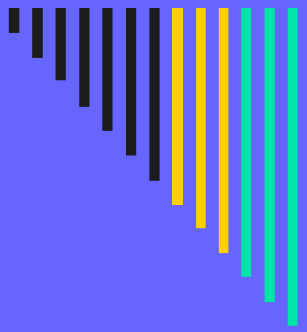
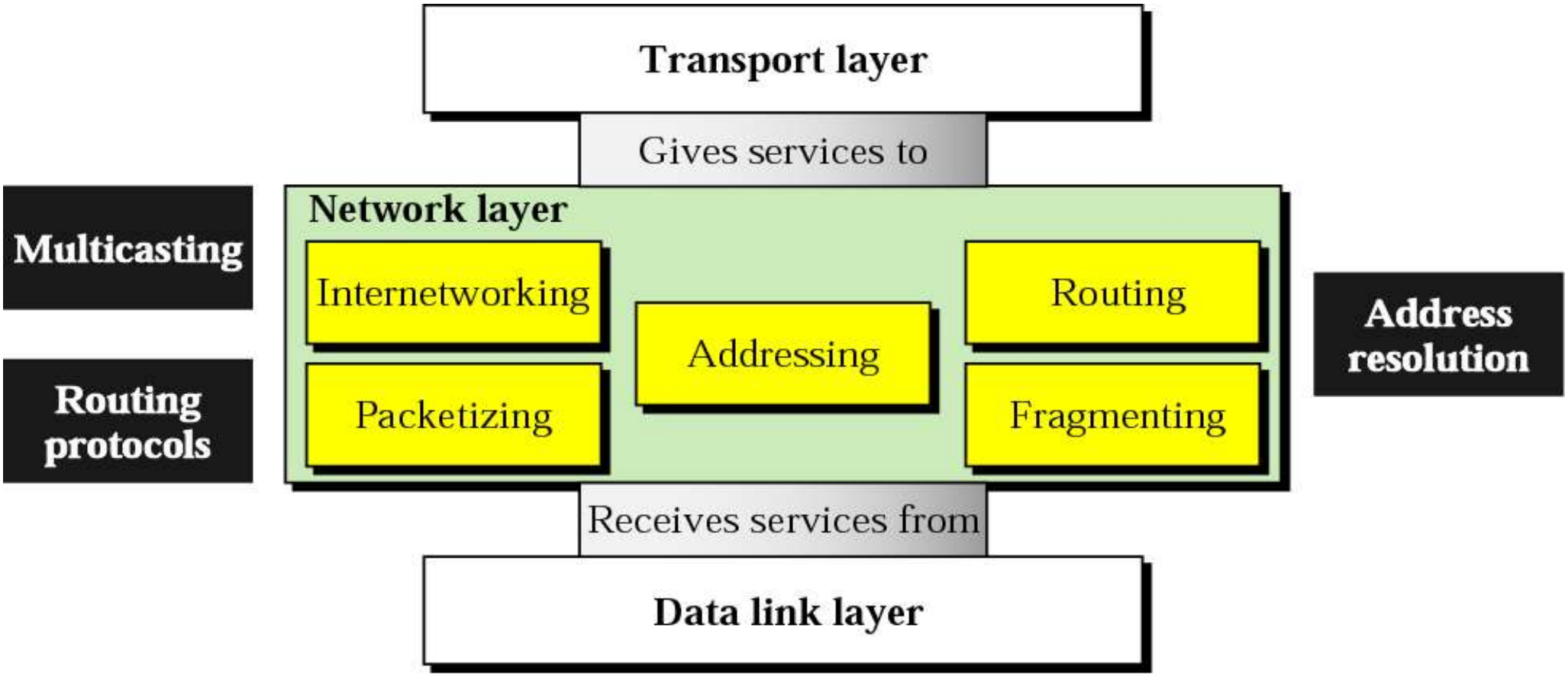
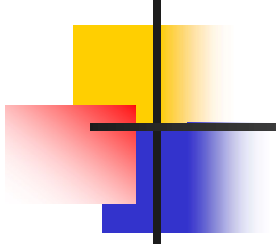


PART IV

Network Layer

Position of network layer





Chapters

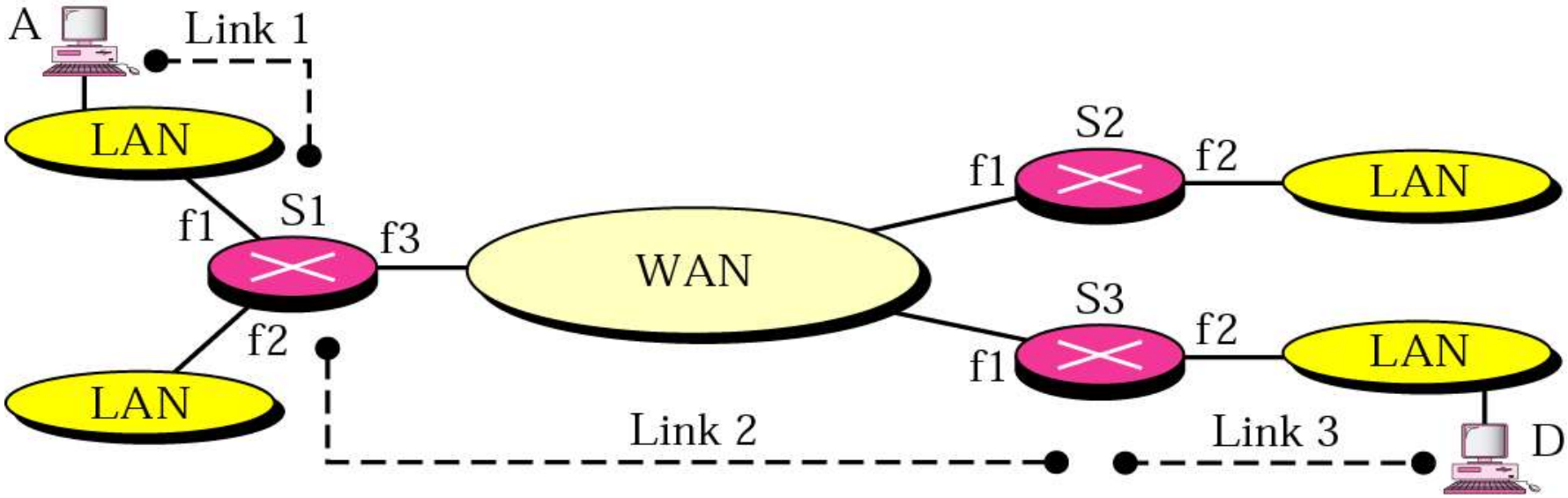
Data Communications and Networking Fourth Edition

Forouzan

Chapter 19

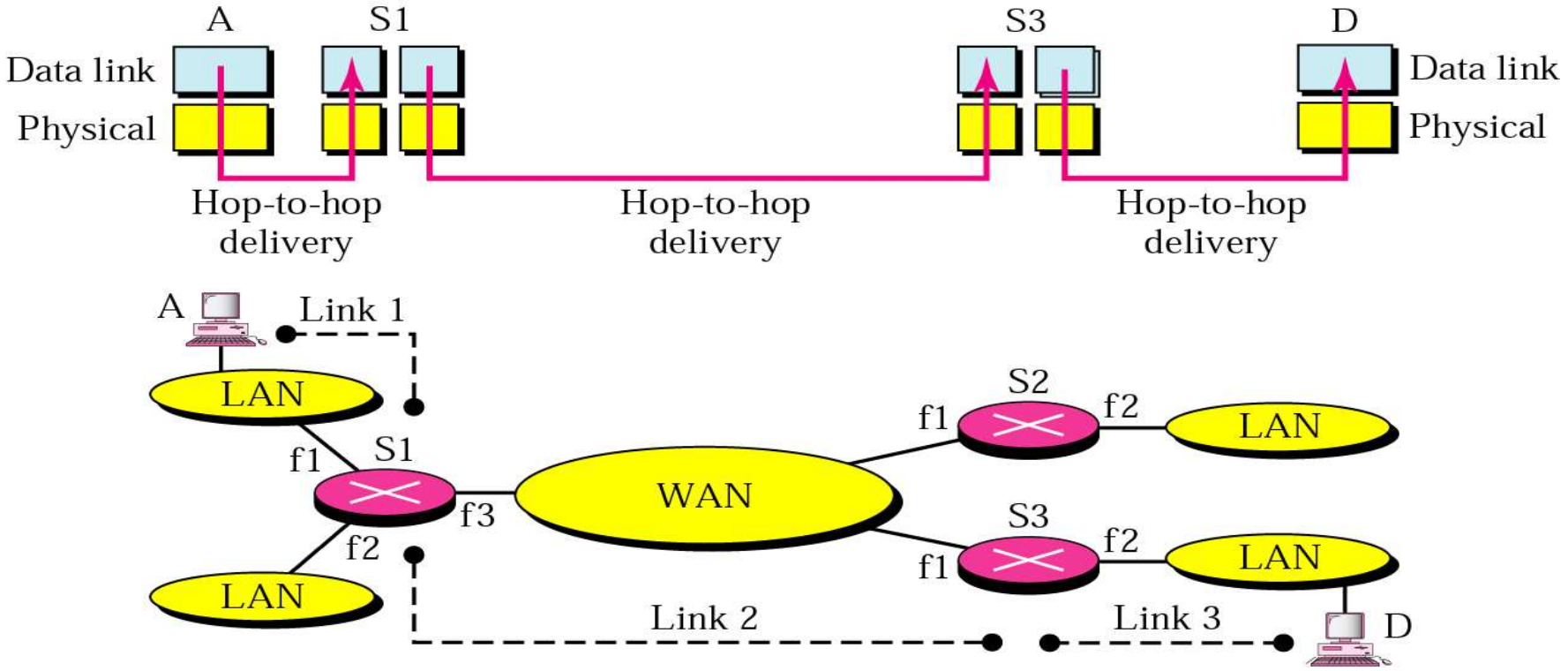
Network Layer: Logical Addressing

How data can be exchanged between networks?
They need to be connected via routers/links to make an internetwork



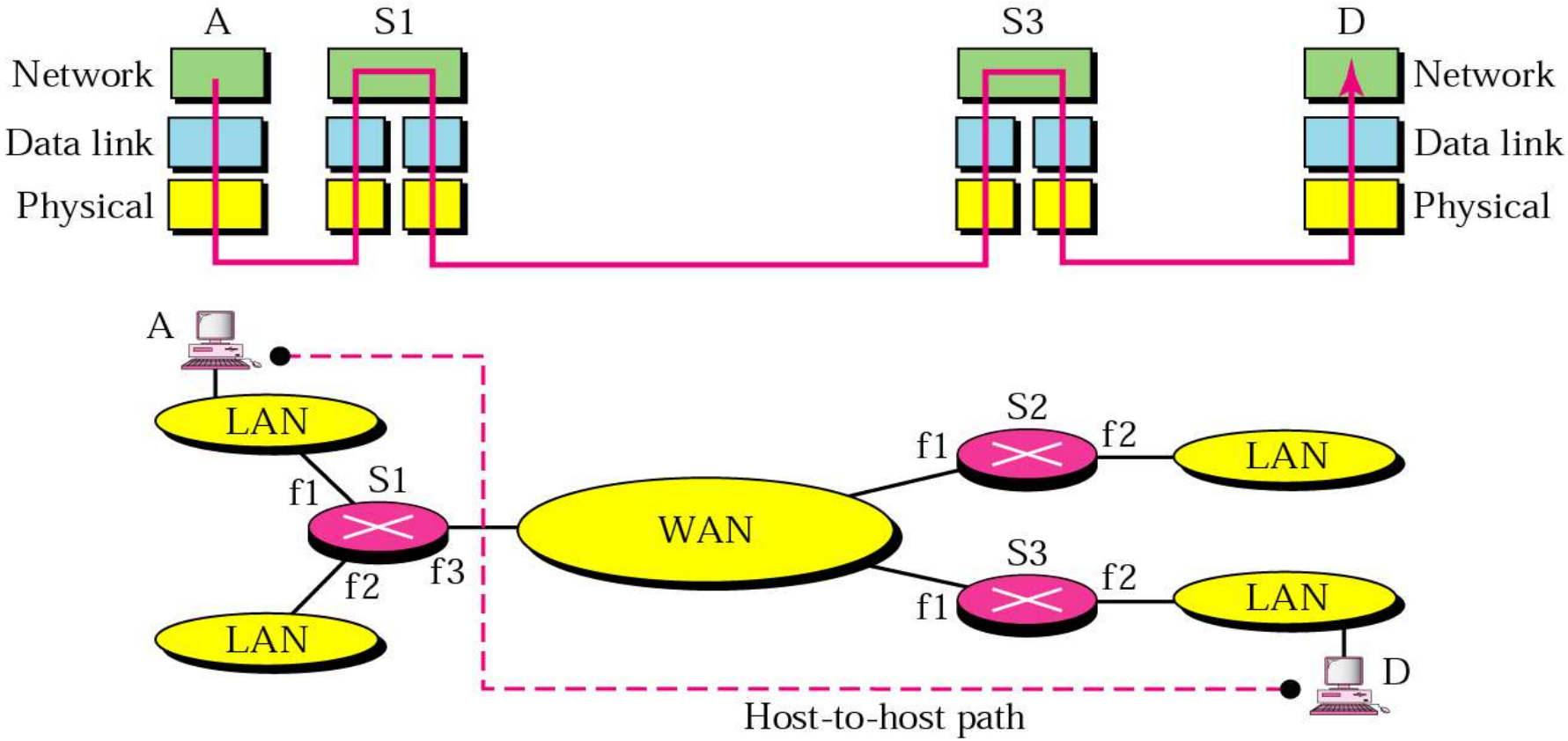
- The above internetwork is made of 5 networks: LAN and 1 WAN
- E.g if host A needs to send a data packet to host D, the packet needs to go from A to S1, then from S1 to S3, and finally from S3 to D. Therefore the packet passes through 3 links

Links in an internetwork



Problem: How does S1 know that they should send out from f3 after the packet arrives at f1 from A ? (No provision in the data link layer to help S1 making the decision and the frame only contains the MAC addresses-pair of 1st link)

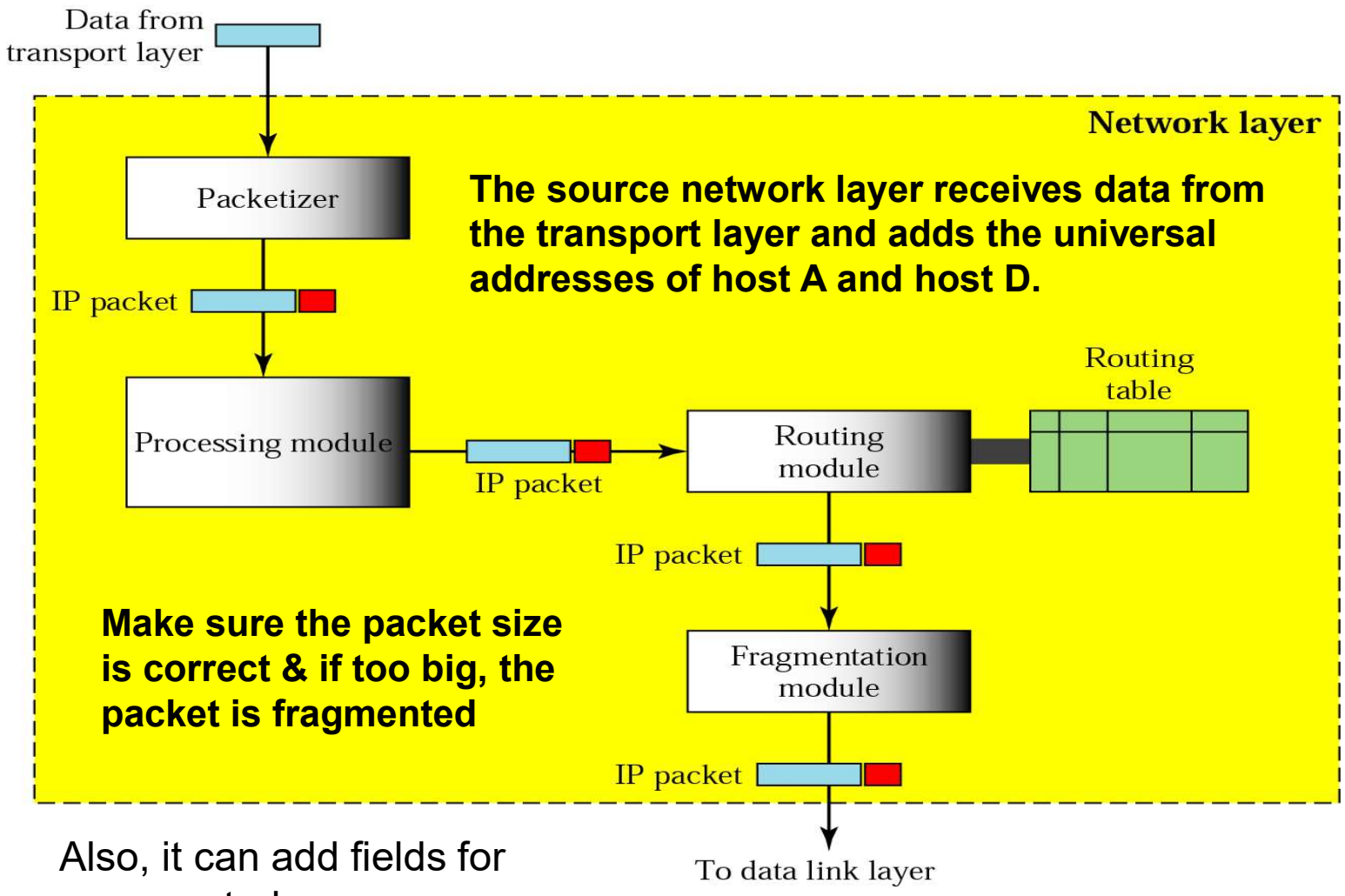
Network layer in an internetwork



To solve the problem of delivery through several links, the network layer was designed and responsible for host-to-host delivery and for routing the packets through different routers

Network layer at the source

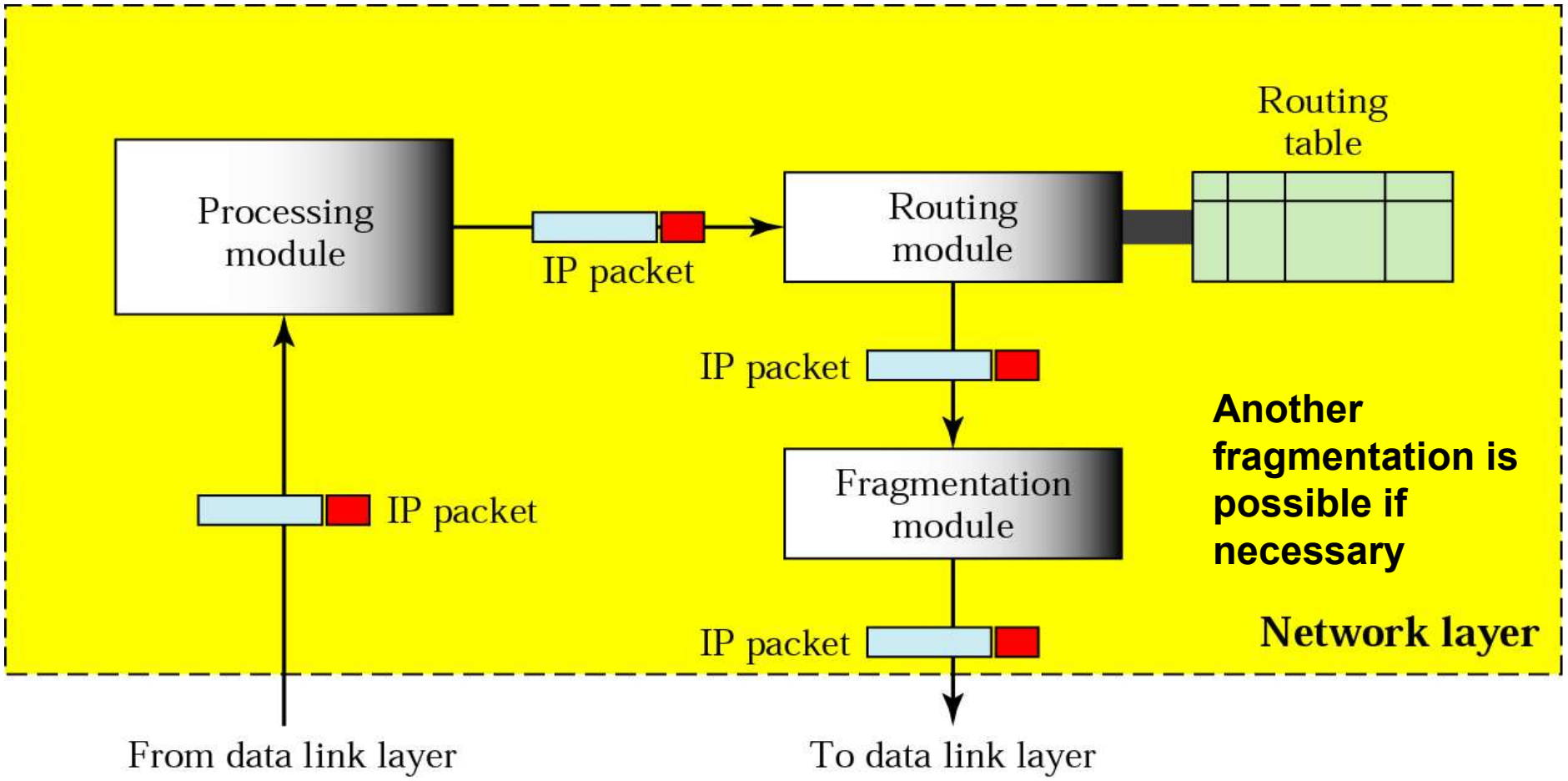
Network layer at the source is responsible for creating a packet that carries 2 universal addresses: source add. and destination add.



Also, it can add fields for error control

Network layer at a router

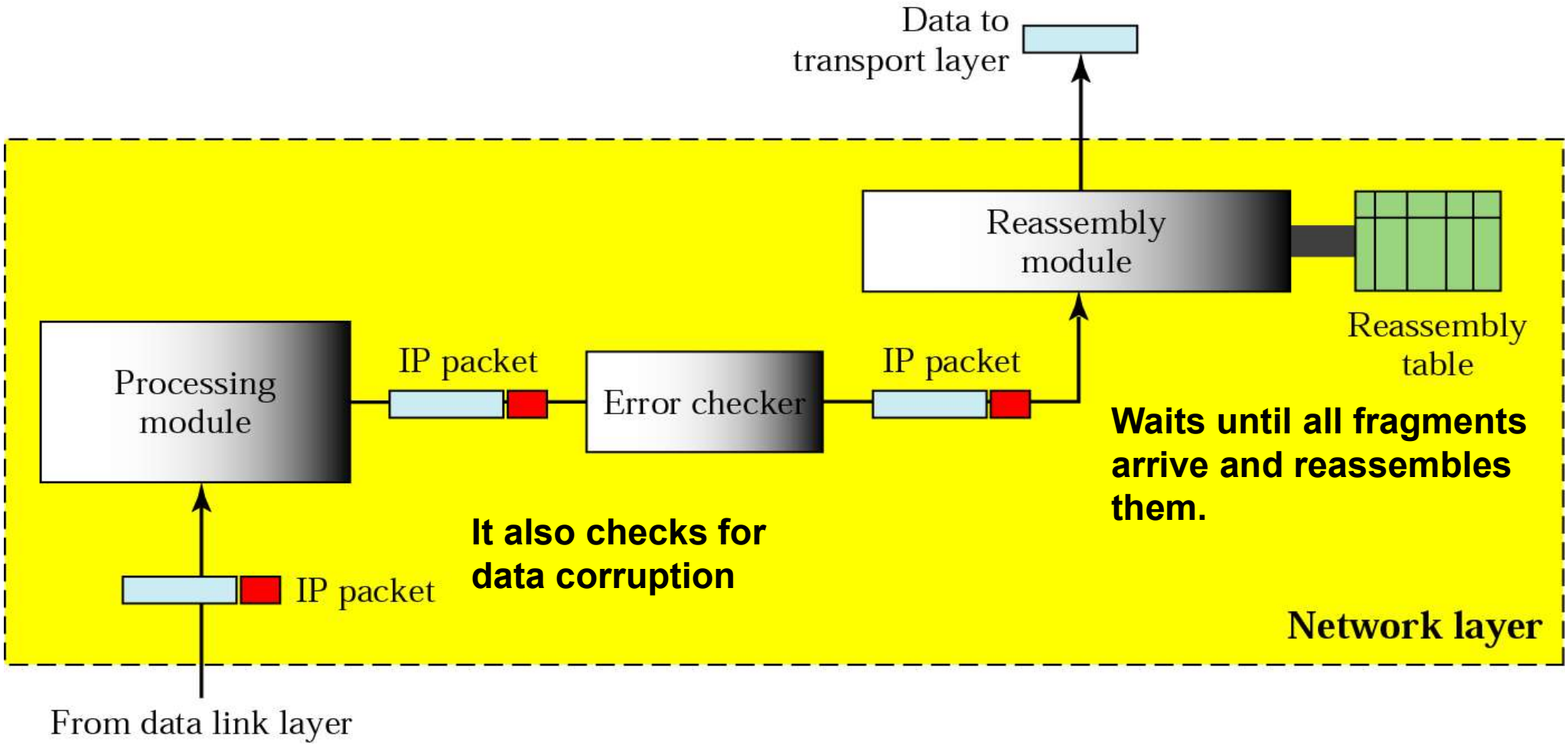
The network layer at the router is responsible for routing the packets



When the packet arrives, the router finds the interface from which the packet must be sent using the routing table

Network layer at the destination

The network layer at the destination is responsible for address verification. It makes sure that the destination address on the packet is the same as the address of the receiving host.



19-1 IPv4 ADDRESSES

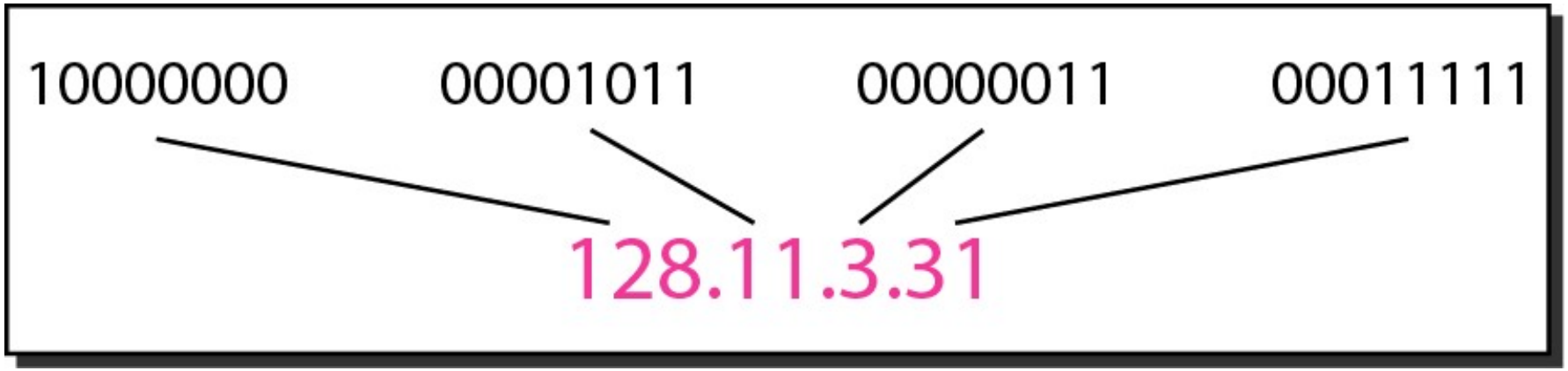
An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a device (for example, a computer or a router) to the Internet.

Topics discussed in this section:

Address Space
Notations
Classful Addressing
Classless Addressing
Network Address Translation (NAT)

Figure 19.1 *Dotted-decimal notation and binary notation for an IPv4 address*

An IPv4 address is 32 bits long.
The IPv4 addresses are unique and universal.
The address space of IPv4 is 2^{32} or 4,294,967,296.



Example 19.1

Change the following IP4 addresses from binary notation to dotted-decimal notation.

a. 10000001 00001011 00001011 11101111
b. 11000001 10000011 00011011 111111111

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

Example 19.2

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

a. 111.56.45.78

b. 221.34.7.82

Solution

We replace each decimal number with its binary equivalent

a. 01101111 00111000 00101101 01001110

b. 11011101 00100010 00000111 01010010

Example 19.3

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 must be no leading zero (045).

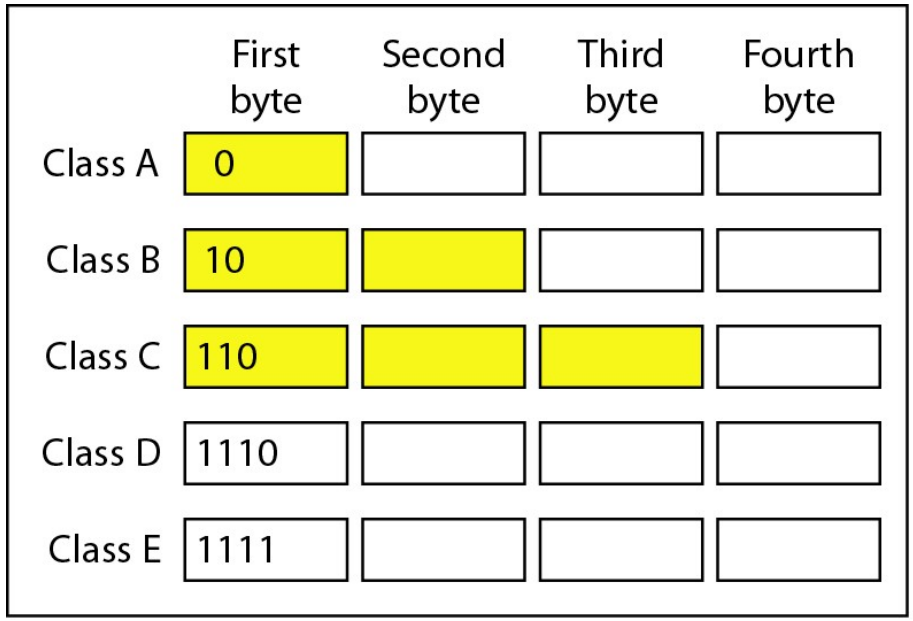
b. There can be no more than four numbers.

c. Each number needs to be less than or equal to 255.

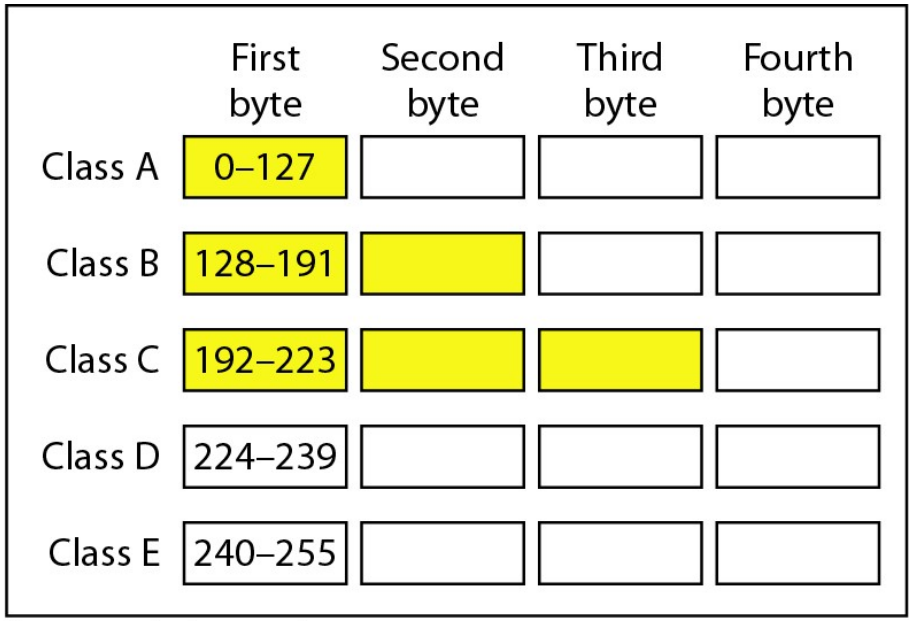
d. A mixture of binary notation and dotted-decimal notation is not allowed.

Figure 19.2 *Finding the classes in binary and dotted-decimal notation*

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



a. Binary notation



b. Dotted-decimal notation

Netid and hostid

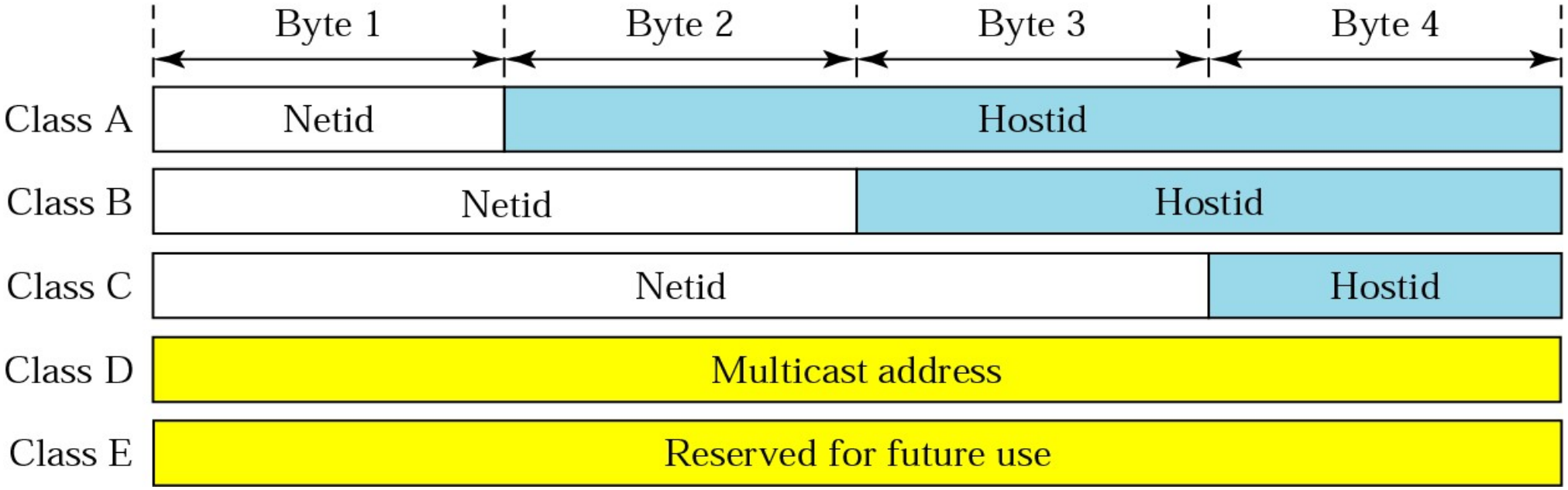
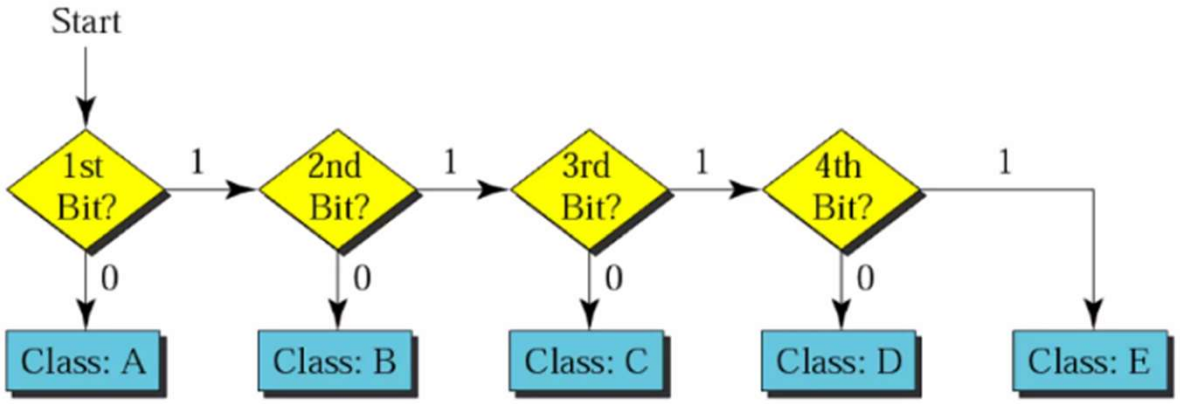


Table 19.1 *Number of blocks and block size in classful IPv4 addressing*

Class	*Number of Blocks*	*Block Size*	*Application*
A	128	16,777,216	Unicast
B	16,384	65,536	Unicast
C	2,097,152	256	Unicast
D	1	268,435,456	Multicast
E	1	268,435,456	Reserved

Finding the address class



Example 19.4

Find the class of each address

a. 00000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 111111111

c. 14.23.120.8

d. 252.5.15.111

Solution

a. The first bit is 0; this is a class-A address

b. The first two bits are 1; the third bit is 0. This is a class C address

c. The first byte is 14; the class is A

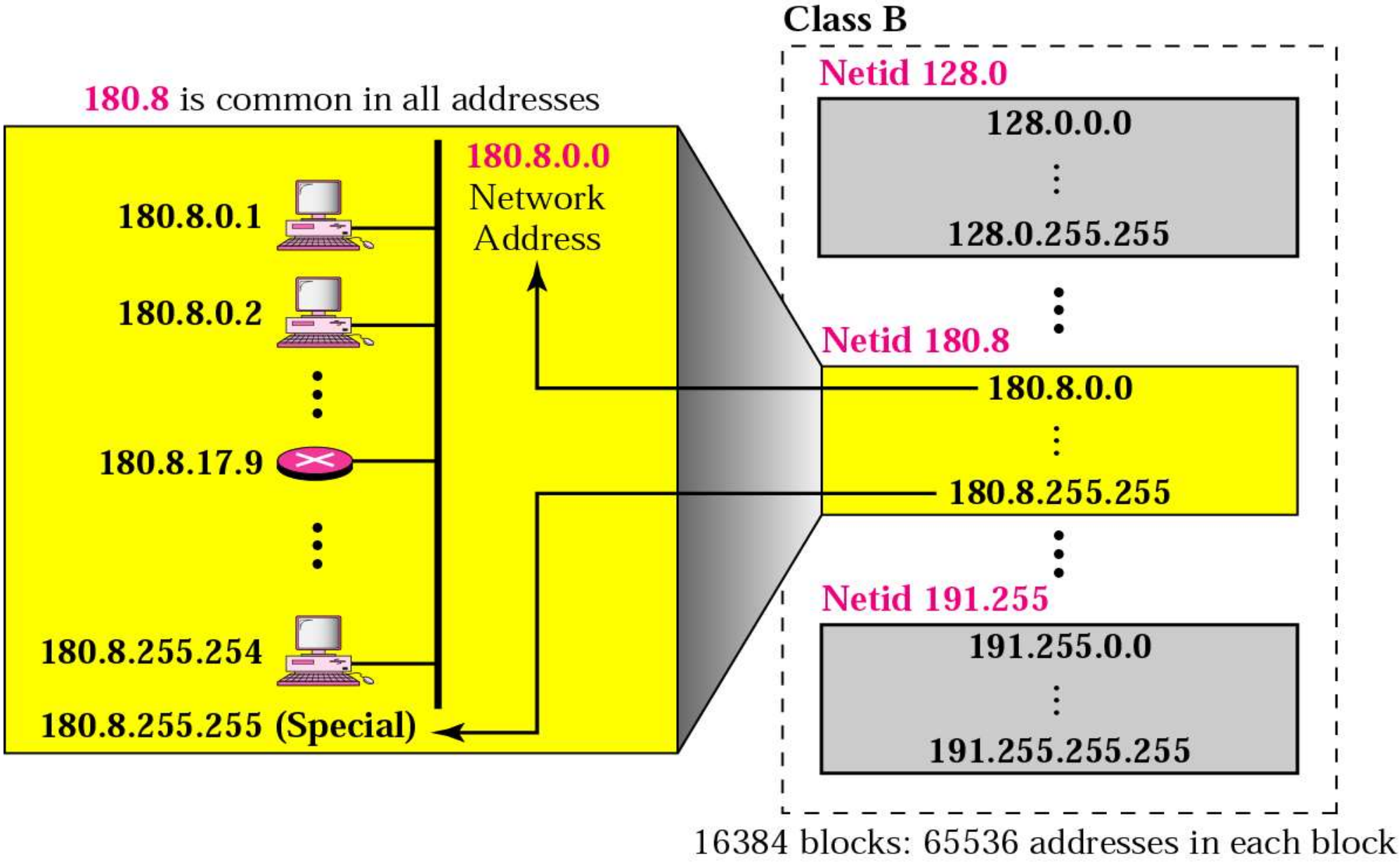
d. The first byte is 252; the class is E

Table 19.2 *Default masks for classful addressing*

Class	*Binary*	*Dotted-Decimal*	*CIDR*
A	11111111 00000000 00000000 00000000	255.0.0.0	/8
B	11111111 11111111 00000000 00000000	255.255.0.0	/16
C	11111111 11111111 11111111 00000000	255.255.255.0	/24

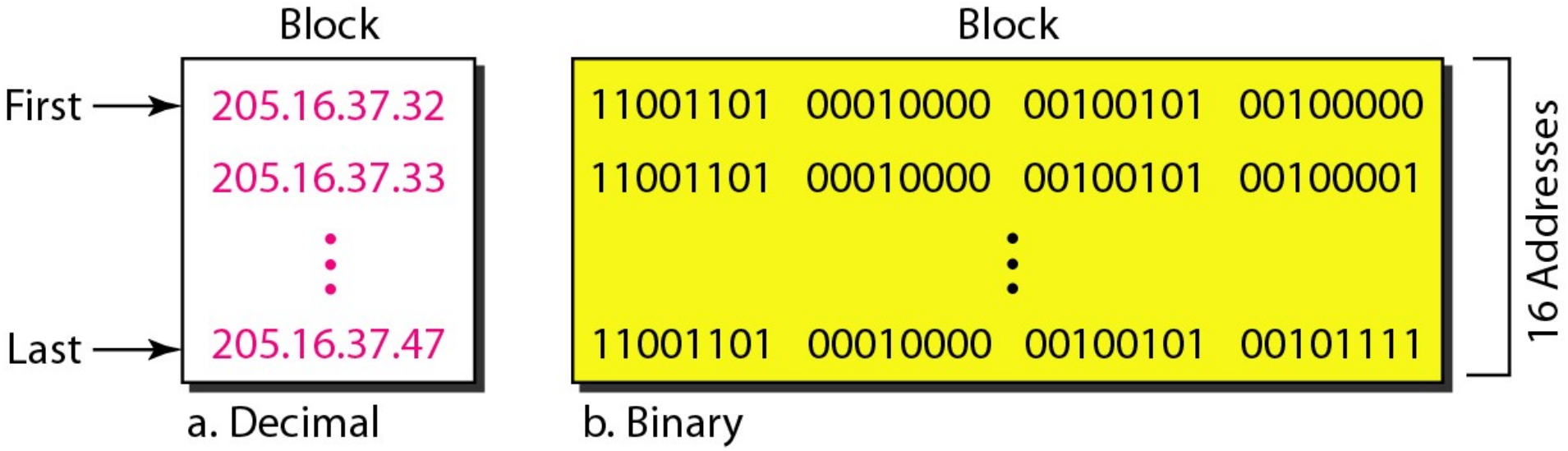
Classful addressing, which is almost obsolete, is replaced with classless addressing.

Classfull Addressing: Blocks in class B



Classless Addressing: *A block of 16 addresses granted to a small organization*

Figure 19.3 shows a block of addresses, in both binary and dotted-decimal notation, granted to a small business that needs 16 addresses.
We can see that the restrictions are applied to this block. The addresses are contiguous. The number of addresses is a power of 2 ($16 = 2^4$).



Note

In IPv4 addressing, a block of addresses can be defined as x.y.z.t /*n* in which x.y.z.t defines one of the addresses and the /*n* defines the mask.

The first address in the block can be found by setting the rightmost 32 − *n* bits to 0s.

The last address in the block can be found by setting the rightmost 32 − n bits to 1s.

The number of addresses in the block can be found by using the formula 2^{32-n}.

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 and the last addresses in the block?

Solution

The binary representation of the given address is

11001101 00010000 00100101 00100111

If we set 32−28 rightmost bits to 0, we get

11001101 00010000 00100101 0010000 or *205.16.37.32.*

If we set 32 − 28 rightmost bits to 1, we get

11001101 00010000 00100101 00101111 or *205.16.37.47*

This is actually the block shown in Figure 19.3.

The value of n is 28, which means that number of addresses is 2^{32-28} *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. The /28 can be represented as

11111111 11111111 11111111 11110000

(twenty-eight 1s and four 0s).

Find

a. The first address

b. The last address

c. The number of addresses.

Example 19.9 (continued)

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 1s; the result is 0 otherwise.

Address:	11001101	00010000	00100101	00100111
Mask:	**11111111**	**11111111**	**11111111**	**11110000**
First address:	11001101	00010000	00100101	00100000

Example 19.9 (continued)

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 0s; the result is 1 otherwise. The complement of a number is found by changing each 1 to 0 and each 0 to 1.***

Address:	11001101	00010000	00100101	00100111
Mask complement:	**00000000**	**00000000**	**00000000**	**00001111**
Last address:	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:	**00000000**	**00000000**	**00000000**	**00001111**
Number of addresses:	15 + 1 = 16			

Note

The first address in a block is normally not assigned to any device; it is used as the network address that represents the organization to the rest of the world.

Figure 19.6 *Two levels of hierarchy in an IPv4 address*

Each address in the block can be considered as a two-level hierarchical structure: the leftmost *n* bits (prefix) define the network; the rightmost 32 − n bits define the host.

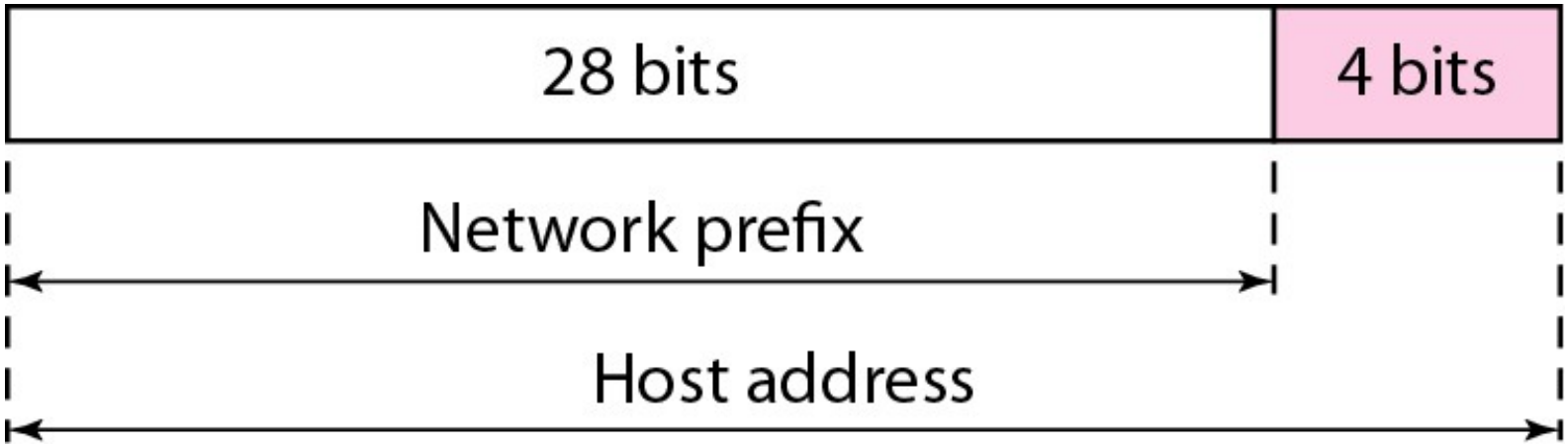


Figure 19.7 *Configuration and addresses in a subnetted network*

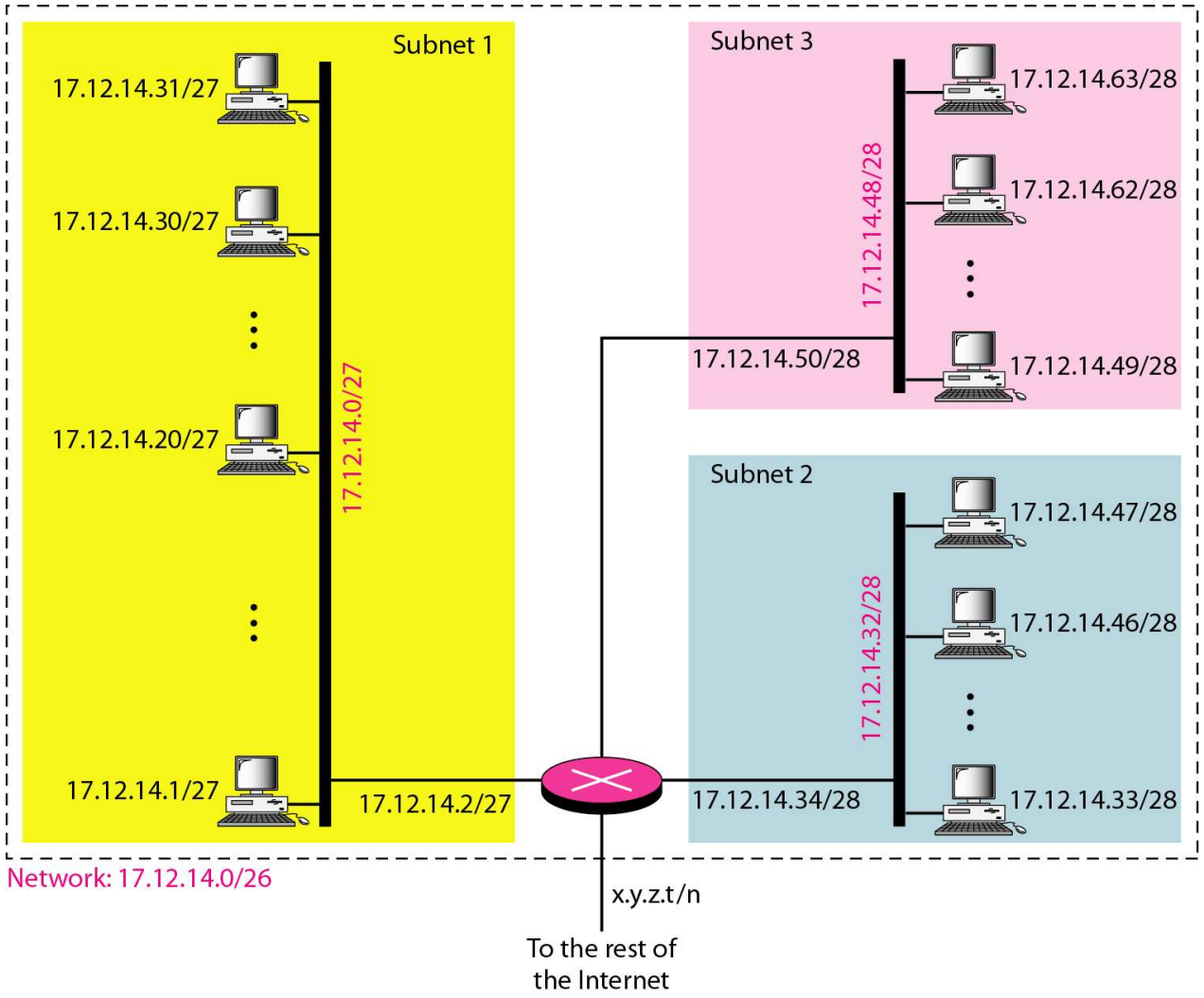
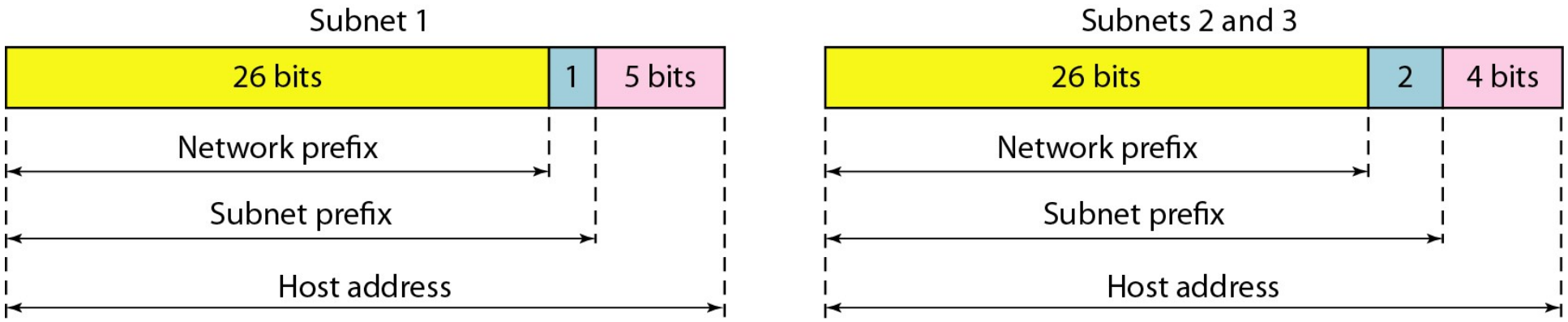


Figure 19.8 *Three-level hierarchy in an IPv4 address*



Example 19.10

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:

a. The first group has 64 customers; each needs 256 addresses.

b. The second group has 128 customers; each needs 128 addresses.

c. The third group has 128 customers; each needs 64 addresses.

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

Example 19.10 (continued)

Solution

Figure 19.9 shows the situation.

Group 1

For this group, each customer needs 256 addresses. This means that 8 ($\log_2 256$) bits are needed to define each address. The prefix length is then $32 - 8 = 24$. The addresses are

1st Customer:	190.100.0.0/24	190.100.0.255/24
2nd Customer:	190.100.1.0/24	190.100.1.255/24
. . .		
64th Customer:	190.100.63.0/24	190.100.63.255/24

Total = $64 \times 256 = 16{,}384$

Example 19.10 (continued)

Group 2

For this group, each customer needs 128 addresses. This means that 7 ($\log_2 128$) bits are needed to define each host. The prefix length is then $32 - 7 = 25$. The addresses are

1st Customer:	*190.100.64.0/25*	*190.100.64.127/25*
2nd Customer:	*190.100.64.128/25*	*190.100.64.255/25*
. . .		
128th Customer:	*190.100.127.128/25*	*190.100.127.255/25*
Total = $128 \times 128 = 16{,}384$		

Example 19.10 (continued)

Group 3

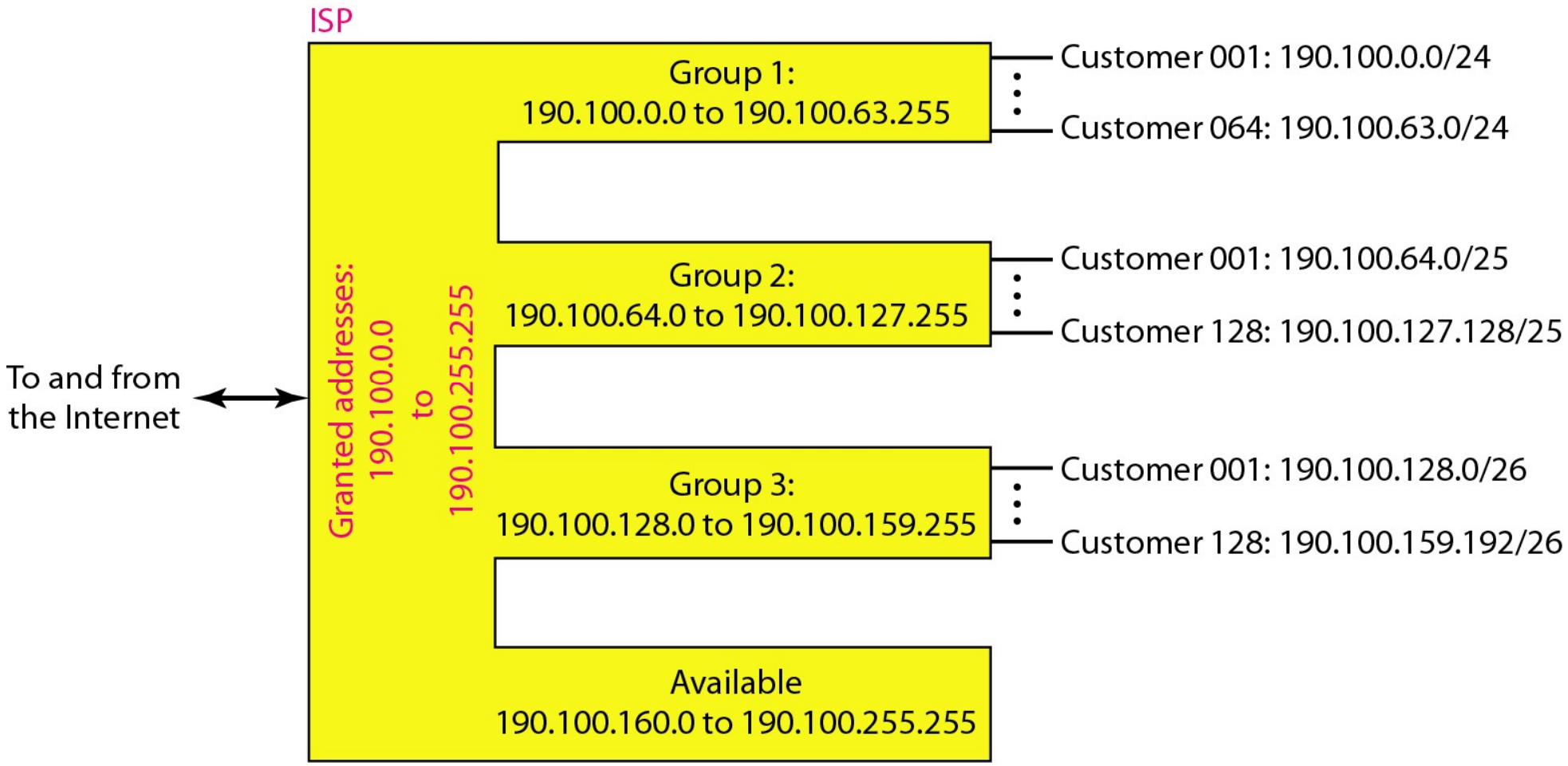
For this group, each customer needs 64 addresses. This means that 6 ($\log_2 64$) bits are needed to each host. The prefix length is then 32 − 6 = 26. The addresses are

1st Customer:	190.100.128.0/26	190.100.128.63/26
2nd Customer:	190.100.128.64/26	190.100.128.127/26
. . .		
128th Customer:	190.100.159.192/26	190.100.159.255/26

Total = 128 × 64 = 8192

Number of granted addresses to the ISP: 65,536
Number of allocated addresses by the ISP: 40,960
Number of available addresses: 24,576

Figure 19.9 *An example of address allocation and distribution by an ISP*



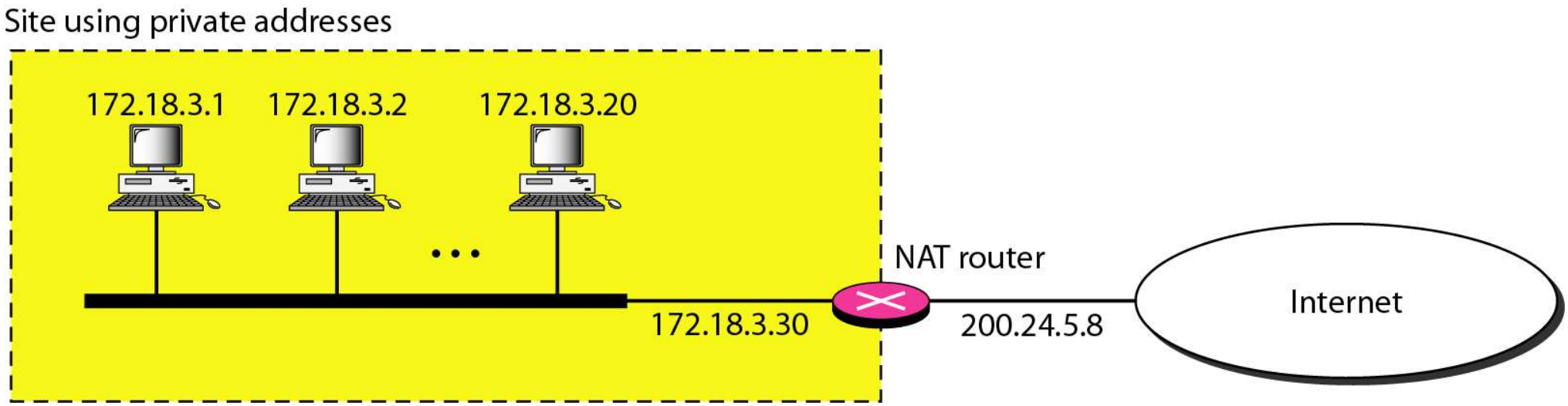
Network Address Translation

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}

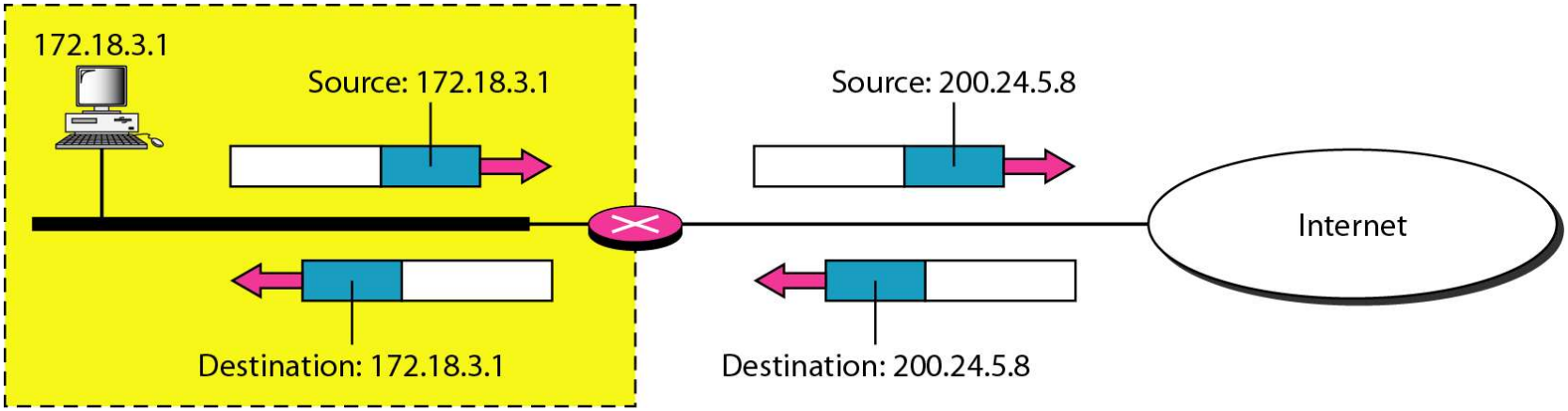
Figure 19.10 *A NAT implementation*

- Private IP addresses are used within a local network for internal communication, while public IP addresses are used to identify devices on the global internet.



- When a device on a private network needs to communicate with the internet, it typically uses a technology called Network Address Translation (NAT) to translate between the private IP address of the device and the public IP address of the router.

Figure 19.11 *Addresses in a NAT*



Outgoing Packet:

- The packet passes through the network's NAT device (typically a router).
- The NAT device replaces the private source IP address with its own public IP address.
- The NAT device keeps track of the translation in a NAT table, including the original private source IP address and port, the translated public source IP address and port, and the destination's IP address and port.

The packet, now with the NAT device's public IP address as the source, is sent to the destination on the public internet.

Incoming Packet:

- NAT Translation (Reverse):
  - The NAT device uses the information stored in its NAT table to reverse the translation. It replaces the public source IP address with the original private source IP address and port.
- Delivery to Private Network:
  - When the response packet comes back from the public internet, it has the NAT device's public IP address as the source.
  - The packet, with the private source IP address restored, is delivered to the original private device that initiated the communication.

Figure 19.12 *NAT address translation*

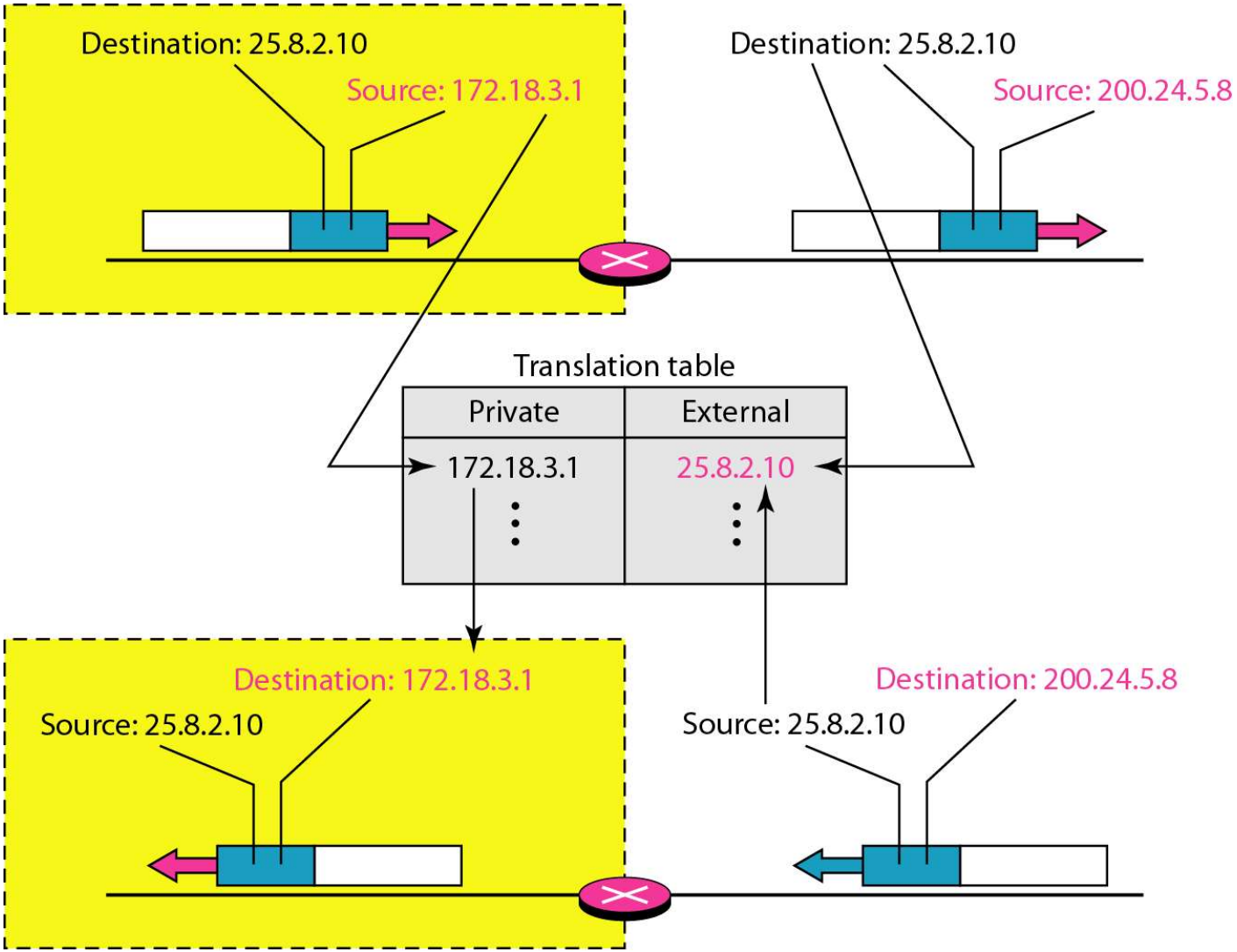


Table 19.4 *Five-column translation table*

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
. . .	. . .	. . .	. . .	. . .

19-2 IPv6 ADDRESSES

Despite all short-term solutions, address depletion is still a long-term problem for the Internet. This and other problems in the IP protocol itself have been the motivation for IPv6.

Topics discussed in this section:

Structure
Address Space

Figure 19.14 *IPv6 address in binary and hexadecimal colon notation*

An IPv6 address is 128 bits long.

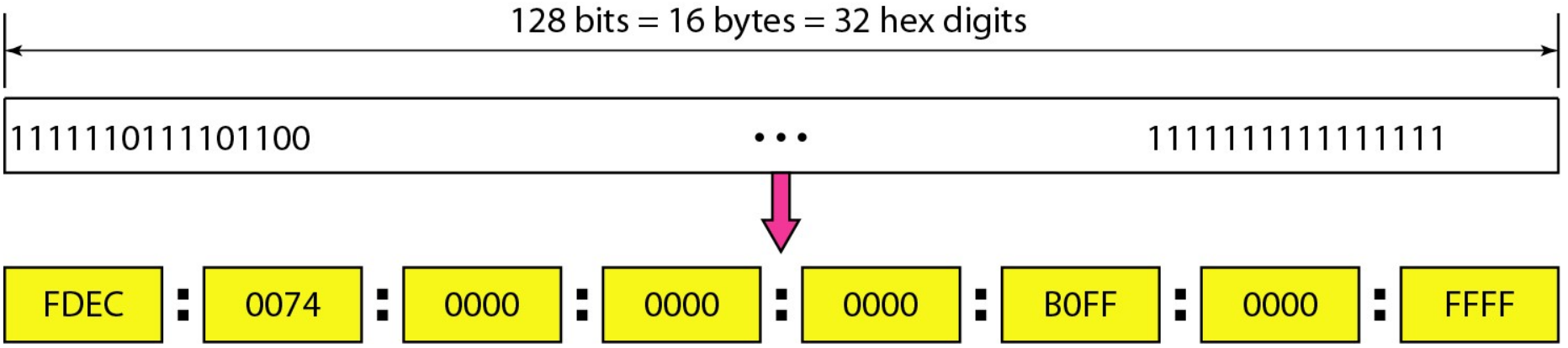
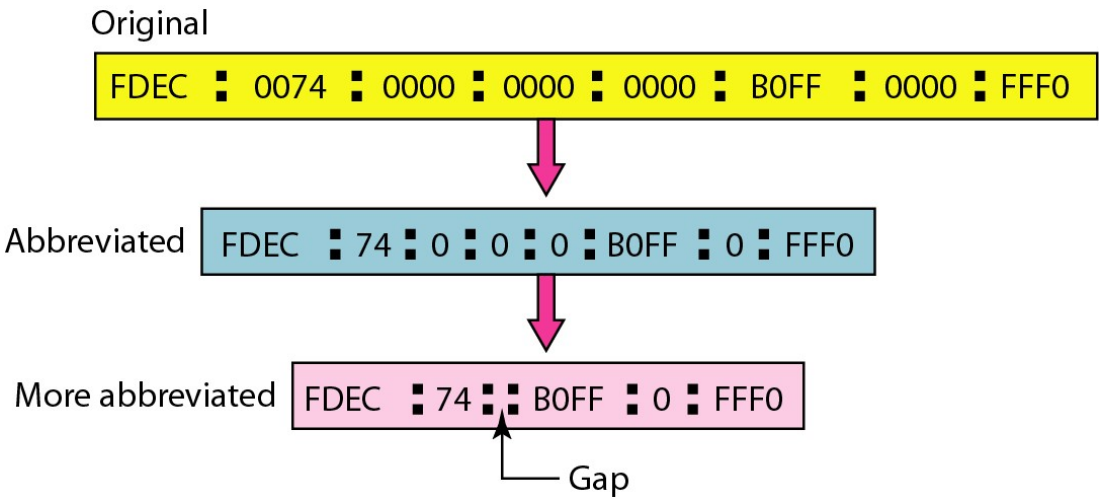


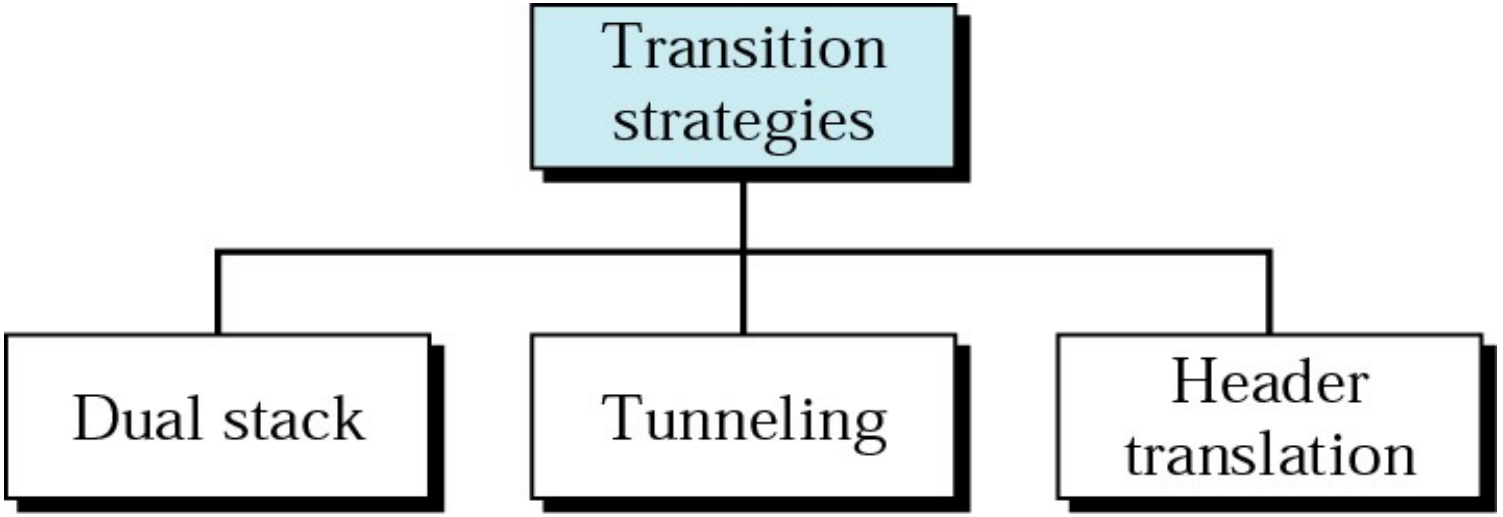
Figure 19.15 *Abbreviated IPv6 addresses*

IPv6 addresses might be specified in two other shortened formats:

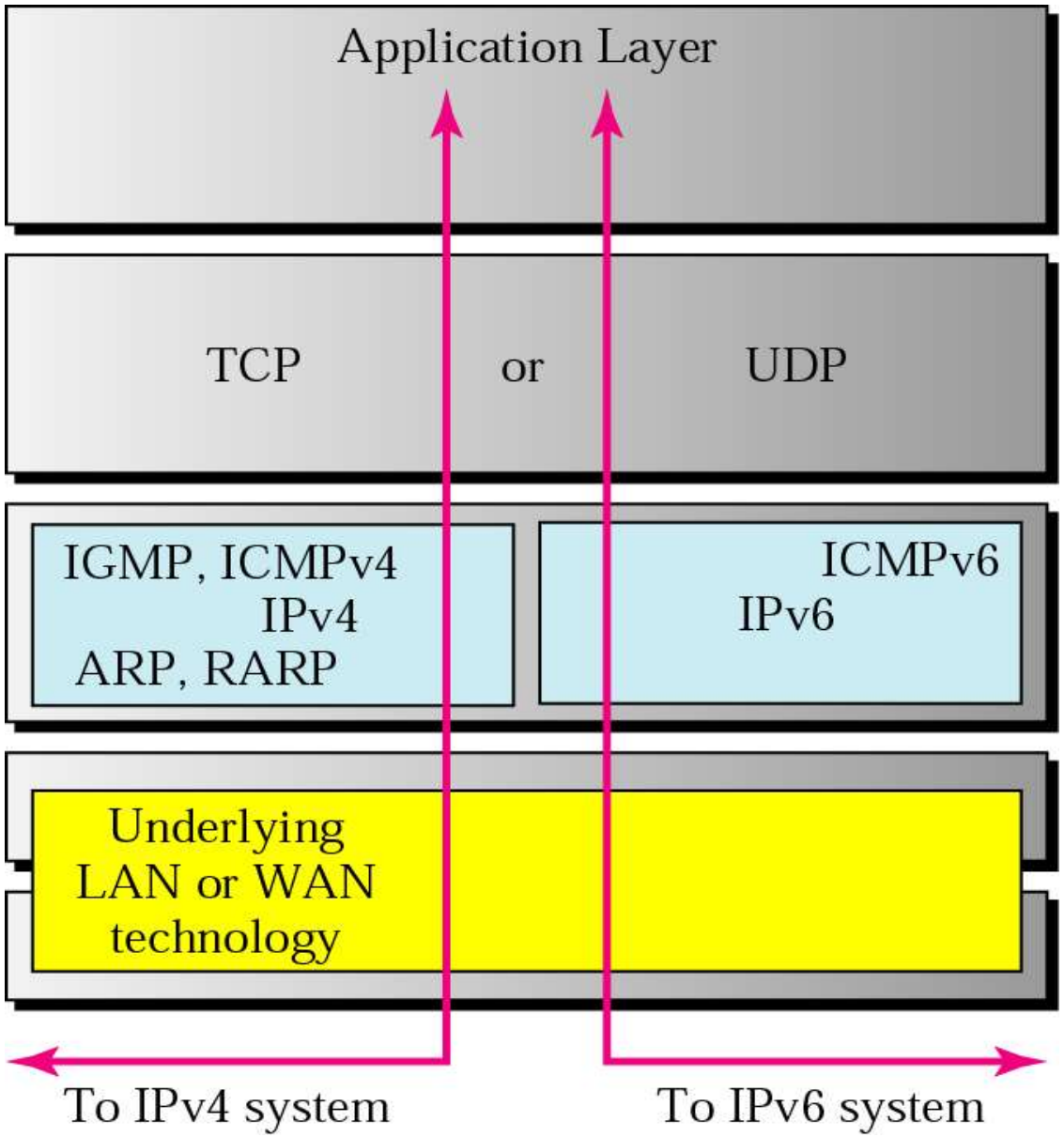
- Omit leading zeros: Specify IPv6 addresses by omitting leading zeros
- Double colon: Specify IPv6 addresses by using double colons (::) in place of a series of zeros



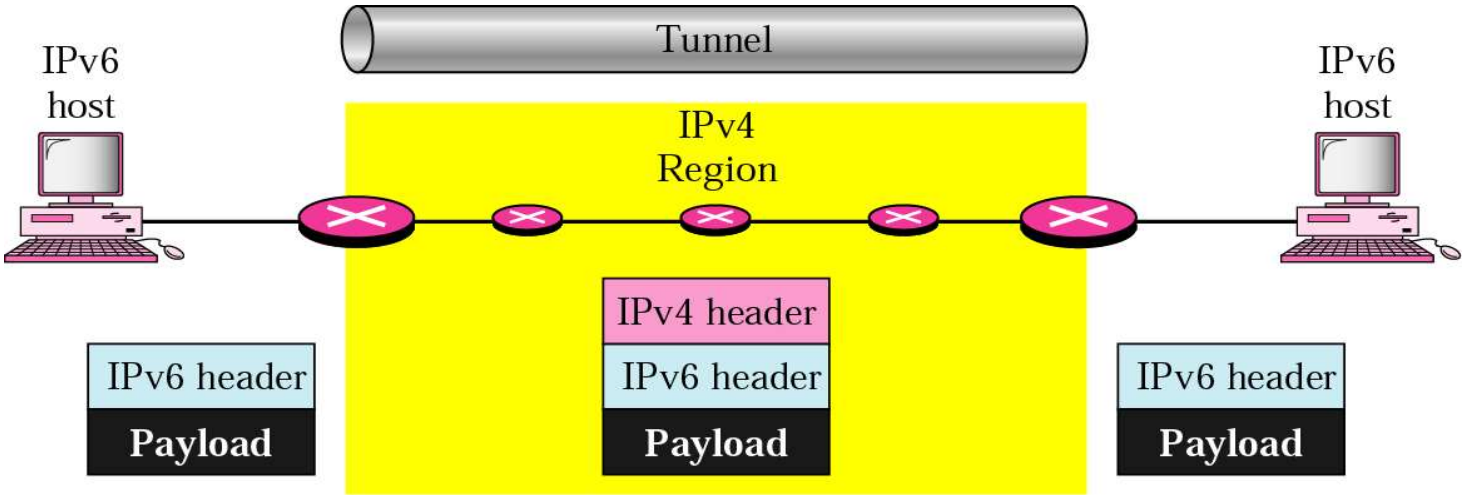
Three transition strategies



Dual Stack


Application Layer
TCP
or
UDP
IGMP, ICMPv4
IPv4
ARP, RARP
ICMPv6
IPv6
Underlying
LAN or WAN
technology
To IPv4 system
To IPv6 system

Tunneling


Tunnel
IPv6
host
IPv6
host
IPv4
Region
IPv4 header
IPv6 header
Payload
IPv6 header
Payload
IPv6 header
Payload

Header translation