IPv4 Addresses

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.

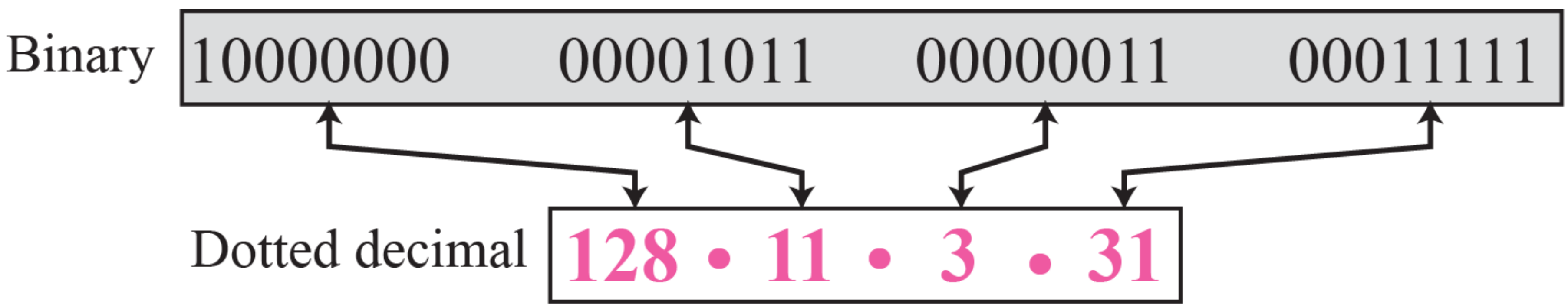
Note

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

Note

Numbers in base 2, 16, and 256 are discussed in Appendix B.

Figure 5.1 *Dotted-decimal notation*



Example 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 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 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 4

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

a. 1000 0001 0000 1011 0000 1011 1110 1111

b. 1100 0001 1000 0011 0001 1011 1111 1111

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. 0X 8 1 0 B 0 B E F or $810B0BEF_{16}$

b. 0X C 1 8 3 1 B F F or $C1831BFF_{16}$

Example 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 6

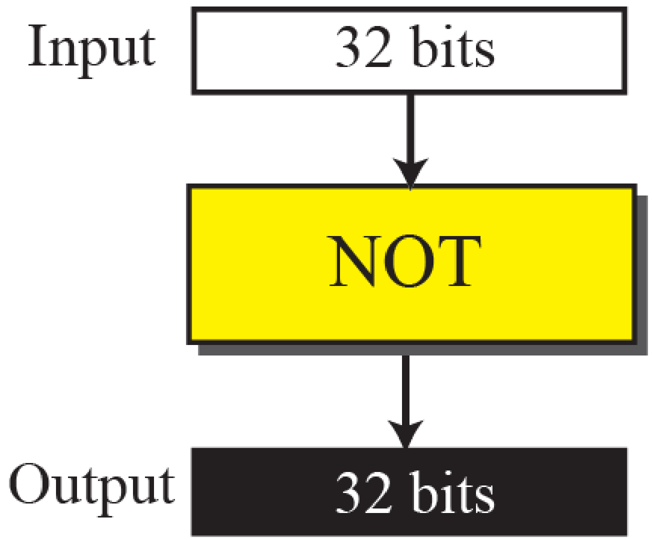
The first address in a range of addresses is 14.11.45.96. If the number of addresses in the range is 32, what is the last address?

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*



NOT operation

Input	Output
0	1
1	0

Operation for each bit

Example 7

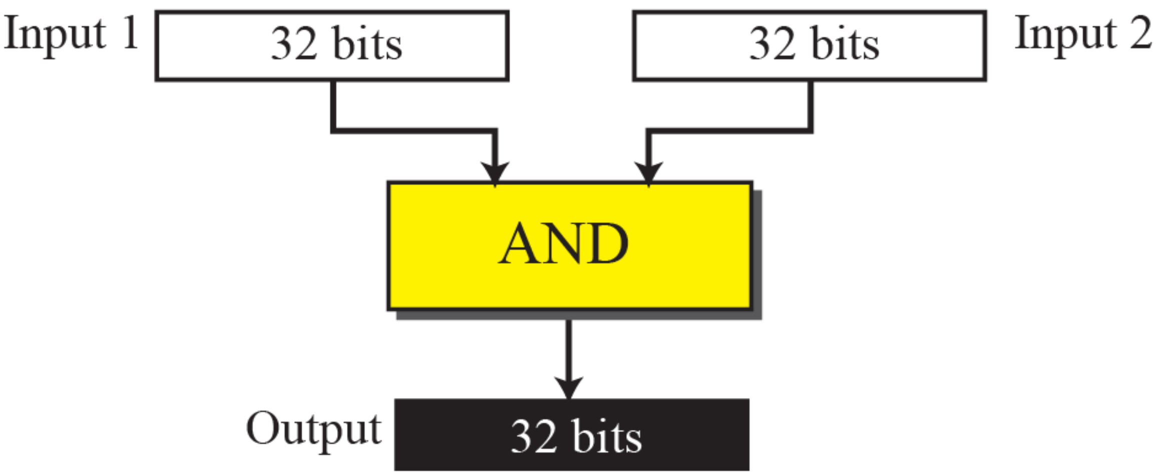
The following shows how we can apply the NOT operation on a 32-bit number in binary.

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 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 8

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

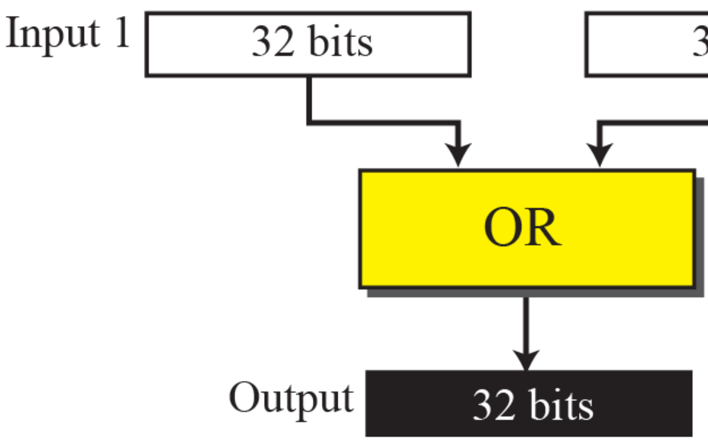
We can use the same operation using the dotted-decimal representation and the short cut.

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 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 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.

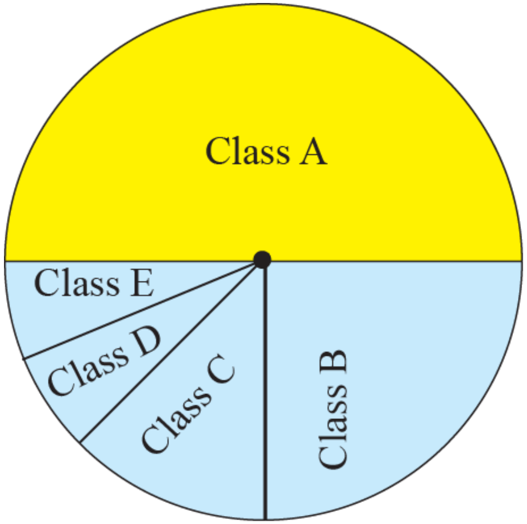
CLASSFUL ADDRESSING

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 *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 6 *Finding the class of address*

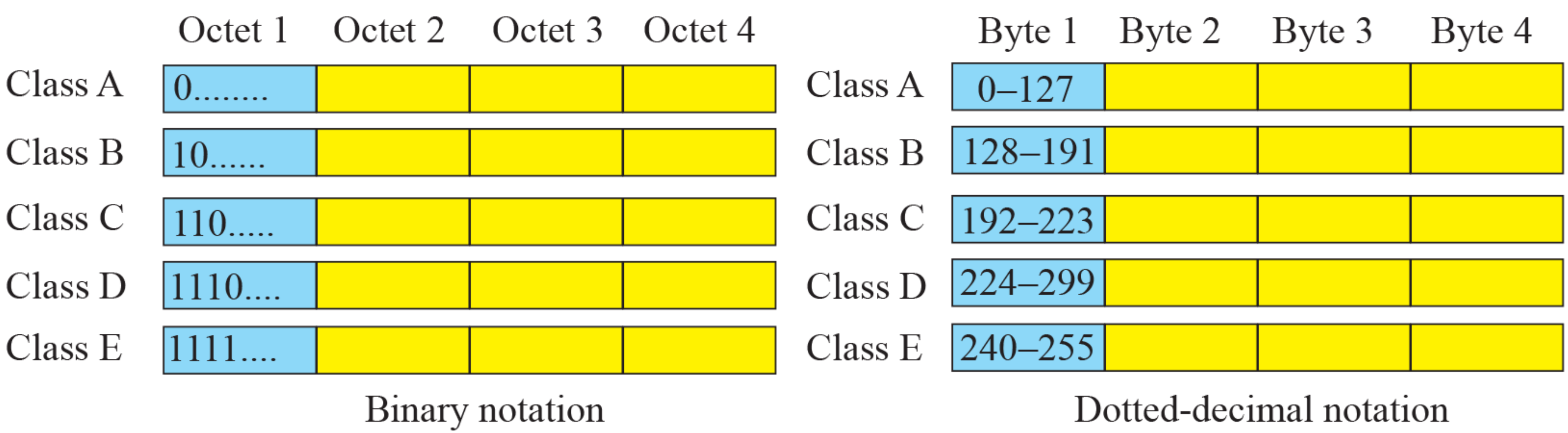
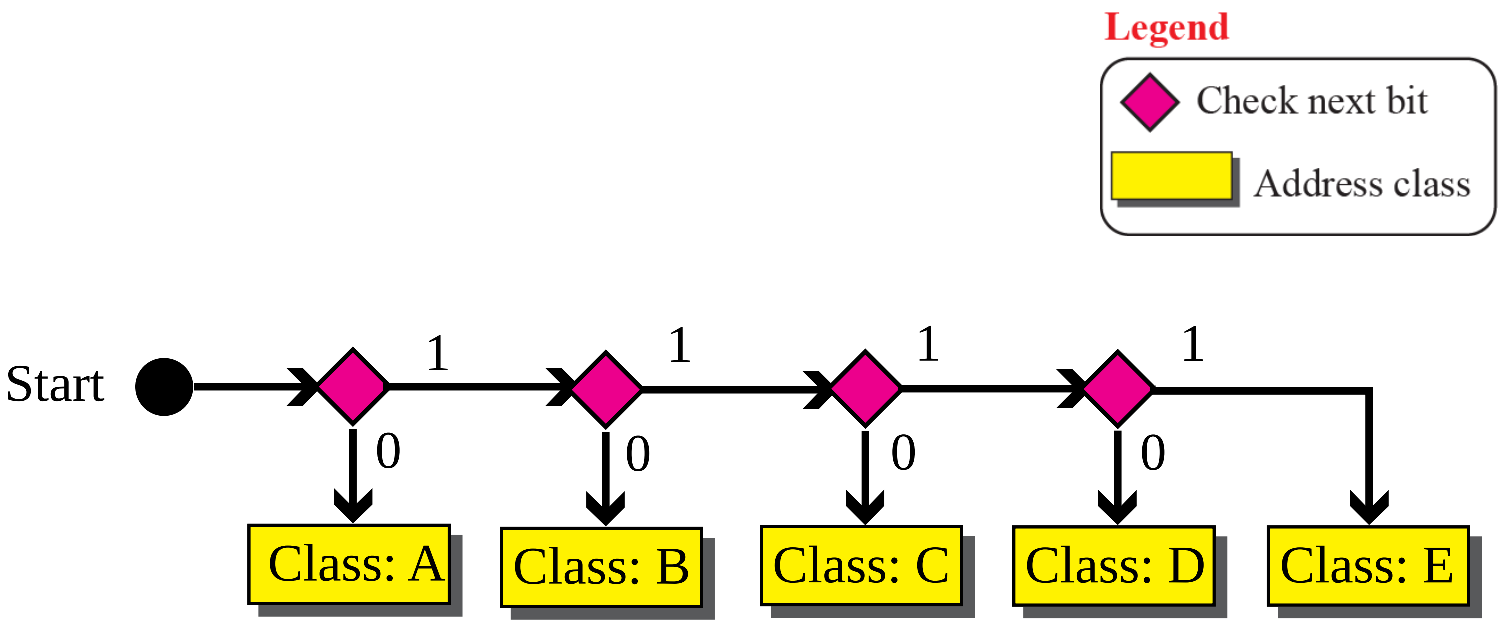


Figure 7 *Finding the class of an address using continuous checking*

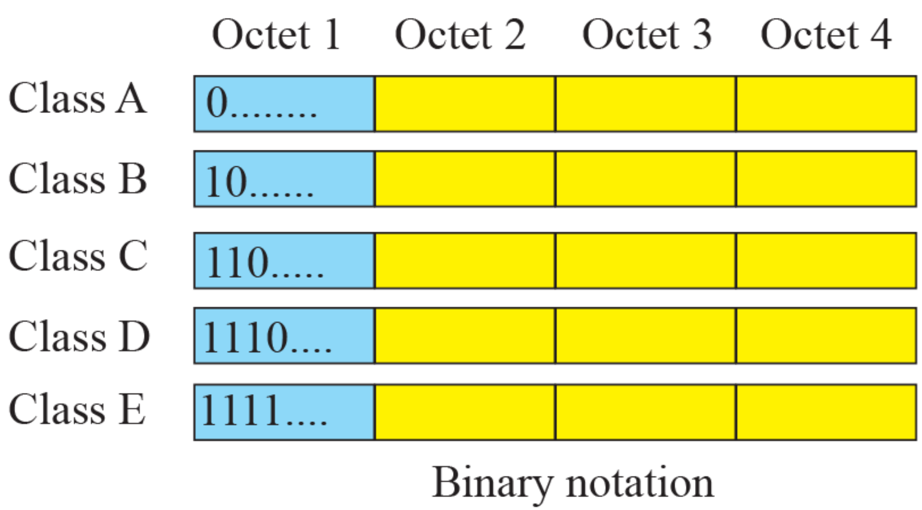


Example 10

Find the class of each address:

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

Solution



Example 11

Find the class of each address:

a. 227.12.14.87 -D

b. 193.14.56.22 -C

c. 14.23.120.8 -A

d. 252.5.15.111 -E

	Byte 1	Byte 2	Byte 3	Byte 4
Class A	0–127			
Class B	128–191			
Class C	192–223			
Class D	224–299			
Class E	240–255			

Dotted-decimal notation

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 8 *Netid and hostid*

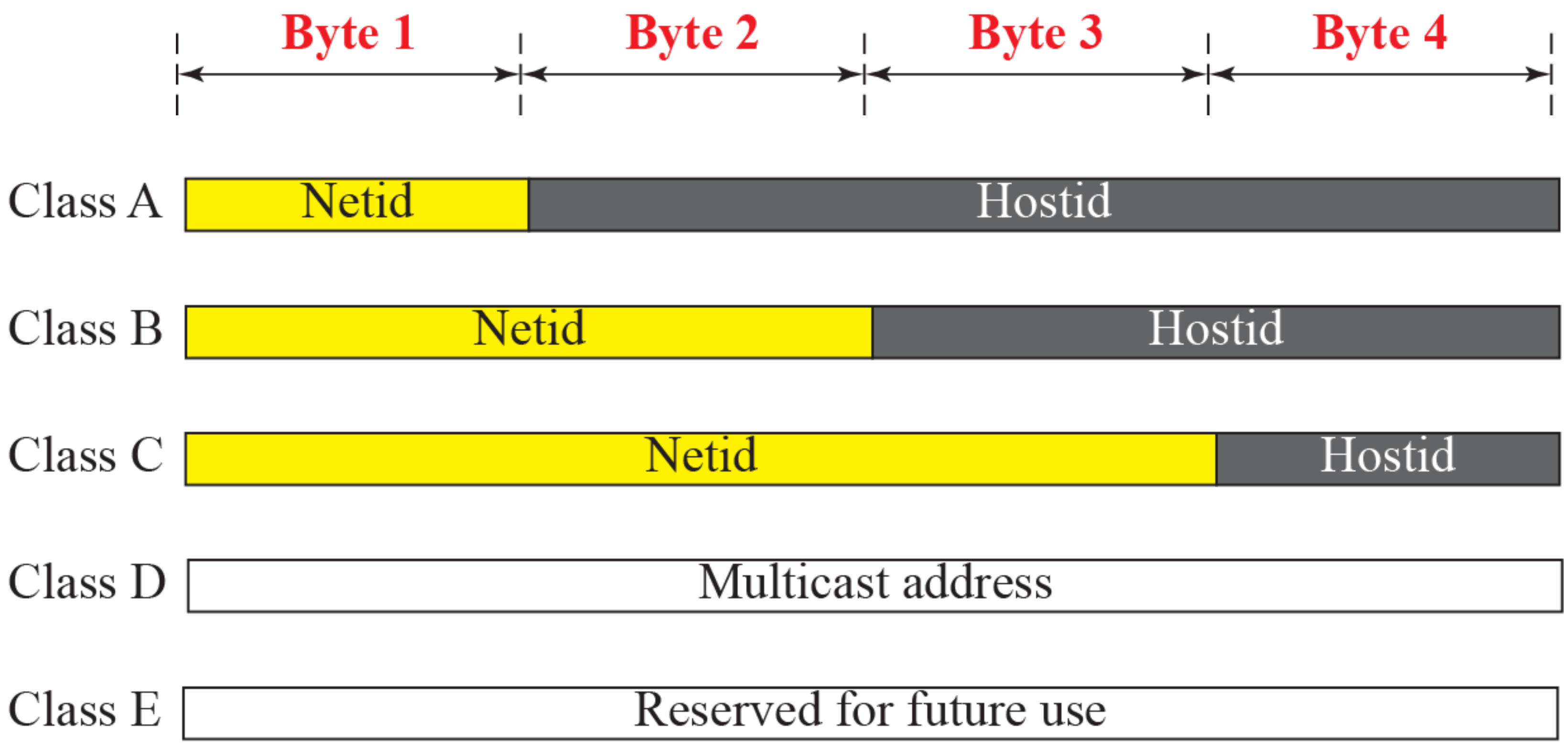
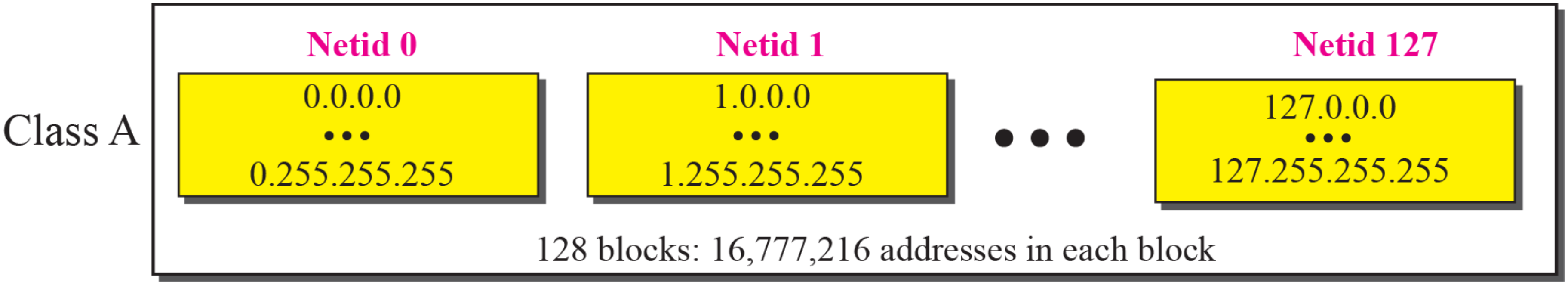


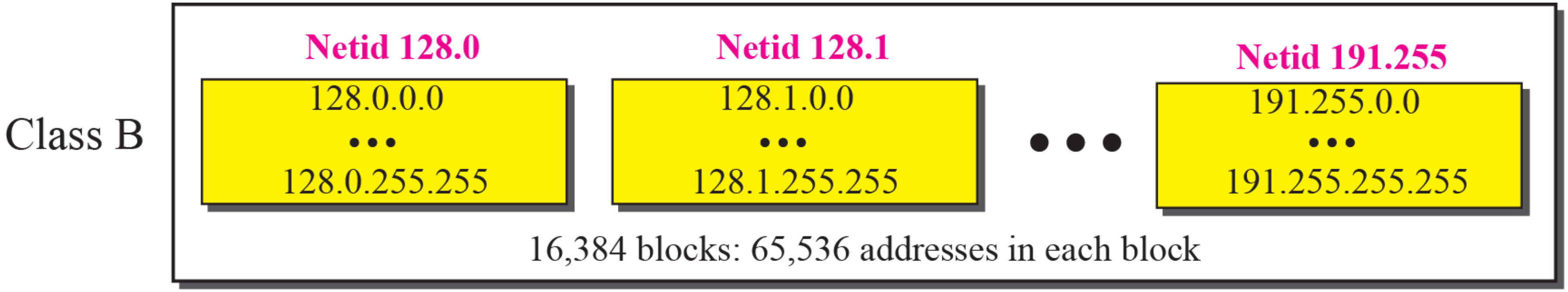
Figure 9 *Blocks in Class A*



Note

Millions of class A addresses are wasted.

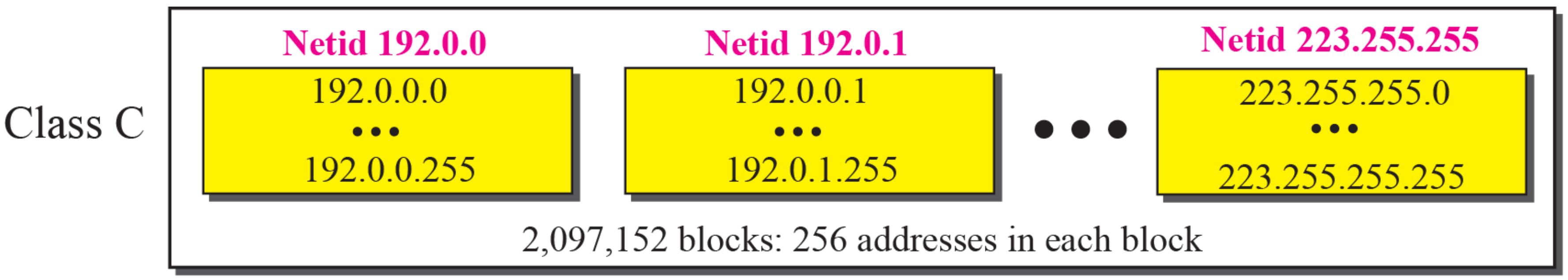
Figure 10 *Blocks in Class B*



Note

Many class B addresses are wasted.

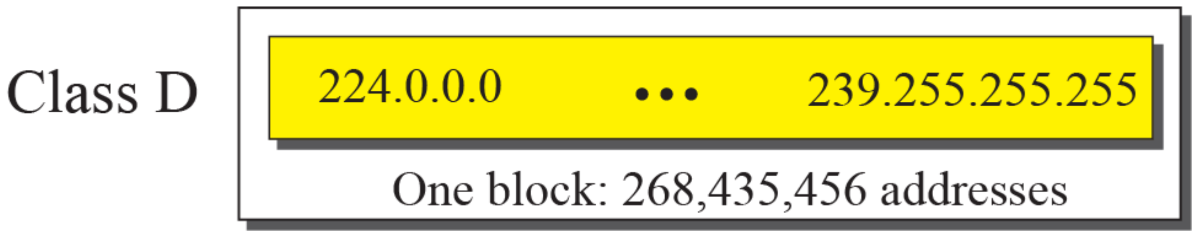
Figure 11 *Blocks in Class C*



Note

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

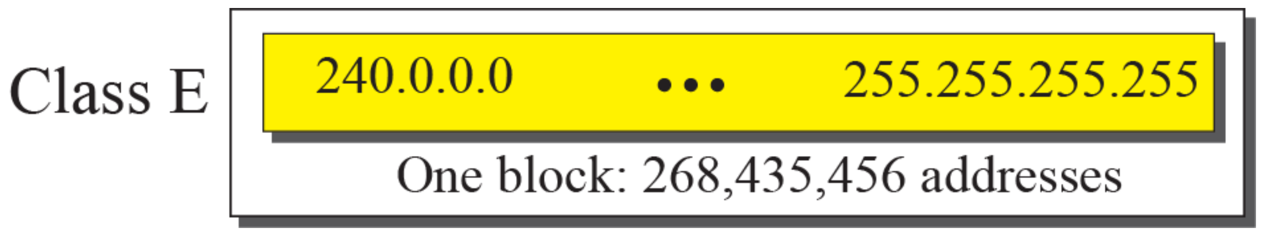
Figure 12 *The single block in Class D*



Note

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

Figure 13 *The single block in Class E*



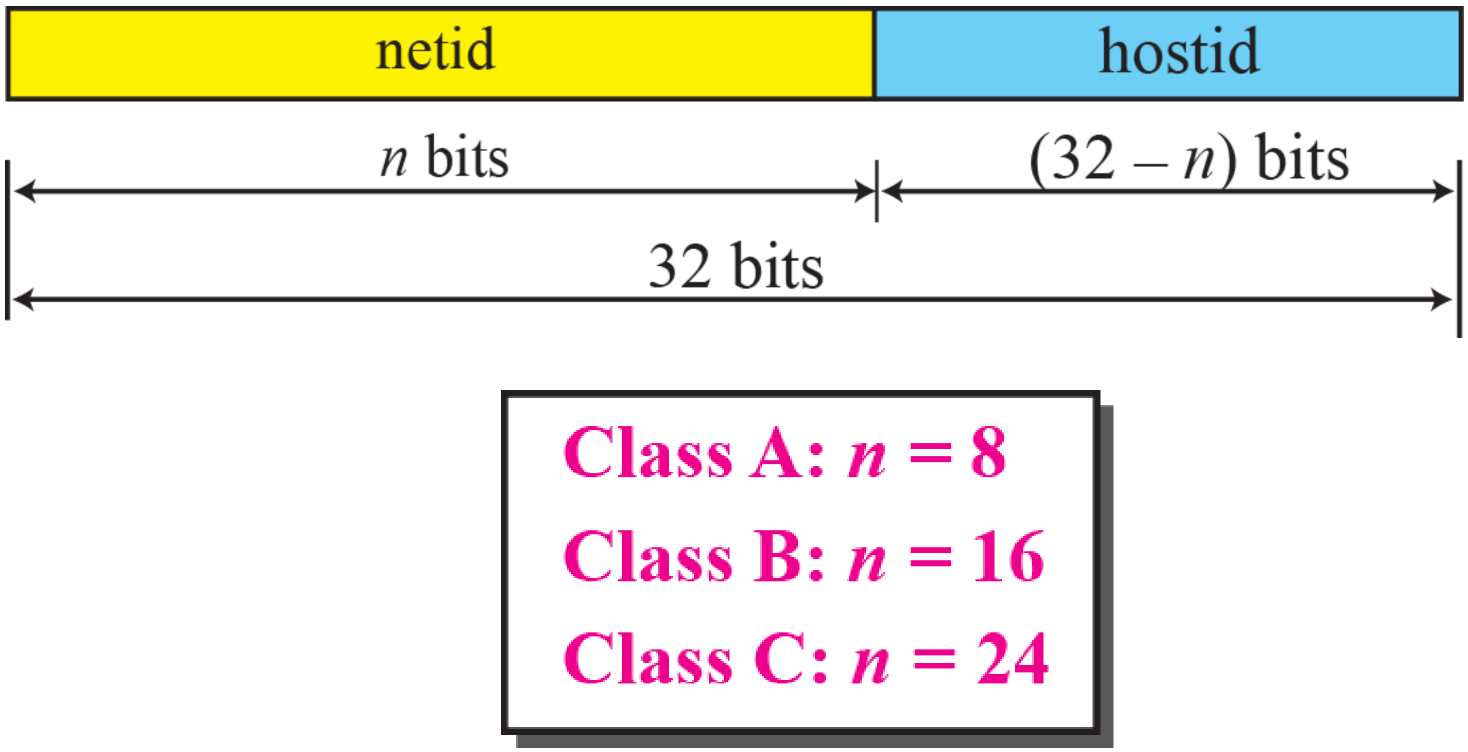
Note

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 14 *Two-level addressing in classful addressing*



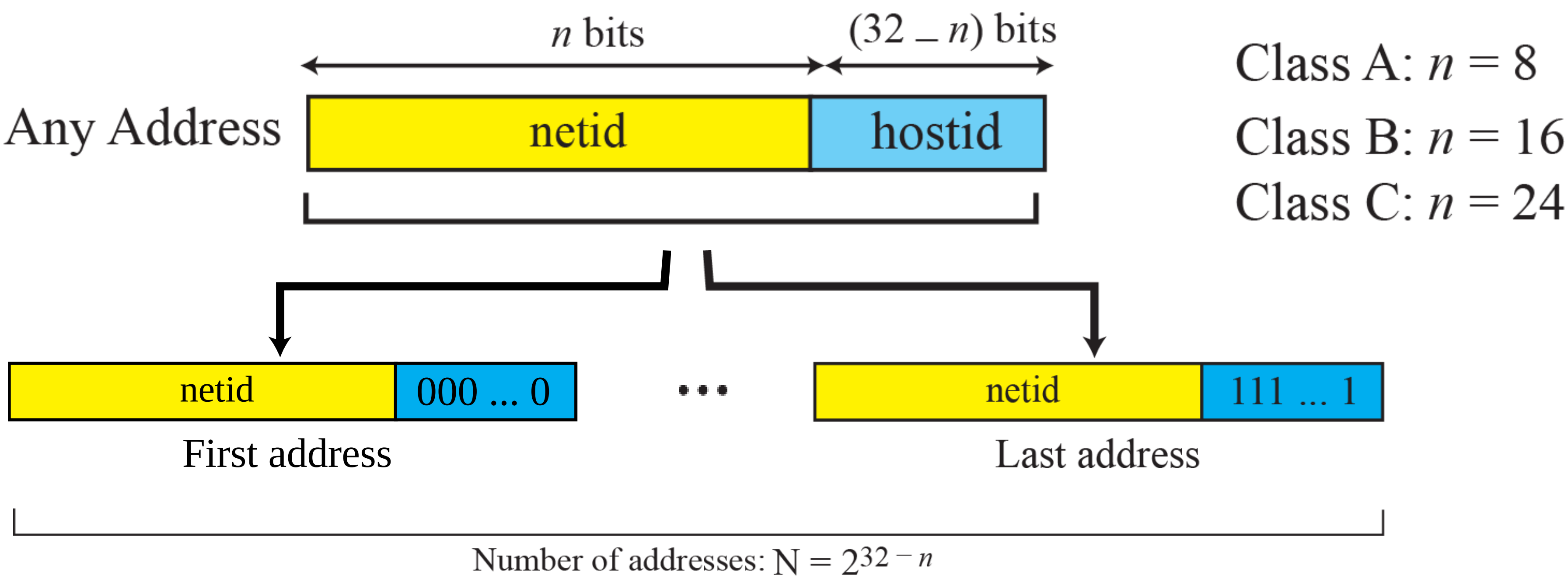
Example 12

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 15 *Information extraction in classful addressing*



Example 13

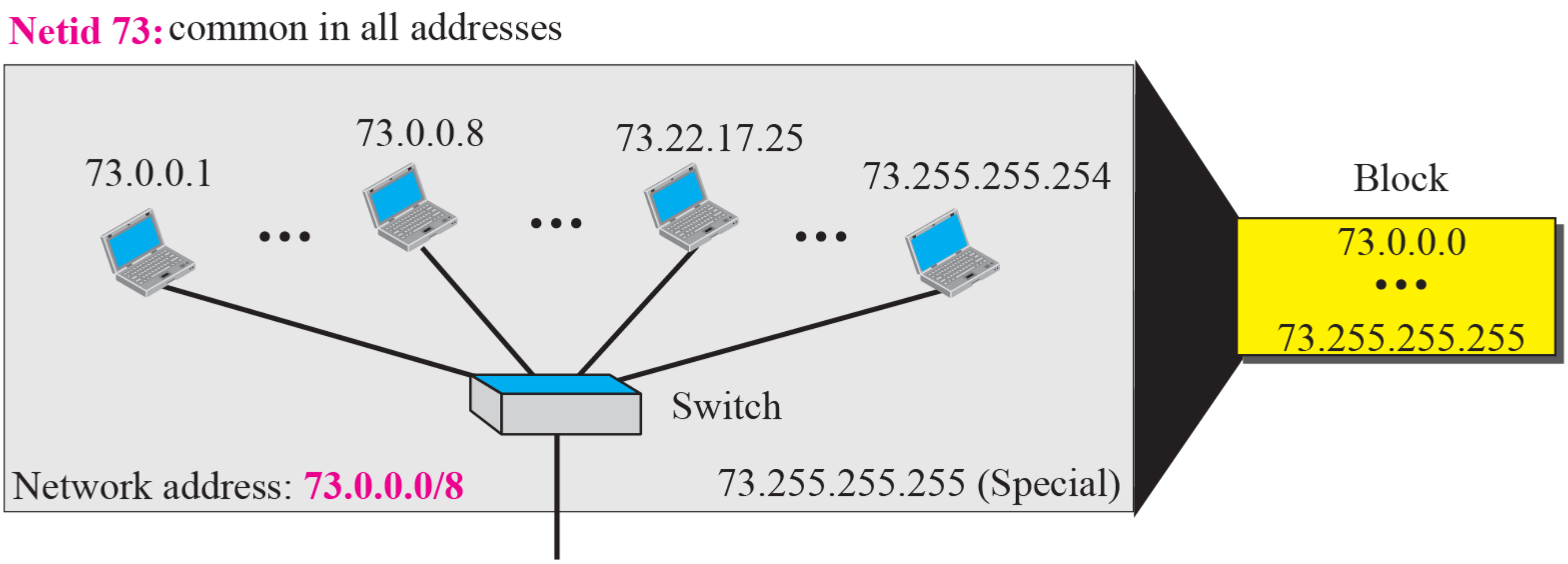
An address in a block is given as 73.22.17.25. Find the number of addresses in the block, the first address, and the last address.

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.

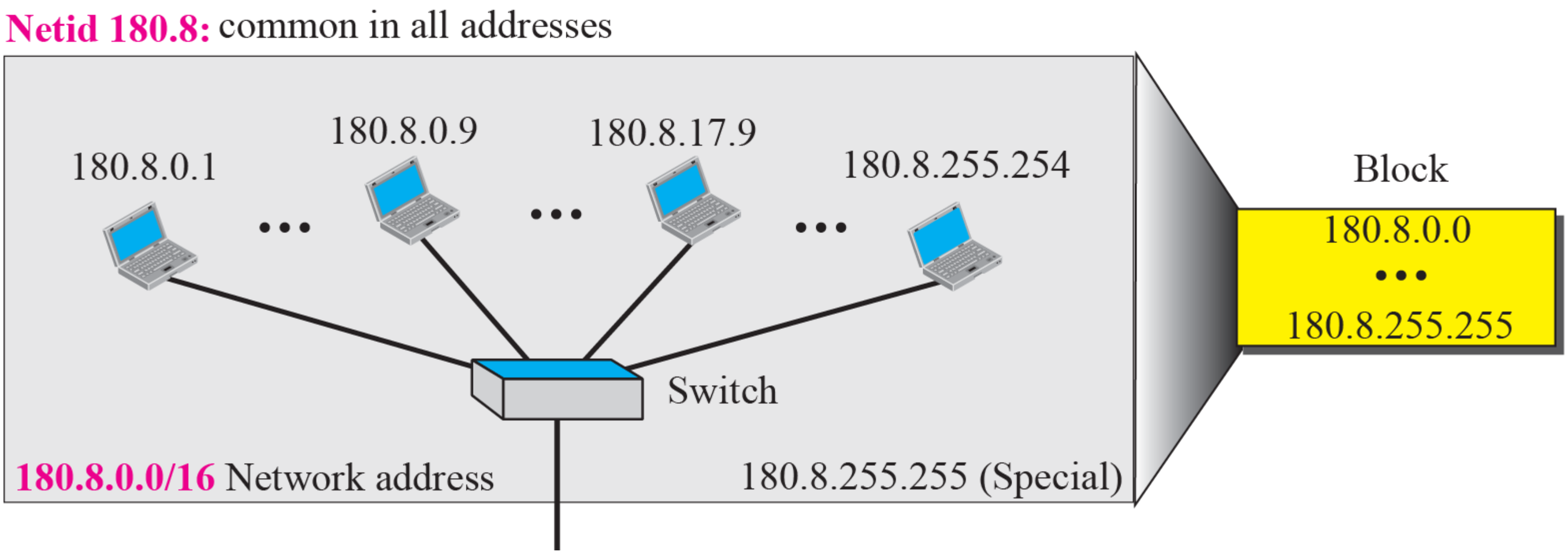
Figure 16 *Solution to Example 5.13*



Example 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.

Figure 17 *Solution to Example 5.14*



Example 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 200.11.8.0/24, 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 200.11.8.255/24.

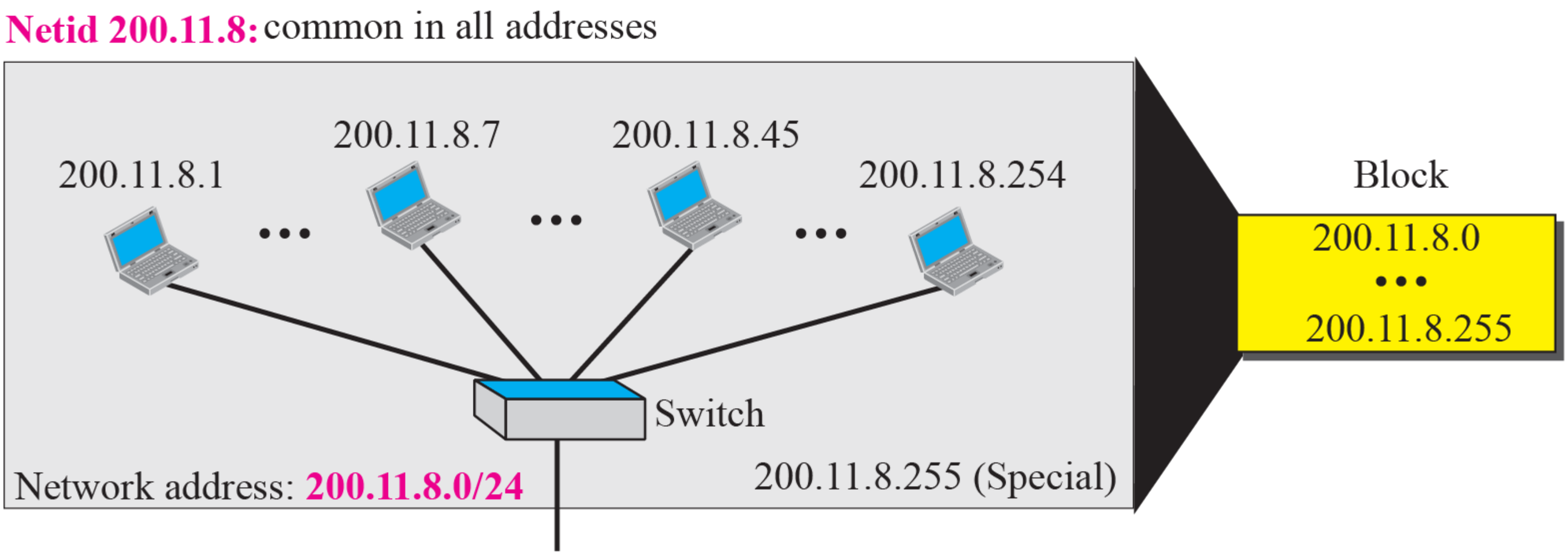
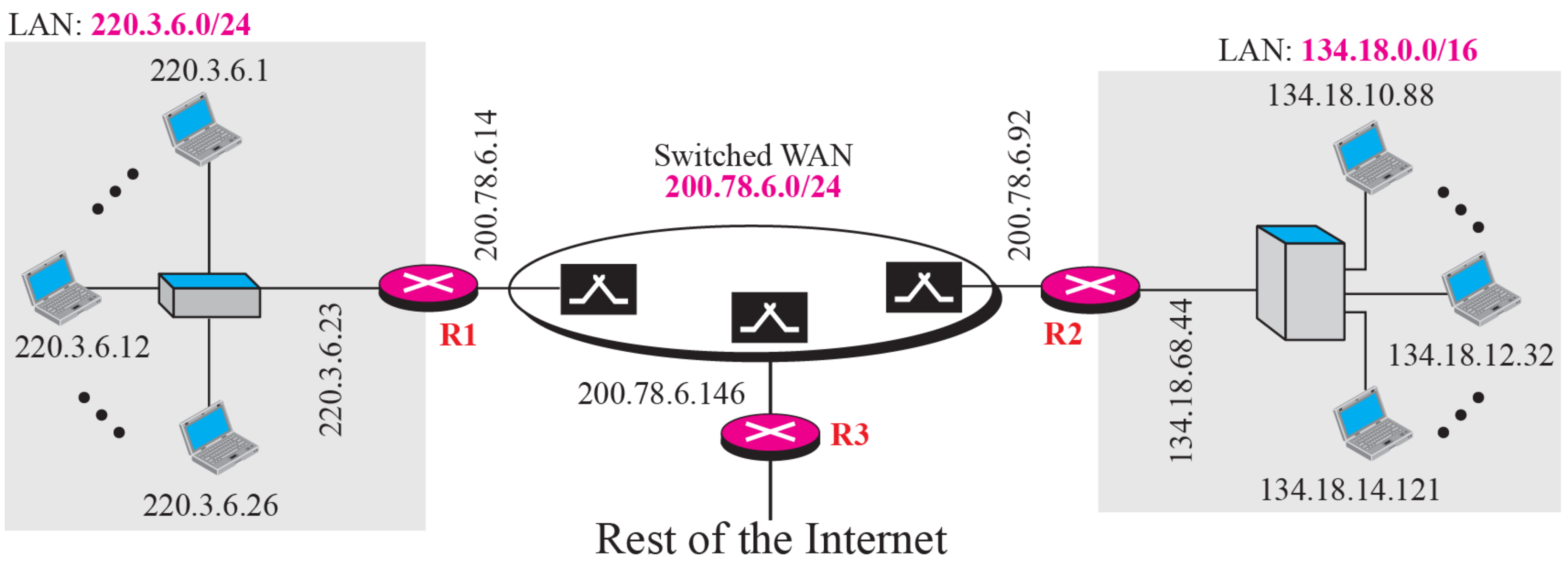

Netid 200.11.8: common in all addresses
200.11.8.1
200.11.8.7
200.11.8.45
200.11.8.254
Switch
Network address: 200.11.8.0/24
200.11.8.255 (Special)
Block
200.11.8.0
200.11.8.255

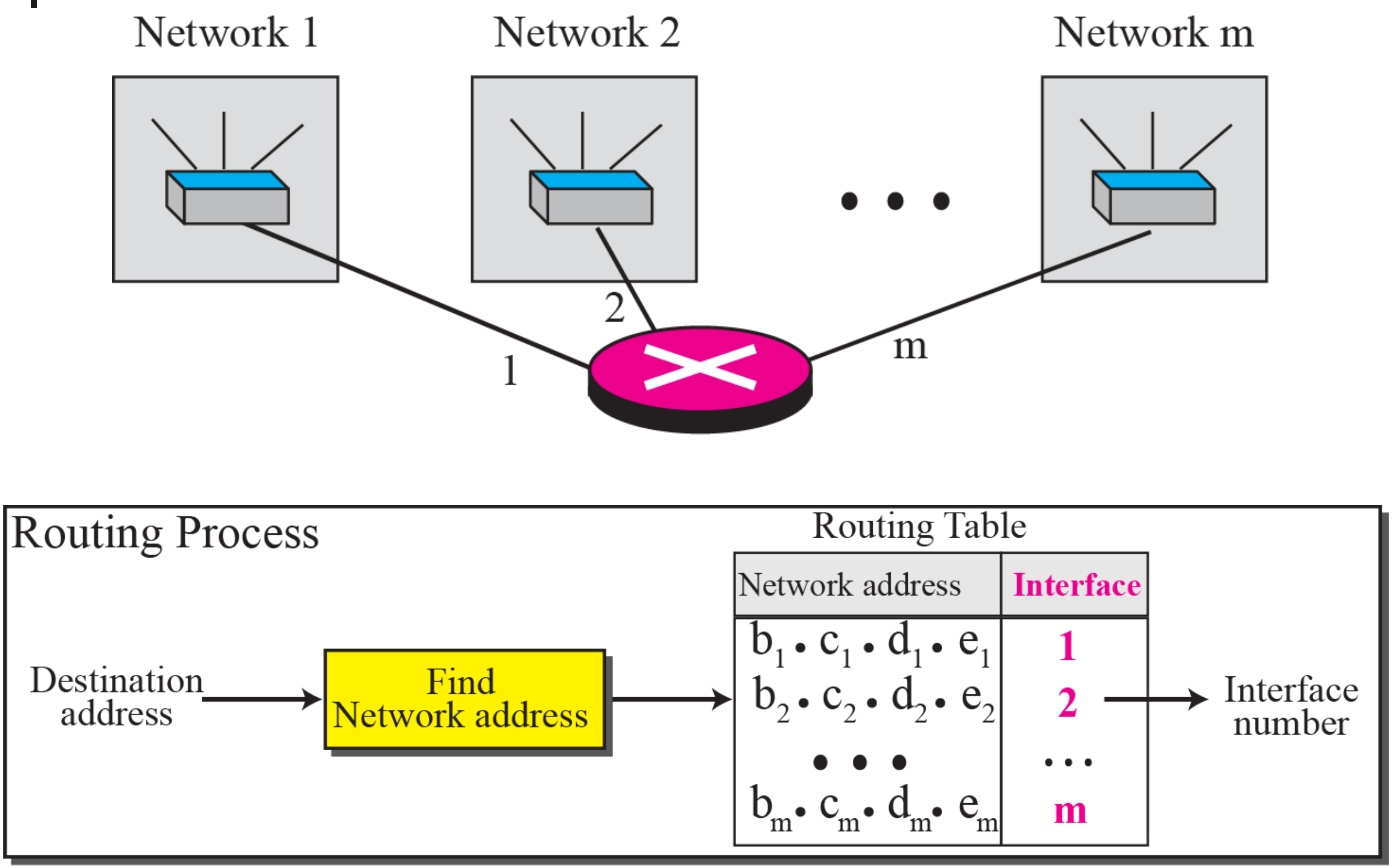
Figure 19 *Sample Internet*



Note

The network address is the identifier of a network.

Figure 20 *Network addresses*



Network address	Interface
$b_1 \cdot c_1 \cdot d_1 \cdot e_1$	1
$b_2 \cdot c_2 \cdot d_2 \cdot e_2$	2
• • •	• • •
$b_m \cdot c_m \cdot d_m \cdot e_m$	m

Figure 21 *Network mask*

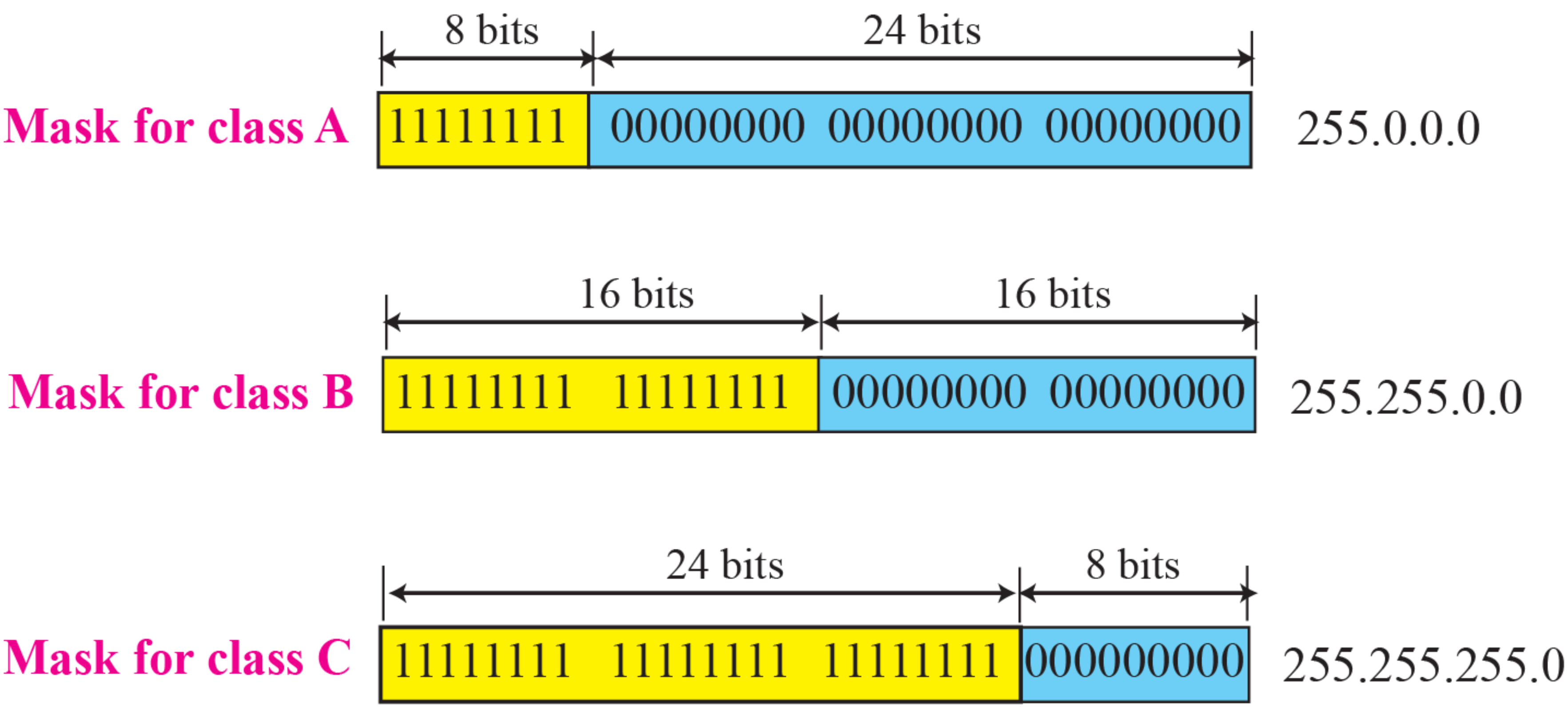
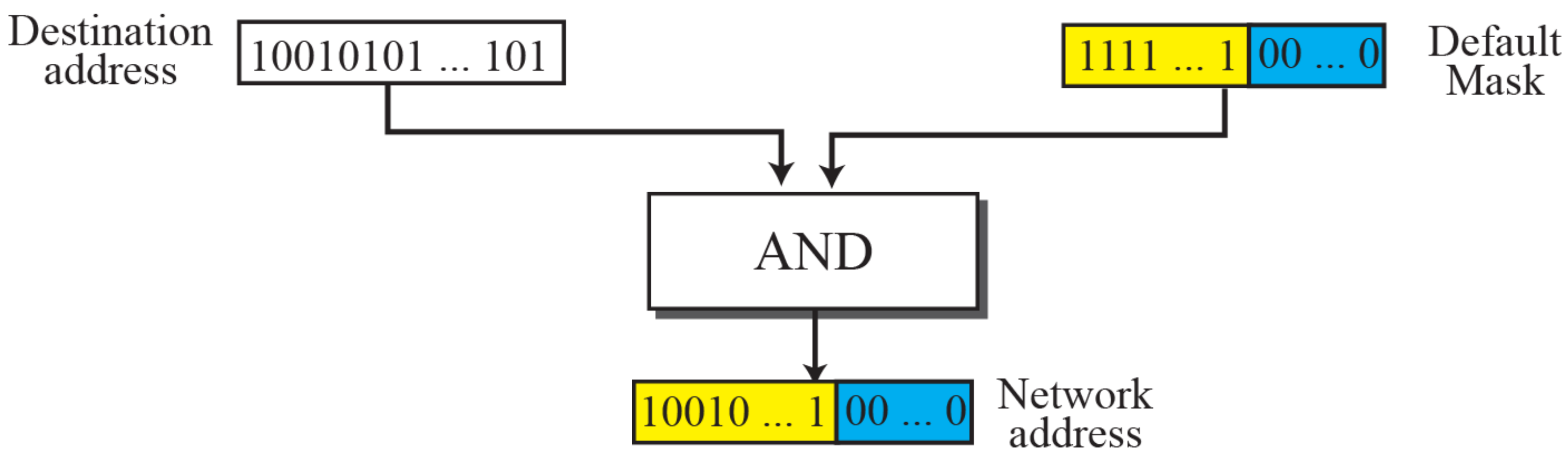


Figure 22 *Finding a network address using the default mask*



Example 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 C, 255.255.255.0 to find the network address.

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

Example 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:

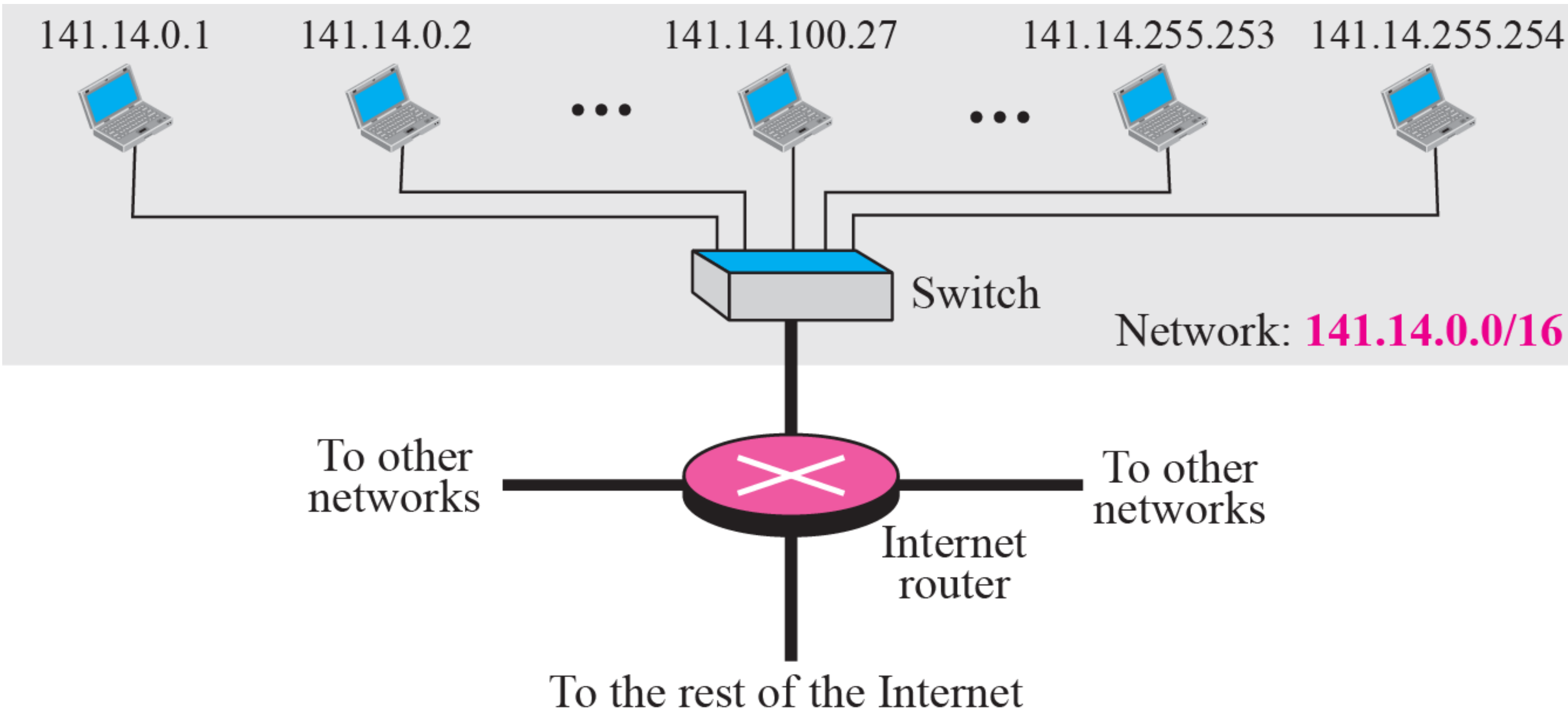
(626) 358 - 1301

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

Example 18

Figure 5.23 shows a network using class B addresses before subnetting. We have just one network with almost 216 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 23 *Example 5.18*



Example 19

Figure 5.24 shows the same network in Figure 5.23 after subnetting. The whole network is still connected to the Internet through the same router. However, the network has used a private router to divide the network into four subnetworks. The rest of the Internet still sees only one network; internally the network is made of four subnetworks. Each subnetwork can now have almost 214 hosts. The network can belong to a university campus with four different schools (buildings). After subnetting, each school has its own subnetworks, but still the whole campus is one network for the rest of the Internet. Note that /16 and /18 show the length of the netid and subnetids.

Figure 5.24 *Example 5.19*

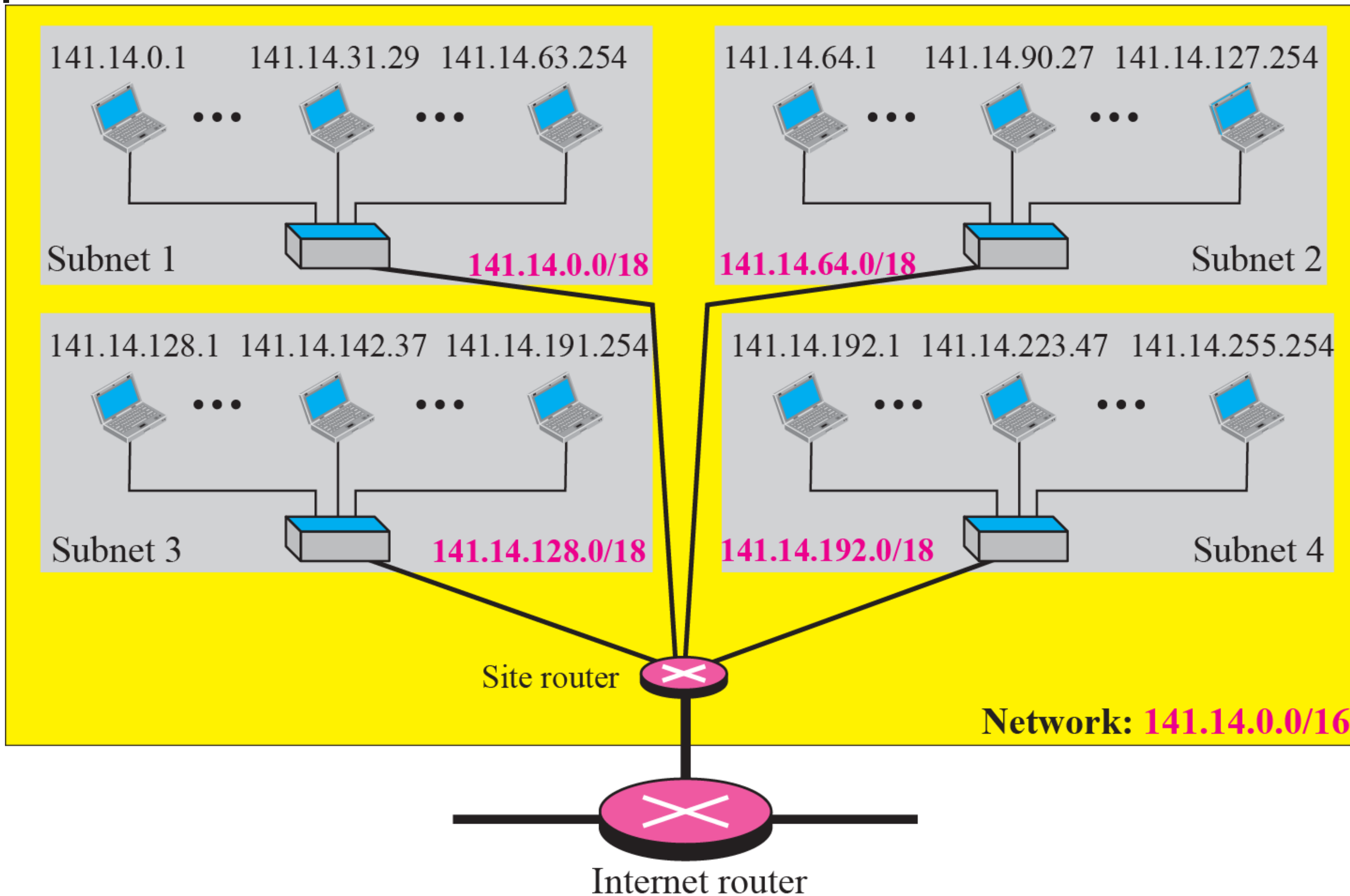
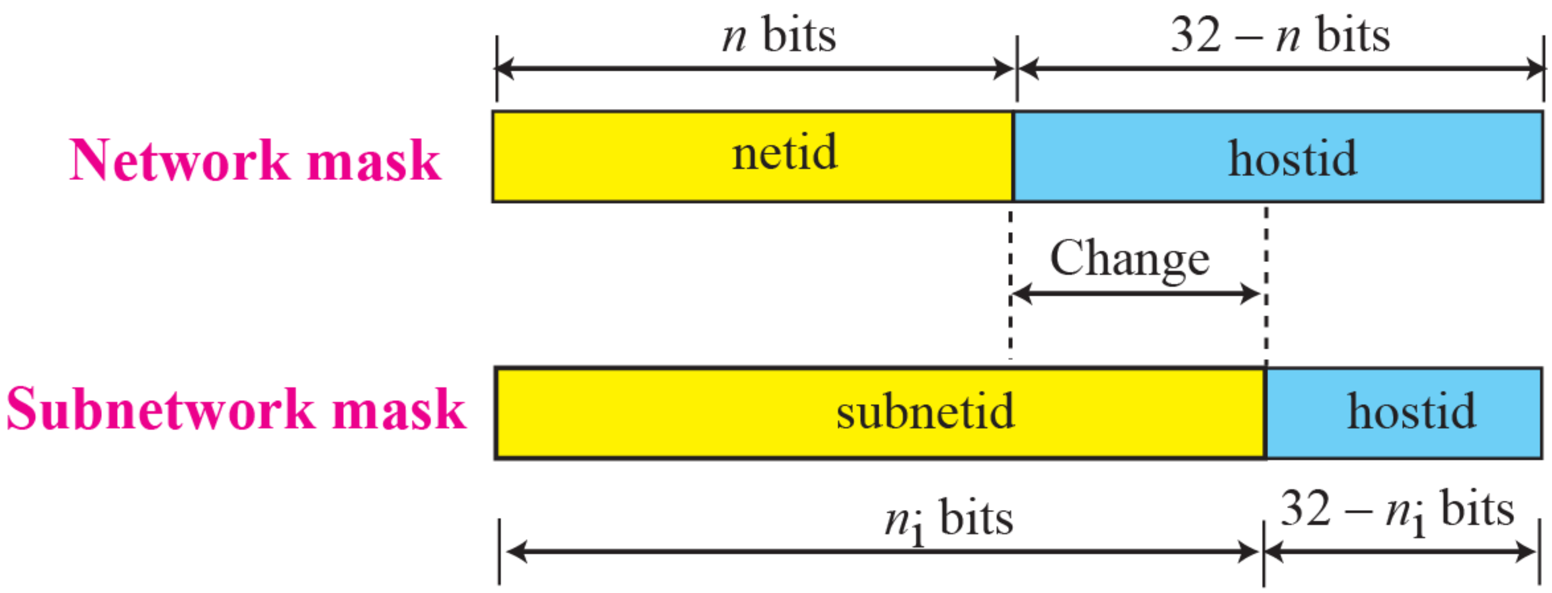


Figure 25 *Network mask and subnetwork mask*



Example 20

In Example 5.19, we divided a class B network into four subnetworks. The value of n = 16 and the value of

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

$$n_1 = n_2 = n_3 = n_4 = 16 + \log_2 4 = 18.$$

This means that the subnet mask has eighteen 1s and fourteen 0s. In other words, the subnet mask is 255.255.192.0 which is different from the network mask for class B (255.255.0.0).

Example 21

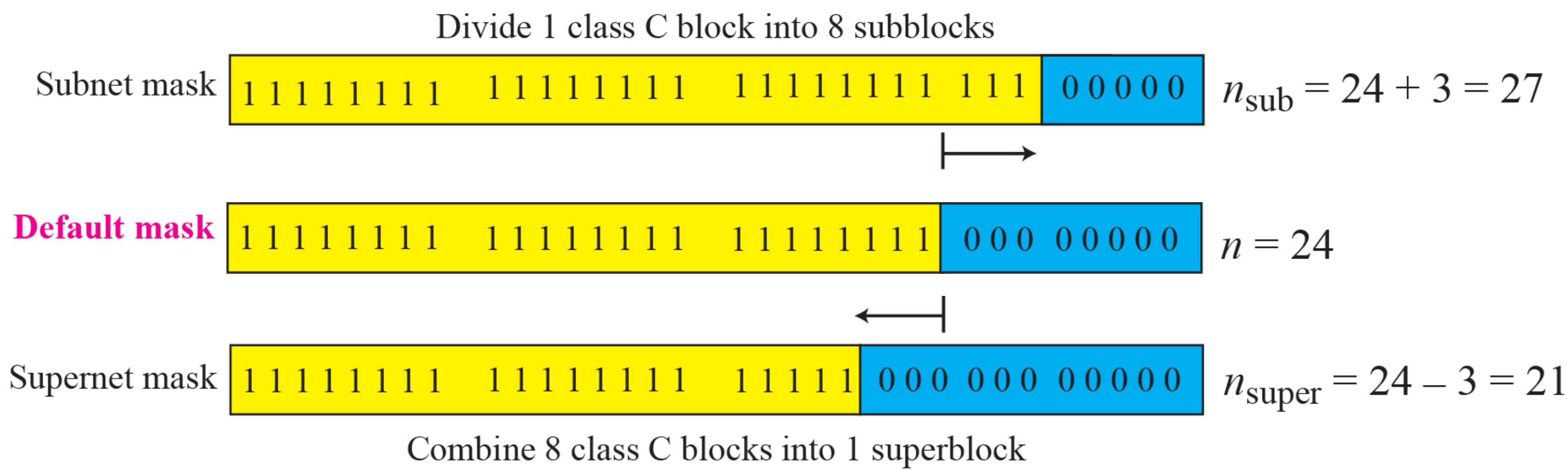
In Example 5.19, we show that a network is divided into four subnets. Since one of the addresses in subnet 2 is 141.14.120.77, we can find the subnet address as:

Address	→	141	.	14	.	120	.	77
Mask	→	255	.	255	.	192	.	0
Subnet Address	→	141	.	14	.	64	.	0

The values of the first, second, and fourth bytes are calculated using the first short cut for AND operation. The value of the third byte is calculated using the second short cut for the AND operation.

Address (120)	0	+	64	+	32	+	16	+	8	+	0	+	0	+	0
Mask (192)	128	+	64	+	0	+	0	+	0	+	0	+	0	+	0
Result (64)	0	+	64	+	0	+	0	+	0	+	0	+	0	+	0

Figure 26 *Comparison of subnet, default, and supernet mask*



Supernetting

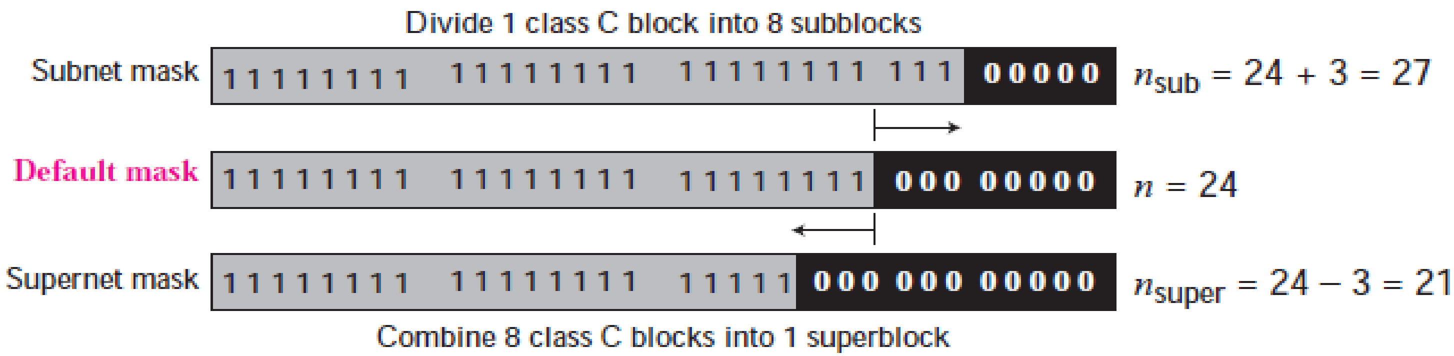
Subnetting could not completely solve address depletion problems in classful addressing because most organizations did not want to share their granted blocks with others. Since class C blocks were still available but the size of the block did not meet the requirement of new organizations that wanted to join the Internet, one solution was **supernetting.** In supernetting, an organization can combine several class C blocks to create a larger range of addresses. In other words, several networks are combined to create a supernetwork. By doing this, an organization can apply for several class C blocks instead of just one. For example, an organization that needs 1000 addresses can be granted four class C blocks.

Supernet Mask

A **supernet mask** is the reverse of a subnet mask. A subnet mask for class C has more 1s than the default mask for this class. A supernet mask for class C has less 1s than the default mask for this class.

Figure 5.26 shows the difference between a subnet mask and a supernet mask. A subnet mask that divides a block into eight subblocks has three more 1s ($2^3 = 8$) than the default mask; a supernet mask that combines eight blocks into one superblock has three less 1s than the default mask.

Figure 5.26 *Comparison of subnet, default, and supernet masks*



In supernetting, the number of class C addresses that can be combined to make a supernet needs to be a power of 2. The length of the supernetid can be found using the formula

$$n_{super} = n - \log_2 c$$

in which n_{super} defines the length of the supernetid in bits and c defines the number of class C blocks that are combined.

Unfortunately, supernetting provided two new problems: First, the number of blocks to combine needs to be a power of 2, which means an organization that needed seven blocks should be granted at least eight blocks (address wasting). Second, supernetting and subnetting really complicated the routing of packets in the Internet.

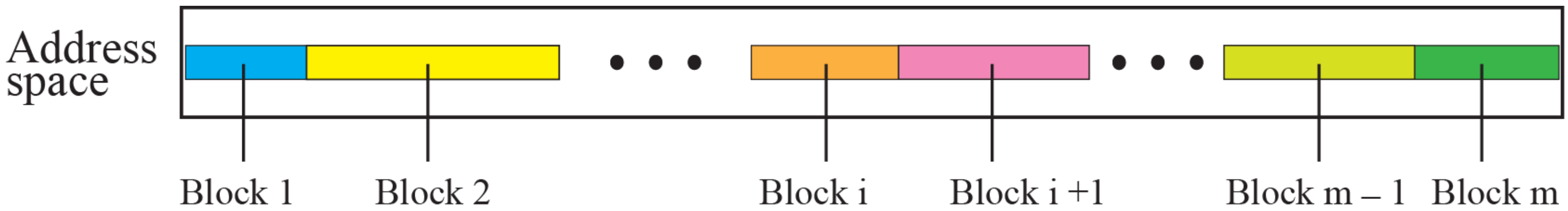
CLASSLESS ADDRESSING

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**

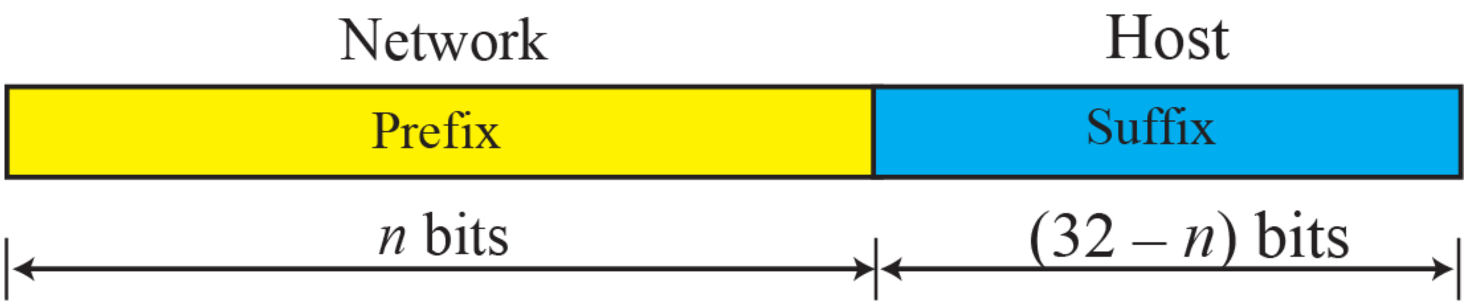
Figure 27 *Variable-length blocks in classless addressing*



Note

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

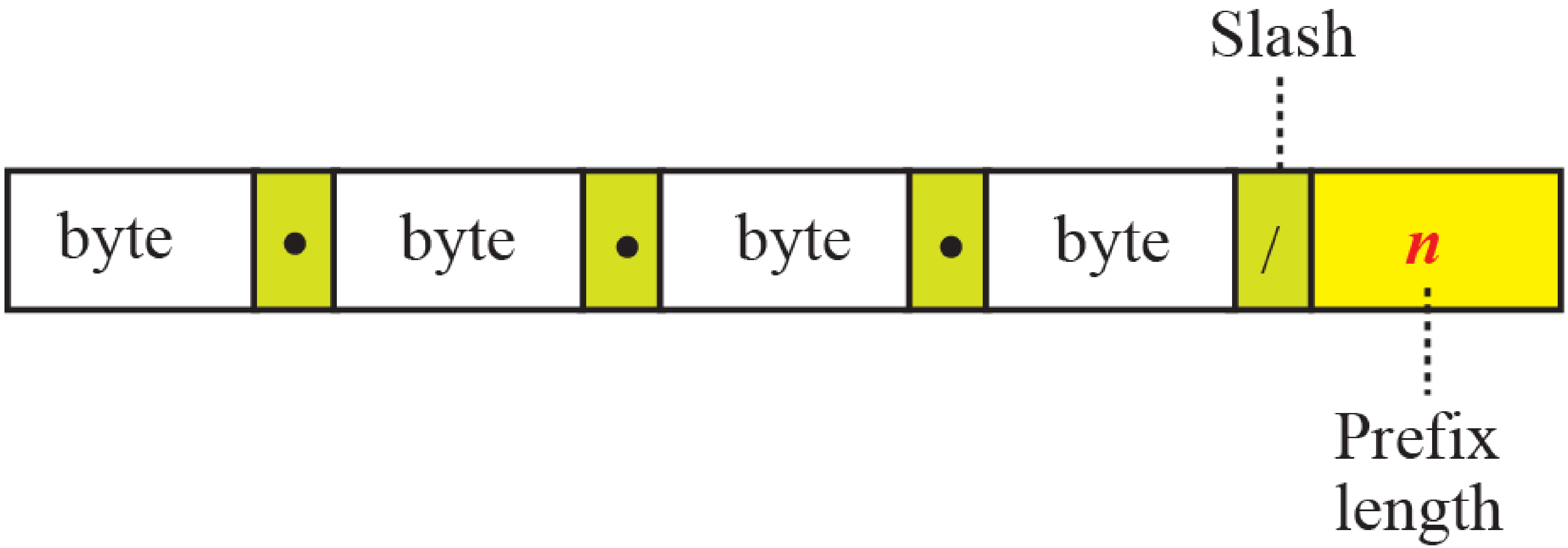
Figure 28 *Prefix and suffix*



Note

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

Figure 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 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 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 255.255.255.224.

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 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 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 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 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 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 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.

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.

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:

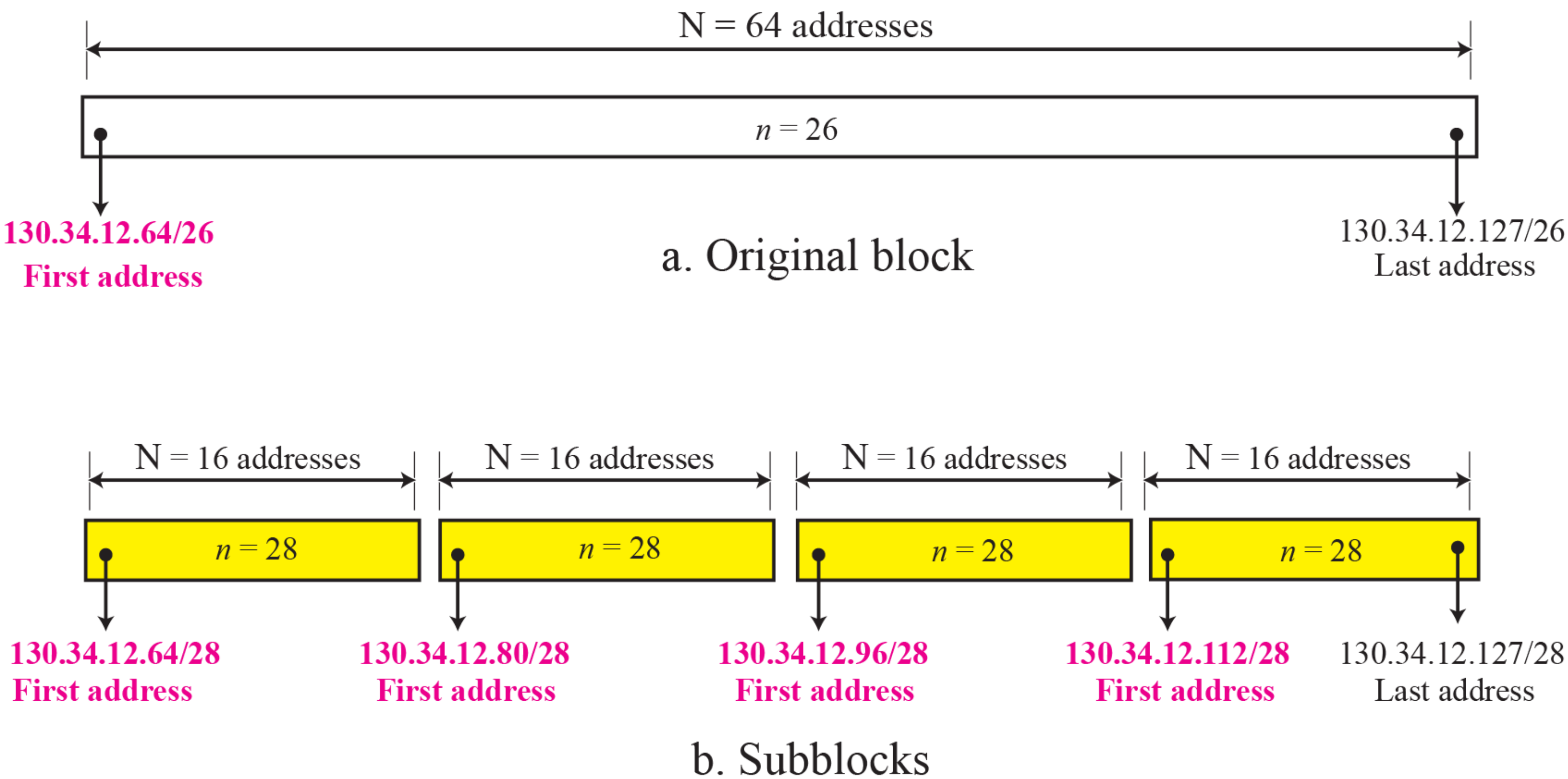
1. We grant 16 addresses for each subnetwork to meet the first requirement (64/16 is a power of 2).
2. The subnetwork mask for each subnetwork is:

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

Example 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 30 *Solution to Example 32*



Example 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 118 addresses.
- ❑ One subblock of 62 addresses.
- ❑ One subblock of 14 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 33 Continued

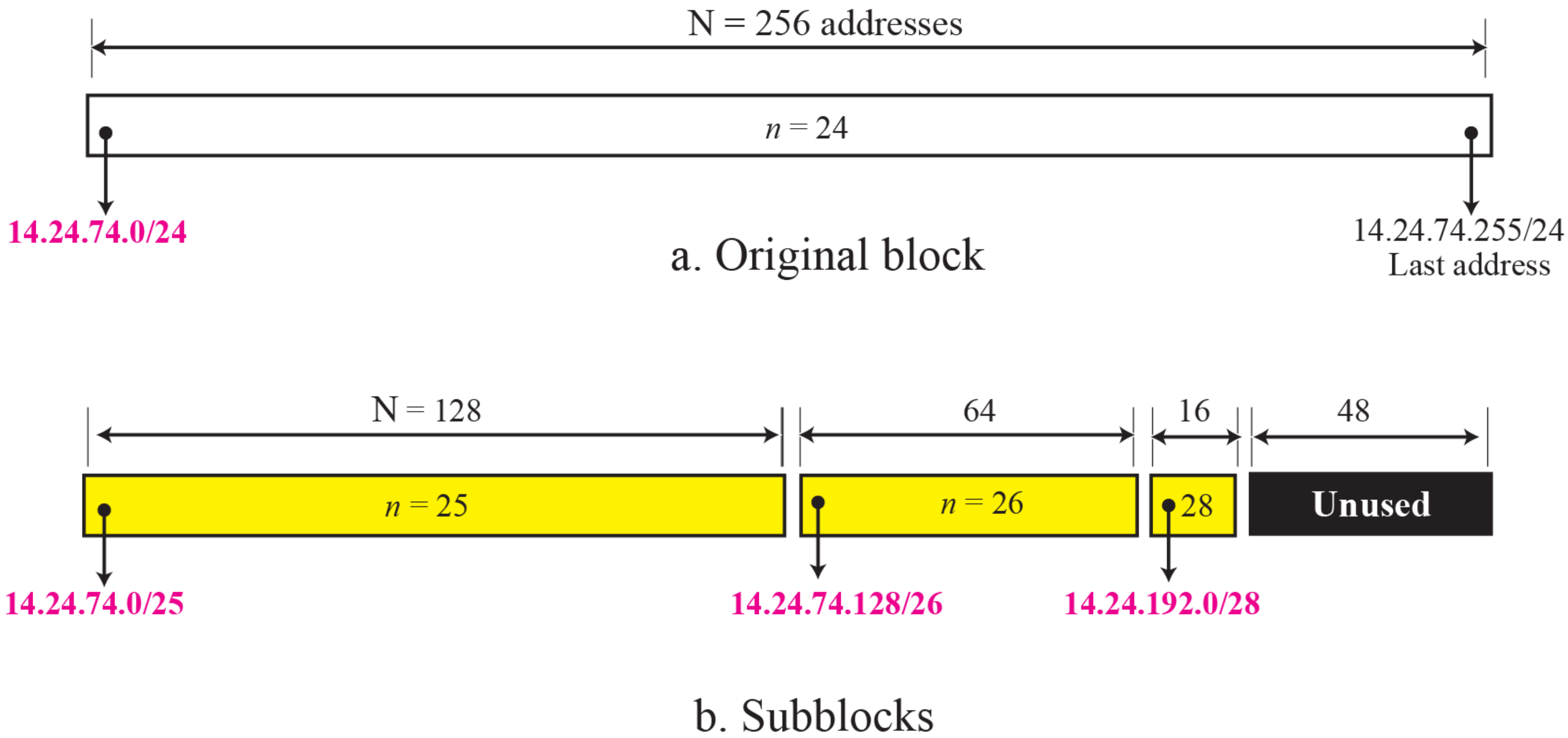
b. The number of addresses in the second subblock is not a power of 2 either. We allocate 64 addresses. The subnet mask is 26. The first address in this block is 14.24.74.128/26; the last address is 14.24.74.191/26.

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 31 *Solution to Example 33*



Example 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:

Central office $N_c = 32$ East office $N_e = 16$ West office $N_w = 16$

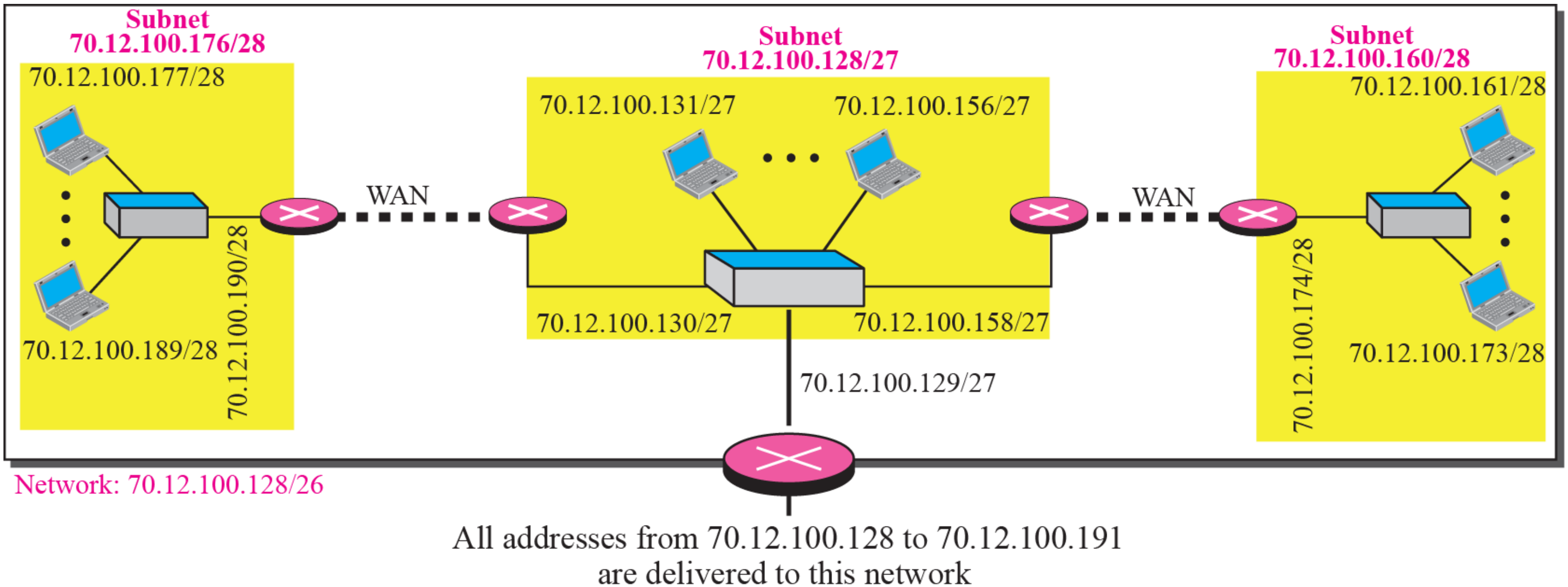
2. We can find the prefix length for each subnetwork:

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

Example 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 32 *Example 34*



Example 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 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 33 *Solution to Example 35: first step*

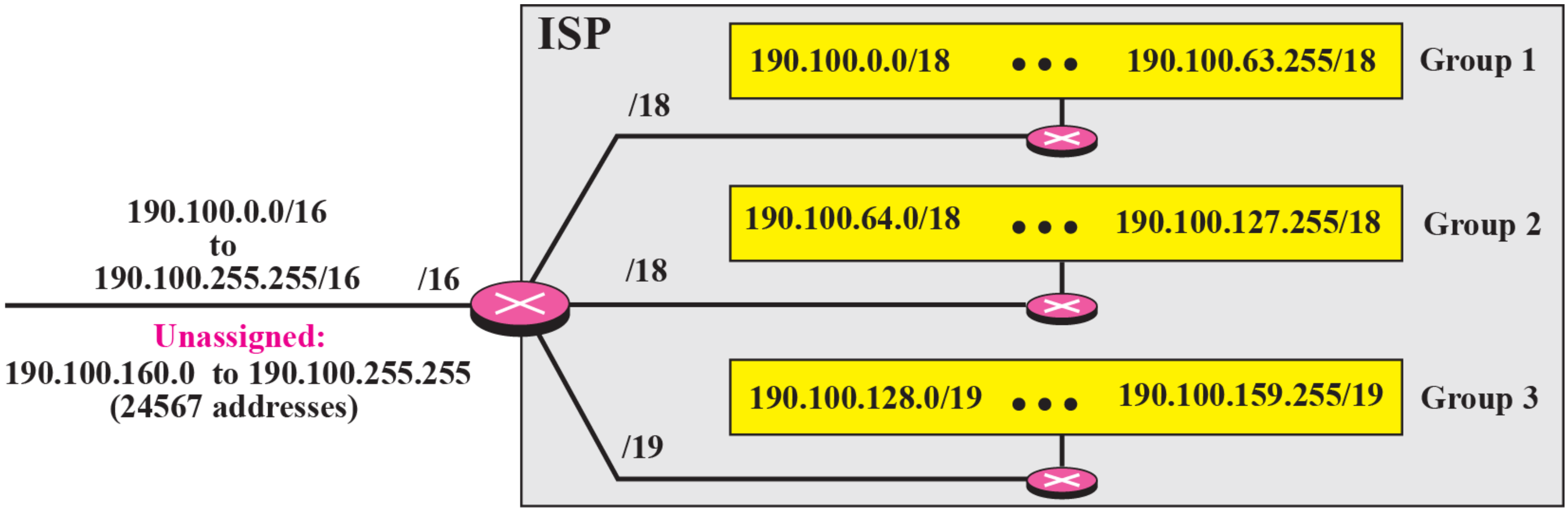


Figure 34 *Solution to Example 5.35: second step*

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