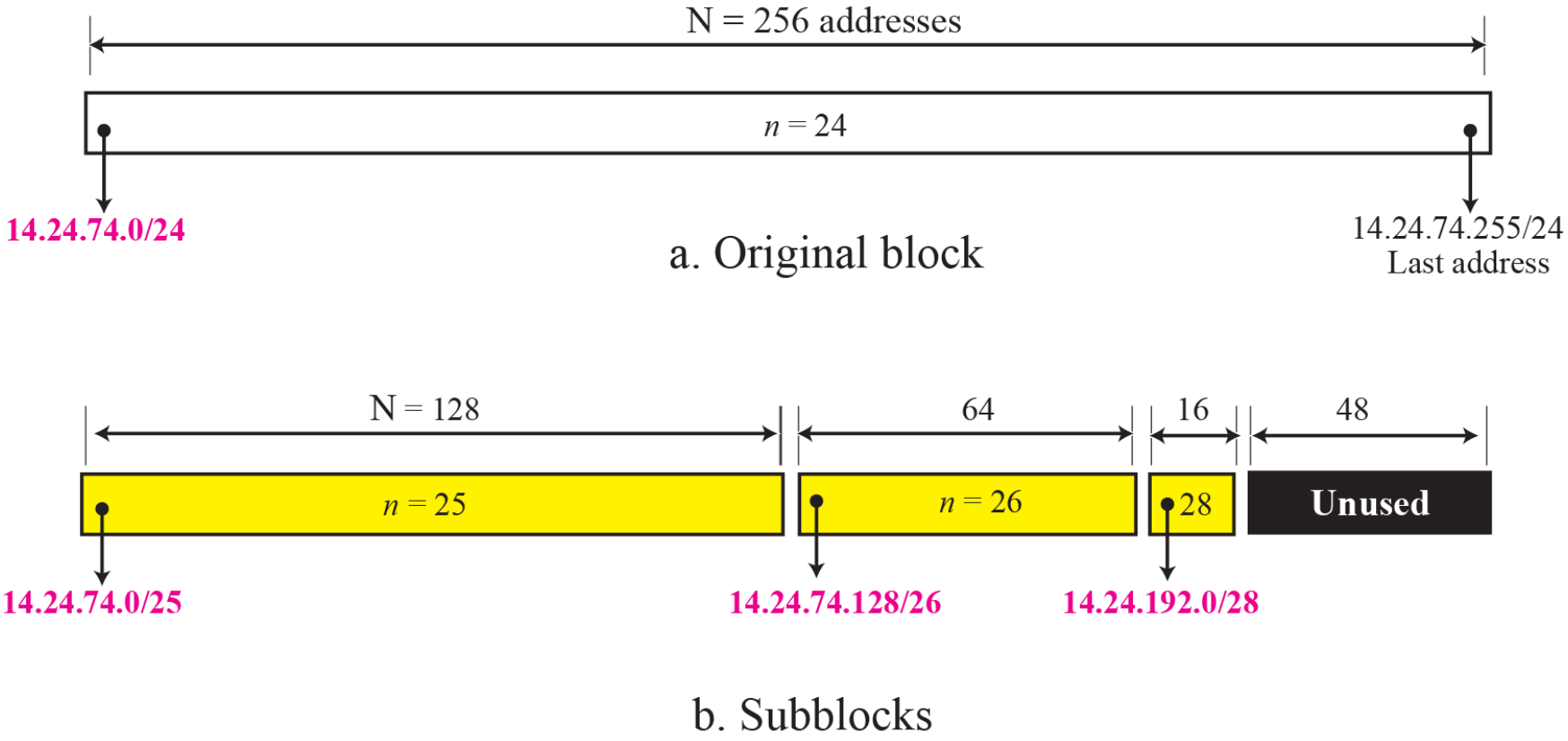
b. The number of addresses in the second subblock is not a power of 2 either. We allocate 64 addresses. The subnet mask is 26. The first address in this block is 14.24.74.128/26; the last address is 14.24.74.191/26.

c. The number of addresses in the third subblock is not a power of 2 either. We allocate 16 addresses. The subnet mask is 28. The first address in this block is 14.24.74.192/28; the last address is 14.24.74.207/28.

d. If we add all addresses in the previous subblocks, the result is 208 addresses, which means 48 addresses are left in reserve. The first address in this range is 14.24.74.209. The last address is 14.24.74.255.

e. Figure 5.31 shows the configuration of blocks. We have shown the first address in each block.

Figure 5.31 *Solution to Example 5.33*



Example 5.34

Assume a company has three offices: Central, East, and West. The Central office is connected to the East and West offices via private, WAN lines. The company is granted a block of 64 addresses with the beginning address 70.12.100.128/26. The management has decided to allocate 32 addresses for the Central office and divides the rest of addresses between the two other offices.

1. The number of addresses are assigned as follows:

$$\text{Central office } N_c = 32 \qquad \text{East office } N_e = 16 \qquad \text{West office } N_w = 16$$

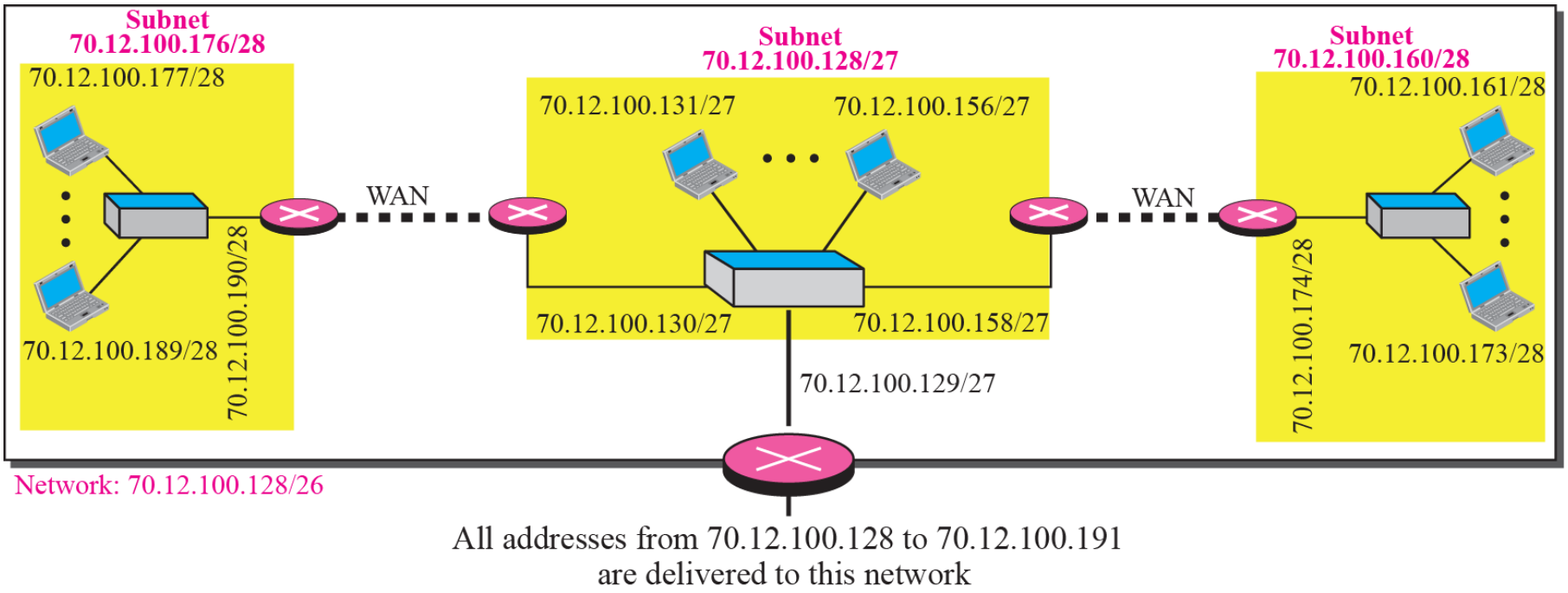
2. We can find the prefix length for each subnetwork:

$$n_c = n + \log_2(64/32) = 27 \qquad n_e = n + \log_2(64/16) = 28 \qquad n_w = n + \log_2(64/16) = 28$$

Example 5.34 Continued

3. Figure 5.32 shows the configuration designed by the management. The Central office uses addresses 70.12.100.128/27 to 70.12.100.159/27. The company has used three of these addresses for the routers and has reserved the last address in the subblock. The East office uses the addresses 70.12.100.160/28 to 70.12.100.175/28. One of these addresses is used for the router and the company has reserved the last address in the subblock. The West office uses the addresses 70.12.100.160/28 to 70.12.100.175/28. One of these addresses is used for the router and the company has reserved the last address in the subblock. The company uses no address for the point-to-point connections in WANs.

Figure 5.32 *Example 5.34*



Example 5.35

An ISP is granted a block of addresses starting with 190.100.0.0/16 (65,536 addresses). The ISP needs to distribute these addresses to three groups of customers as follows:

- The first group has 64 customers; each needs approximately 256 addresses.
- The second group has 128 customers; each needs approximately 128 addresses.
- The third group has 128 customers; each needs approximately 64 addresses.

We design the subblocks and find out how many addresses are still available after these allocations.

Example 5.35 Continued

Solution

Let us solve the problem in two steps. In the first step, we allocate a subblock of addresses to each group. The total number of addresses allocated to each group and the prefix length for each subblock can found as

Group 1: $64 \times 256 = 16{,}384$	$n_1 = 16 + \log_2 (65536/16384) = 18$
Group 2: $128 \times 128 = 16{,}384$	$n_2 = 16 + \log_2 (65536/16384) = 18$
Group 3: $128 \times 64 = 8192$	$n_3 = 16 + \log_2 (65536/8192) = 19$

Figure 5.33 shows the design for the first hierarchical level. Figure 5.34 shows the second level of the hierarchy. Note that we have used the first address for each customer as the subnet address and have reserved the last address as a special address.

Figure 5.33 *Solution to Example 5.35: first step*

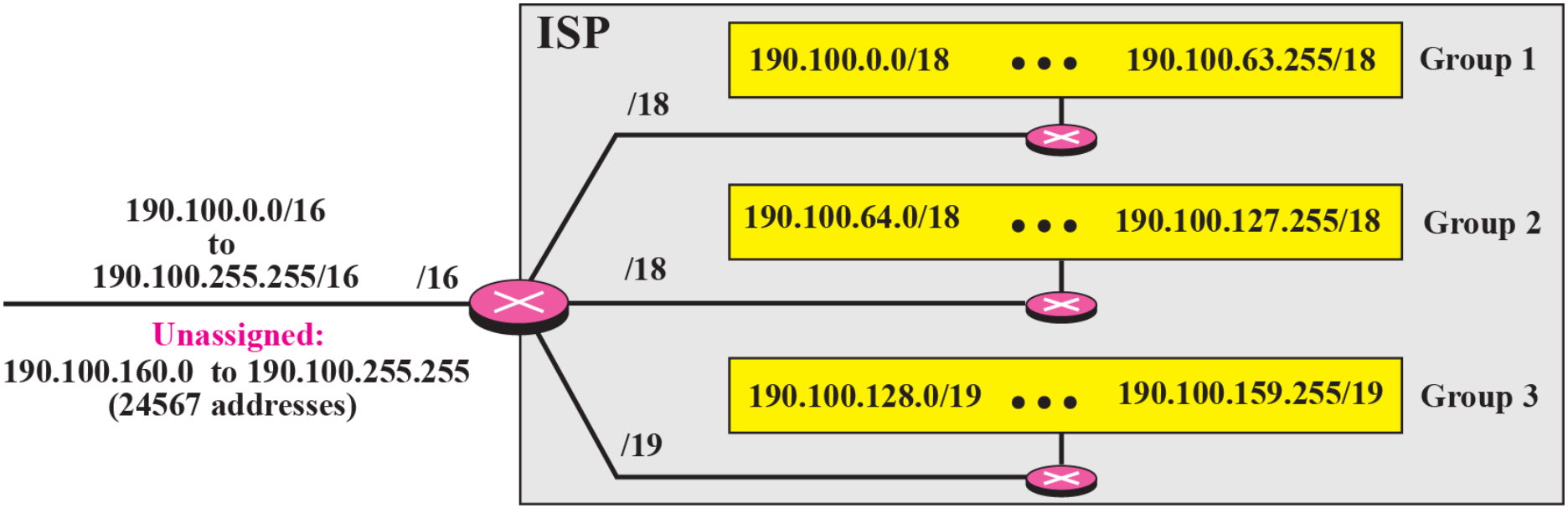


Figure 5.34 *Solution to Example 5.35: second step*

Group: $n = 18$

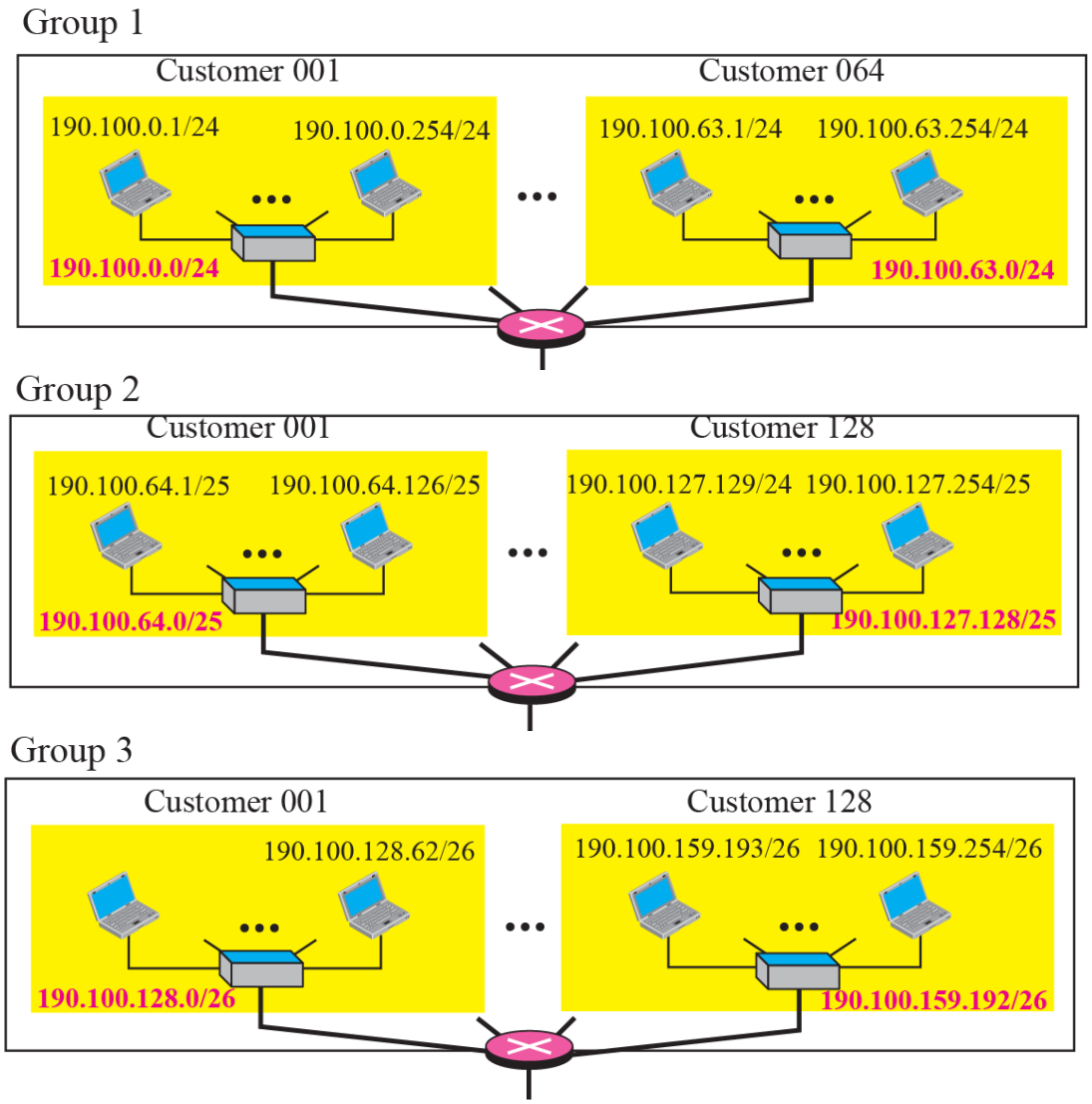
Subnet: $n = 18 + \log_2 (16385/256) = 24$

Group: $n = 18$

Subnet: $n = 18 + \log_2 (16385/128) = 25$

Group: $n = 19$

Subnet: $n = 19 + \log_2 (8192/64) = 26$



5-4 SPECIAL ADDRESSES

In classful addressing some addresses were reserved for special purposes. The classless addressing scheme inherits some of these special addresses from classful addressing.

Topics Discussed in the Section

- ✓ **Special Blocks**
- ✓ **Special Addresses in each Block**

Figure 5.35 *Example of using the all-zero address*

Special Blocks

Some blocks of addresses are reserved for special purposes.

All-Zeros Address

The block 0.0.0.0/32, which contains only one single address, is reserved for communication when a host needs to send an IPv4 packet but it does not know its own address.

This is normally used by a host at bootstrap time when it does not know its IPv4 address.

The host sends an IPv4 packet to a bootstrap server using this address as the source address and a **limited broadcast address** as the destination address to find its own address (see Figure 5.35).

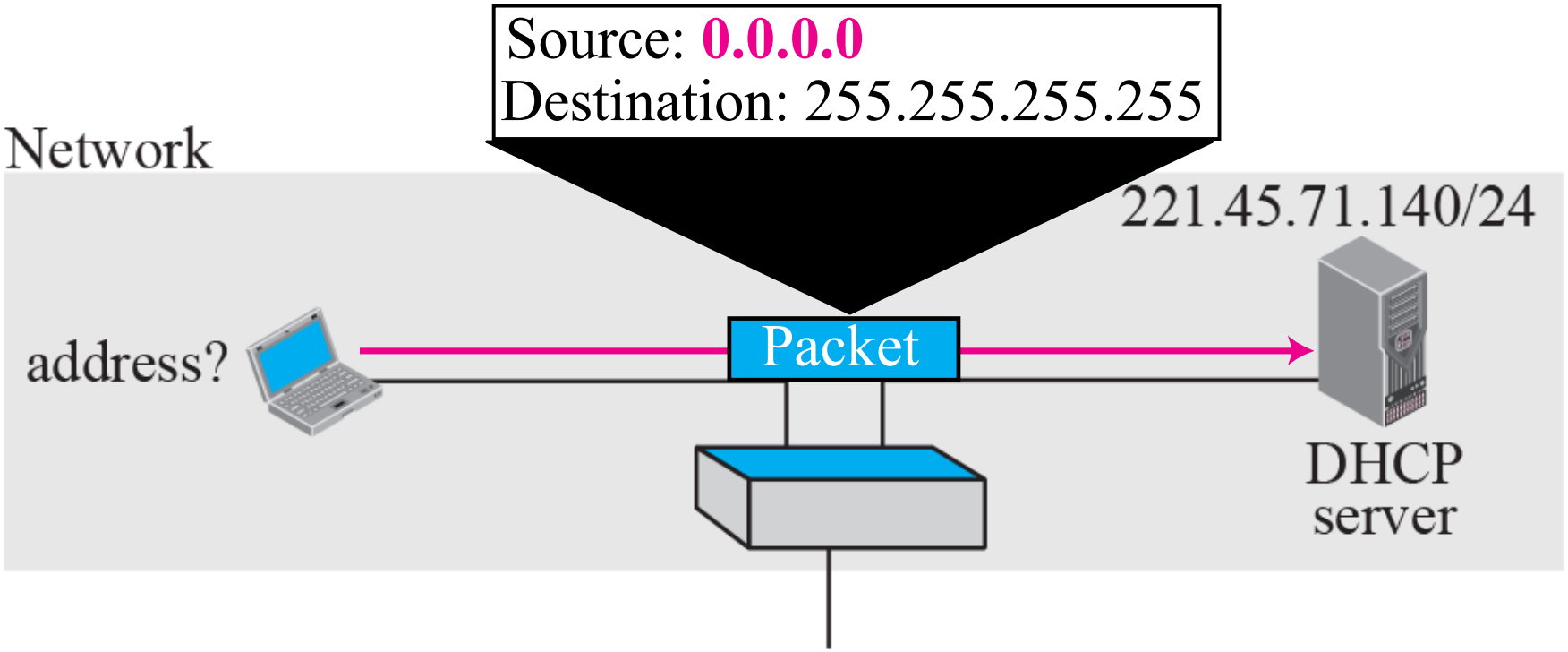
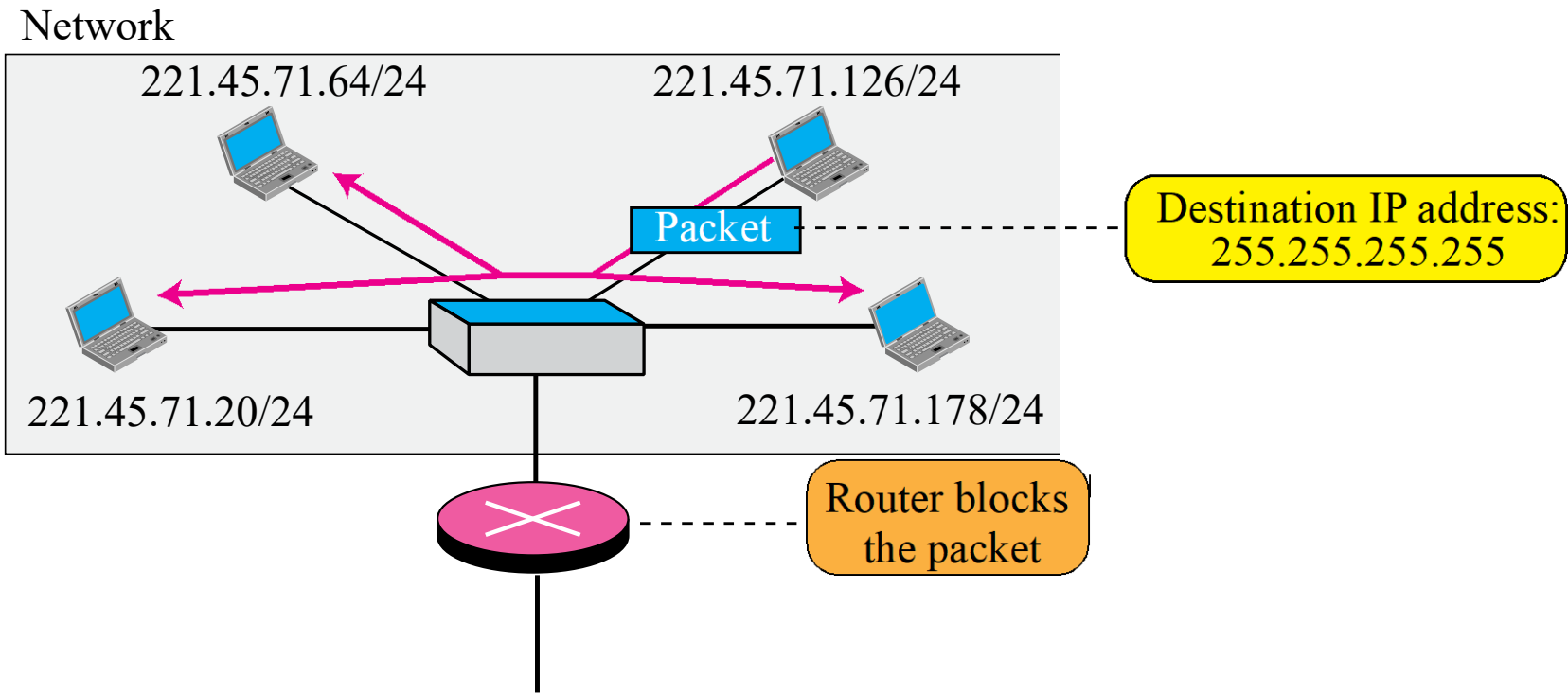


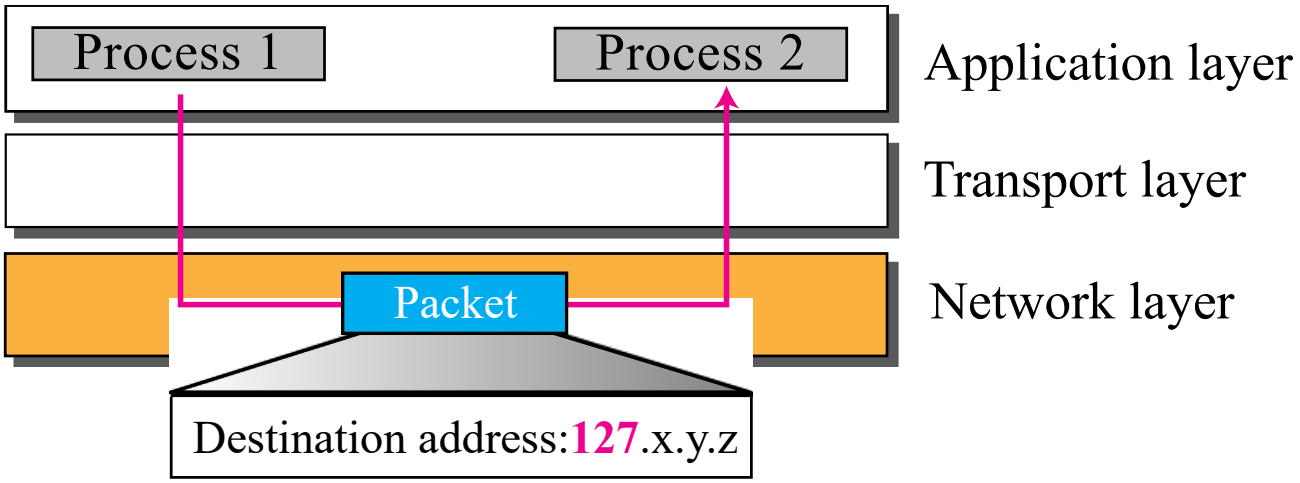
Figure 5.36 *Example of limited broadcast address*



All-Ones Address: Limited Broadcast Address

The block 255.255.255.255/32, which contains one single address, is reserved for limited broadcast address in the current network. A host that wants to send a message to every other host can use this address as a destination address in an IPv4 packet. However, a router will block a packet having this type of address to confine the broadcasting to the local network. In Figure 5.36, a host sends a datagram using a destination IPv4 address consisting of all 1s. All devices on this network receive and process this datagram.

Figure 5.37 *Example of loopback address*



Loopback Addresses

The block 127.0.0.0/8 is used for the **loopback address,** which is an address used to test the software on a machine. When this address is used, a packet never leaves the machine; it simply returns to the protocol software. It can be used to test the IPv4 software.

For example, an application such as "ping" can send a packet with a loopback address as the destination address to see if the IPv4 software is capable of receiving and processing a packet. As another example, the loopback address can be used by a *client process* (a running application program) to send a message to a server process on the same machine. Note that this can be used only as a destination address in an IPv4 packet (see Figure 5.37).

Private Addresses

A number of blocks are assigned for private use. They are not recognized globally. These blocks are depicted in Table 5.2. These addresses are used either in isolation or in connection with network address translation techniques .

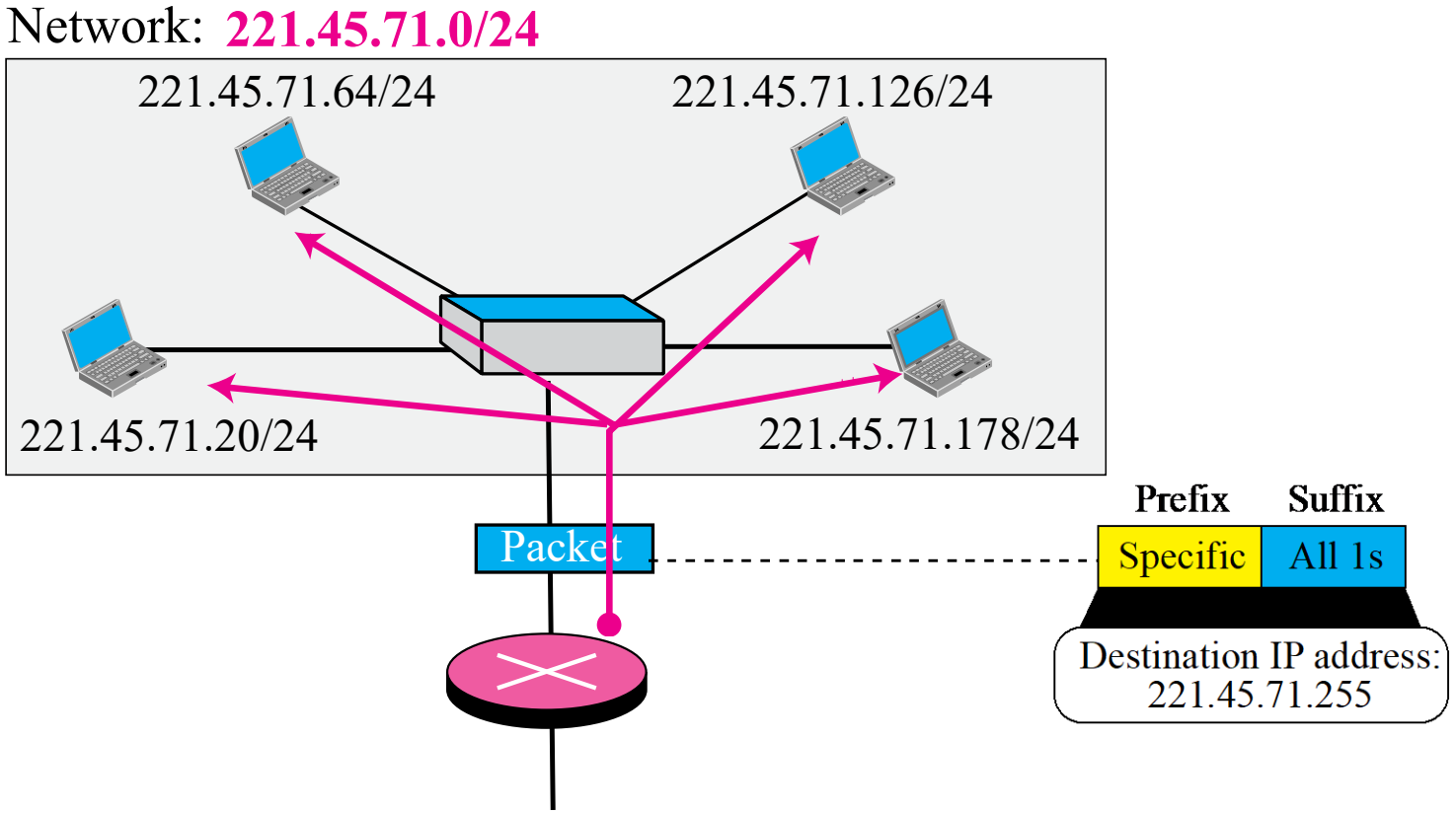
Multicast Addresses

The block 224.0.0.0/4 is reserved for multicast communication.

Table 5.2 *Addresses for private networks*

Block	*Number of addresses*	*Block*	*Number of addresses*
10.0.0.0/8	16,777,216	192.168.0.0/16	65,536
172.16.0.0/12	1,047,584	169.254.0.0/16	65,536

Figure 5.38 *Example of a directed broadcast address*



Direct Broadcast Address

The last address in a block or subblock (with the suffix set all to 1s) can be used as a **direct broadcast address.** This address is usually used by a router to send a packet to all hosts in a specific network. All hosts will accept a packet having this type of destination address. Note that this address can be used only as a destination address in an IPv4 packet. In Figure 5.38, the router sends a datagram using a destination IPv4 address with a suffix of all 1s. All devices on this network receive and process the datagram.

5-5 NAT

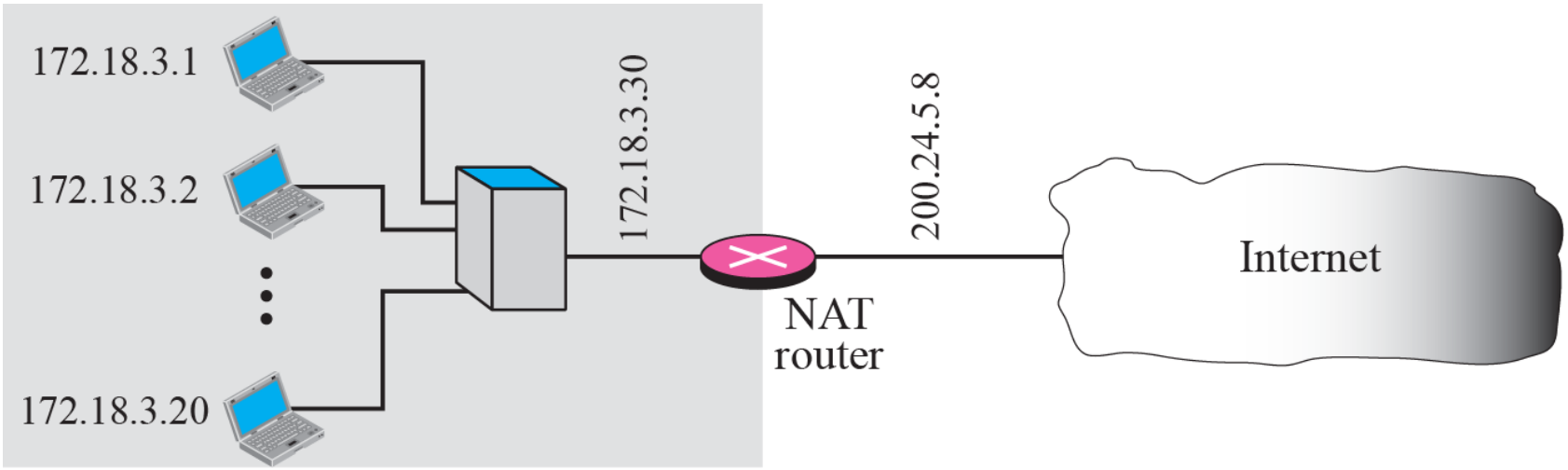
The distribution of addresses through ISPs has created a new problem. If the business grows or the household needs a larger range, the ISP may not be able to grant the demand because the addresses before and after the range may have already been allocated to other networks. In most situations, however, only a portion of computers in a small network need access to the Internet simultaneously. A technology that can help in this cases is *network address translation (NAT)*.

Topics Discussed in the Section

- ✓ **Address Translation**
- ✓ **Translation Table**

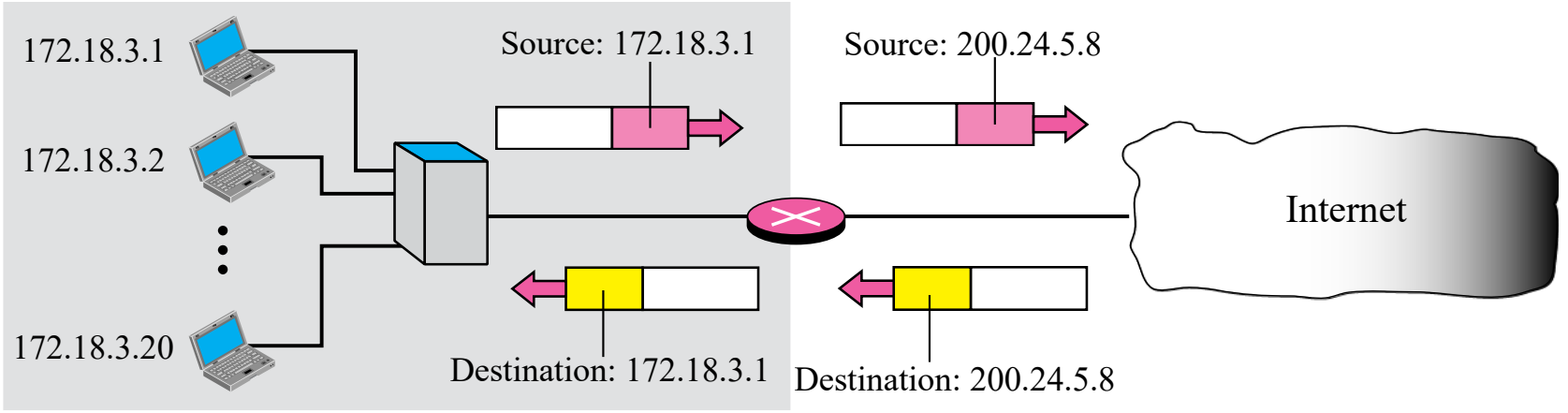
Figure 5.39 *NAT*

As the figure shows, the private network uses private addresses. The router that connects the network to the global address uses one private address and one global address. The private network is transparent to the rest of the Internet; the rest of the Internet sees only the NAT router with the address 200.24.5.8.



Address Translation

All of the outgoing packets go through the NAT router, which replaces the *source address* in the packet with the global NAT address. All incoming packets also pass through the NAT router, which replaces the *destination address* in the packet (the NAT router global address) with the appropriate private address. Figure 5.40 shows an example of address translation.



Address Translation

All of the outgoing packets go through the NAT router, which replaces the *source address* in the packet with the global NAT address. All incoming packets also pass through the NAT router, which replaces the *destination address* in the packet (the NAT router global address) with the appropriate private address. Figure 5.40 shows an example of address translation.

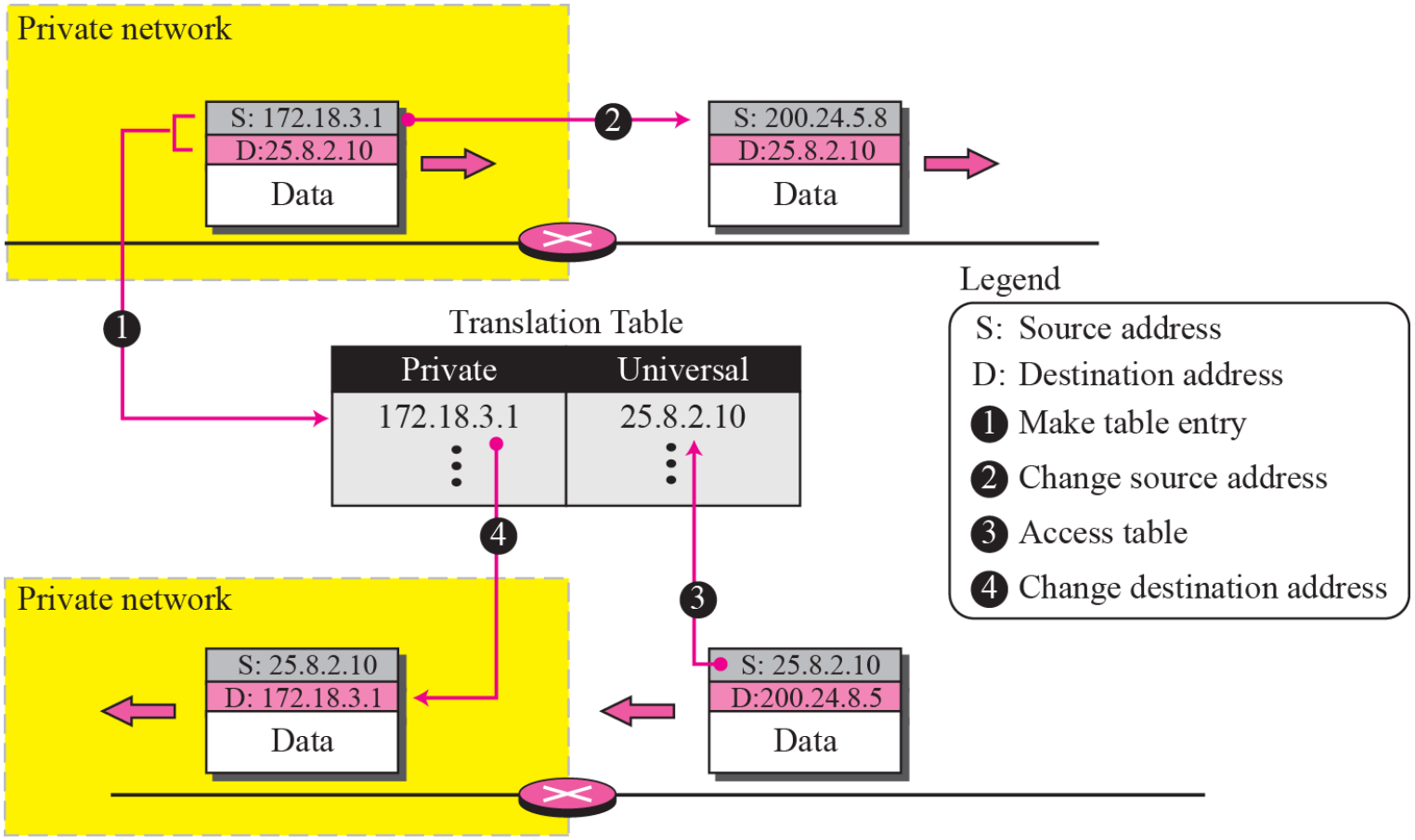
Translation Table

The reader may have noticed that translating the source addresses for an outgoing packet is straightforward. But how does the NAT router know the destination address for a packet coming from the Internet? There may be tens or hundreds of private IP addresses, each belonging to one specific host. The problem is solved if the NAT router has a translation table.

Using One IP Address

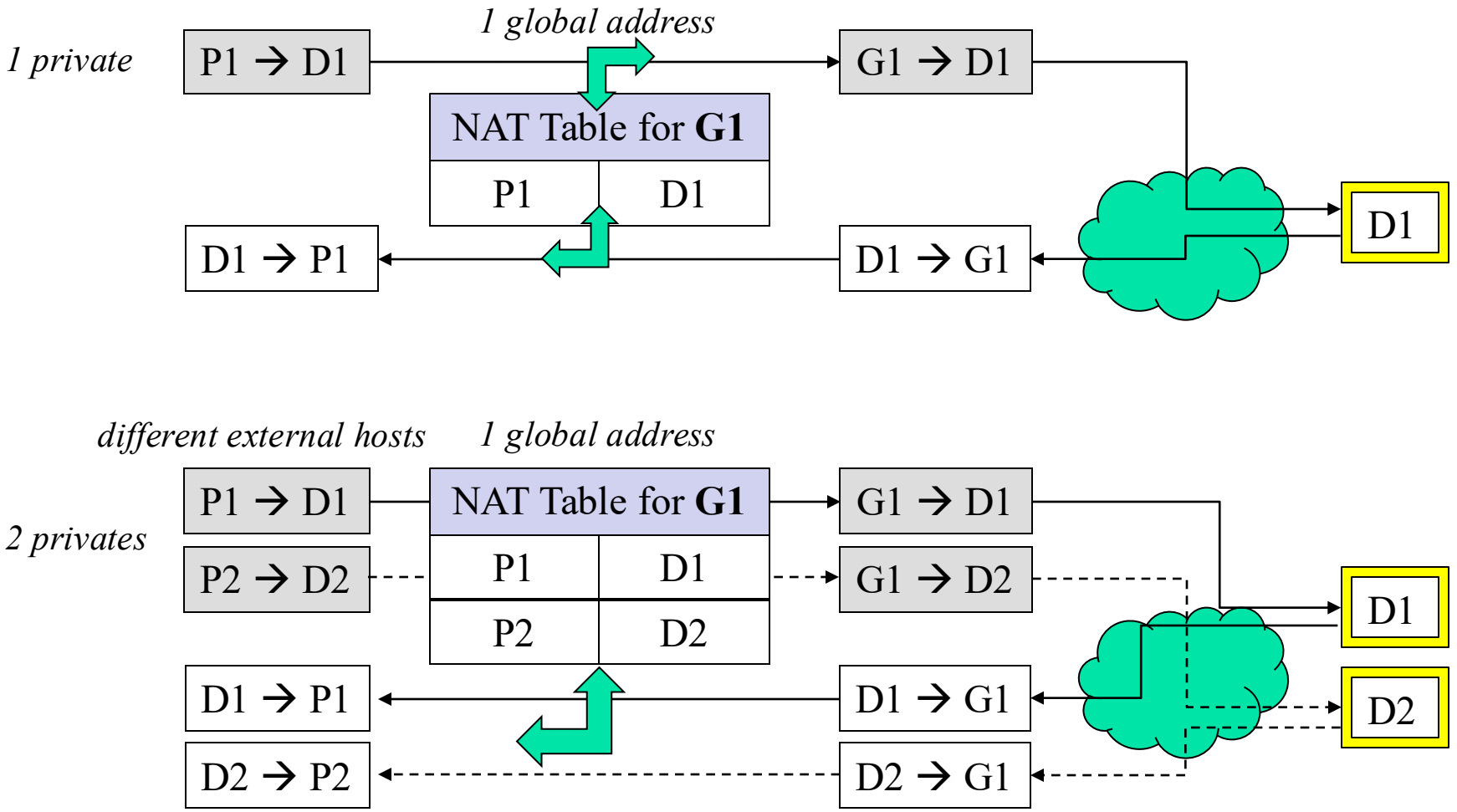
In its simplest form, a translation table has only two columns: the private address and the external address (destination address of the packet). When the router translates the source address of the outgoing packet, it also makes note of the destination address— where the packet is going. When the response comes back from the destination, the router uses the source address of the packet (as the external address) to find the private address of the packet. Figure 5.41 shows the idea. for a packet coming from the Internet? There may be tens or hundreds of private IP

Figure 5.41 *Translation*

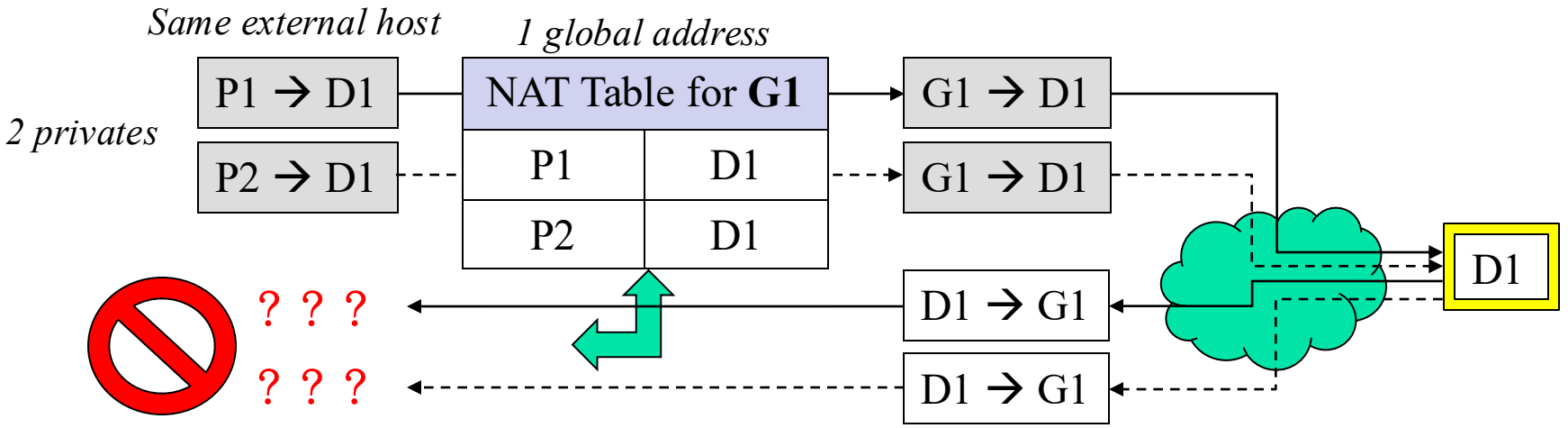


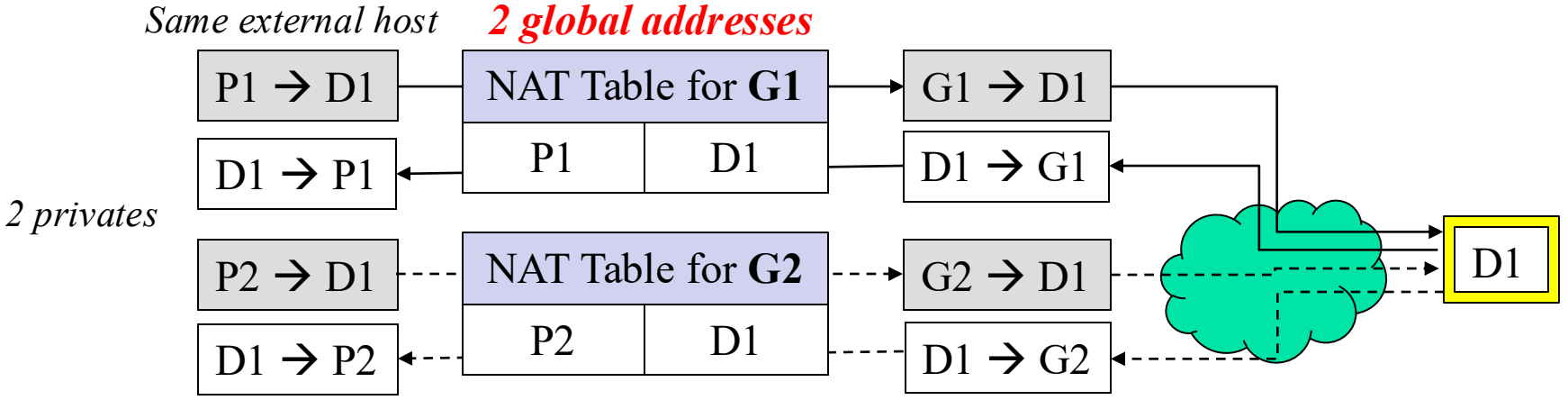
In this strategy, communication must always be initiated by the private network.
The NAT mechanism described requires that the private network start the communication.
As we will see, NAT is used mostly by ISPs that assign one single address to a customer. The customer, however, may be a member of a private network that has many private addresses. In this case, communication with the Internet is always initiated from the customer site, using a client program such as HTTP, TELNET, or FTP to access the corresponding server program. For example, when e-mail that originates from a noncustomer site is received by the ISP e-mail server, it is stored in the mailbox of the customer until retrieved with a protocol such as POP.
A private network cannot run a server program for clients outside of its network if it is using NAT technology.

NAT Table with only IP addresses (1)



NAT Table with only IP addresses (2)

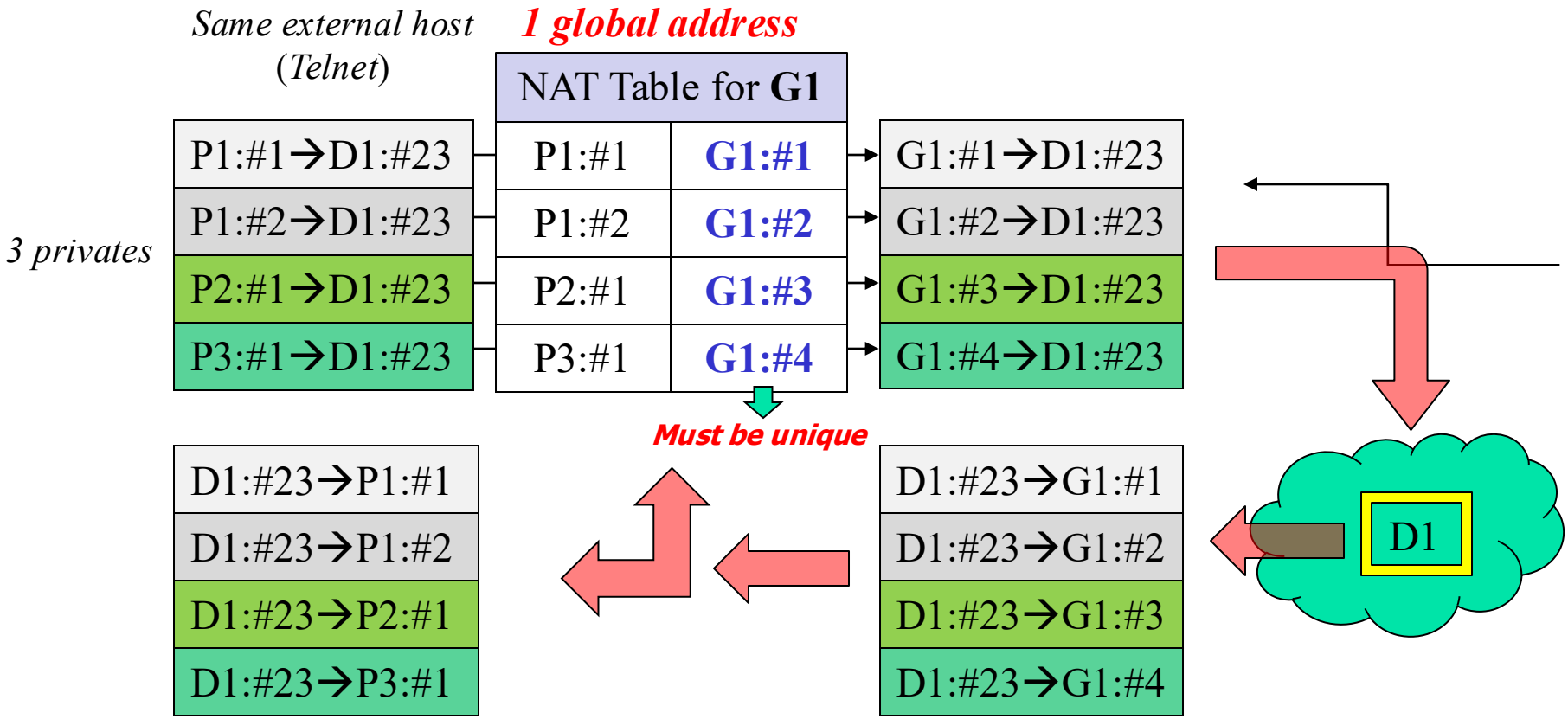




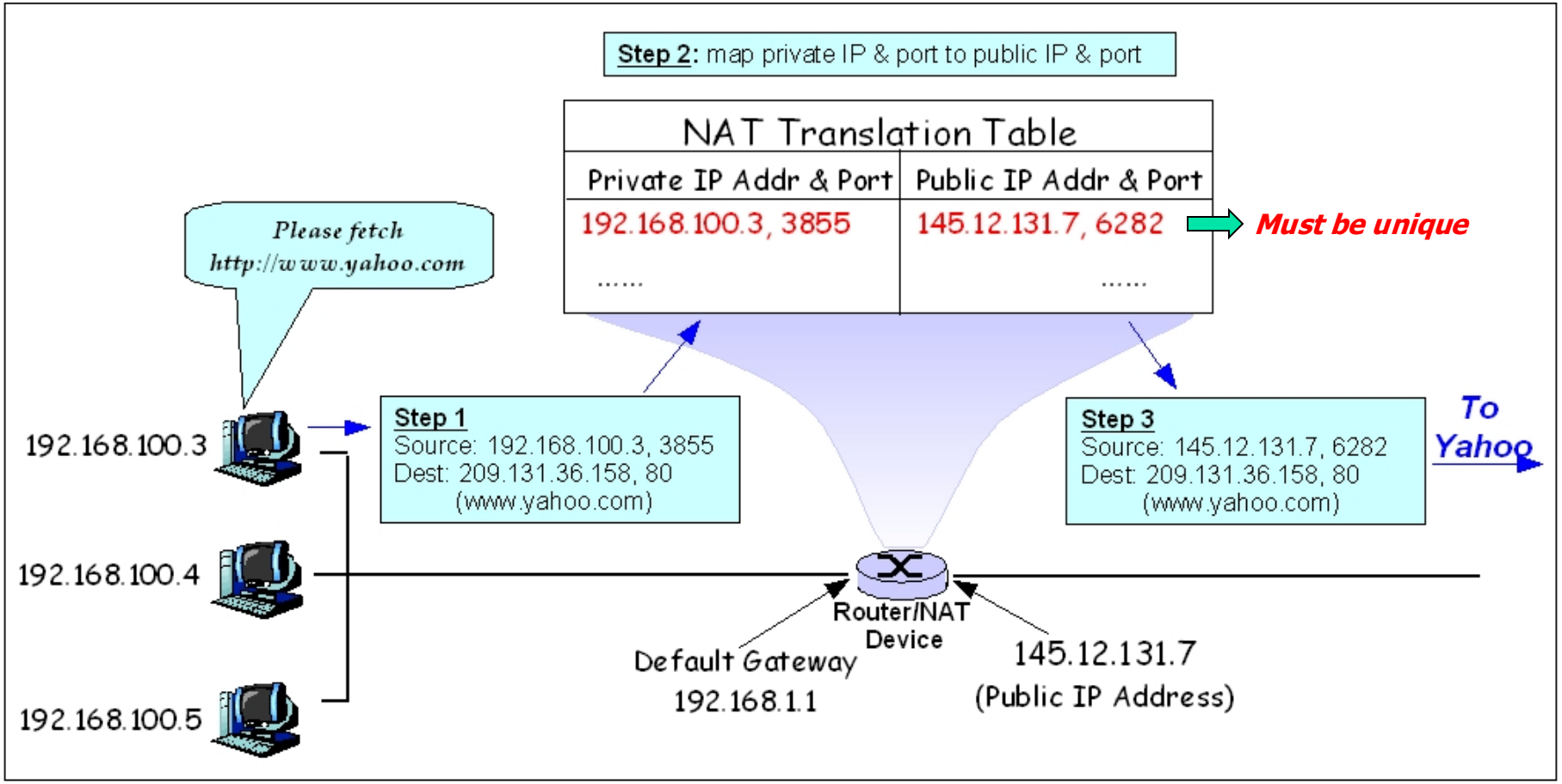
NAT Table with only IP addresses (3)

- If using only one global address
  - Only one private-network host to access the same external host
- If using a pool of global addresses (e.g. 4 addr)
  - No more than 4 connections can be made to the same destination
  - No private-network host can access two external server programs (e.g. HTTP and TELNET) at the same external host at the same time **???**
  - Two private-network hosts cannot access the same external server program at the same time (by using the same global address)

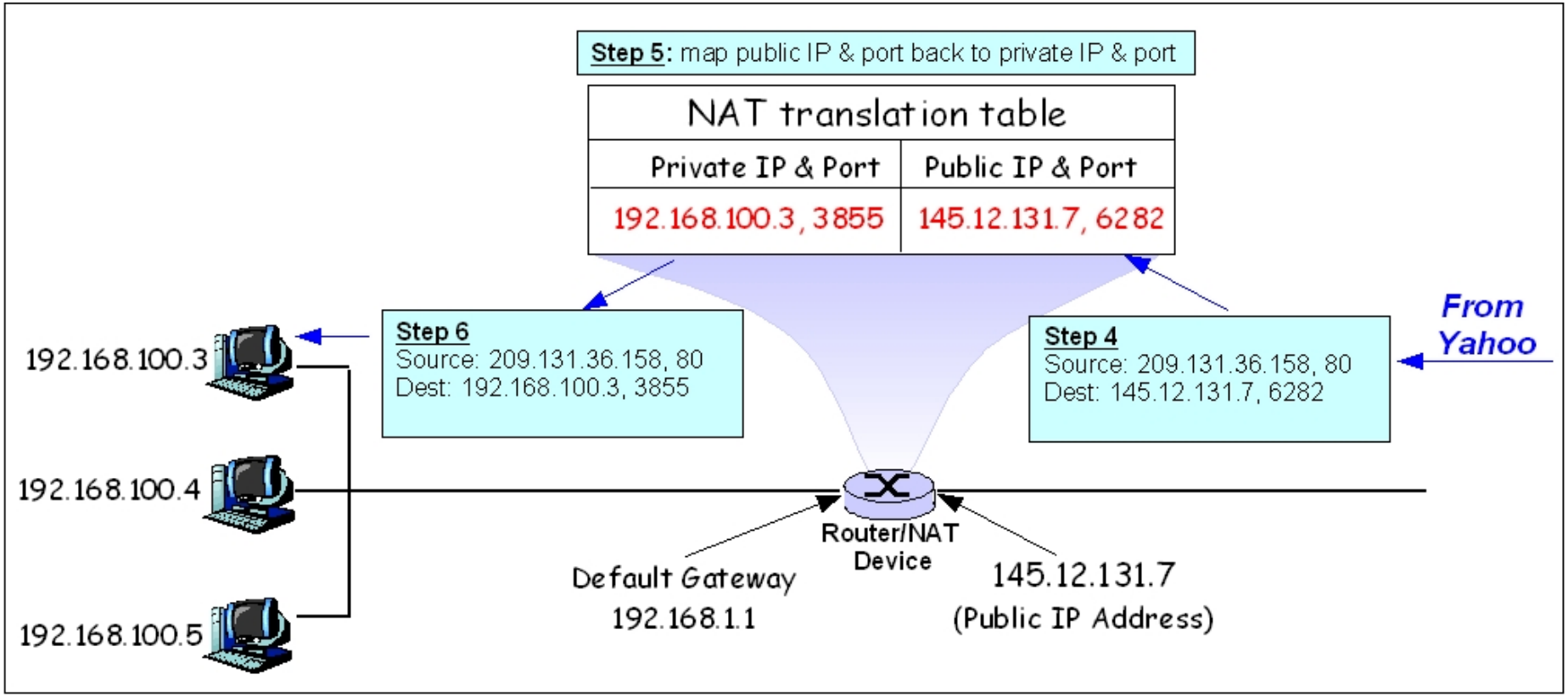
NAT Table with IP address & Port # (1)



NAT Table with IP address & Port # (2)



NAT Table with IP address & Port # (3)



NAT Table with IP address & Port # (4)

- NAT Table in the textbook
  - This may not be the best way to understand NAT

Server

Private Address	*Private Port*	*External Address*	*External Port*	*Transport Protocol*
172.18.3.1	1400	25.8.3.2	80	TCP
172.18.3.2	1401	25.8.3.2	80	TCP
…	…	…	…	…

Must be unique

Otherwise reassign a new private port and cache the mapping (the new port --> the old port)