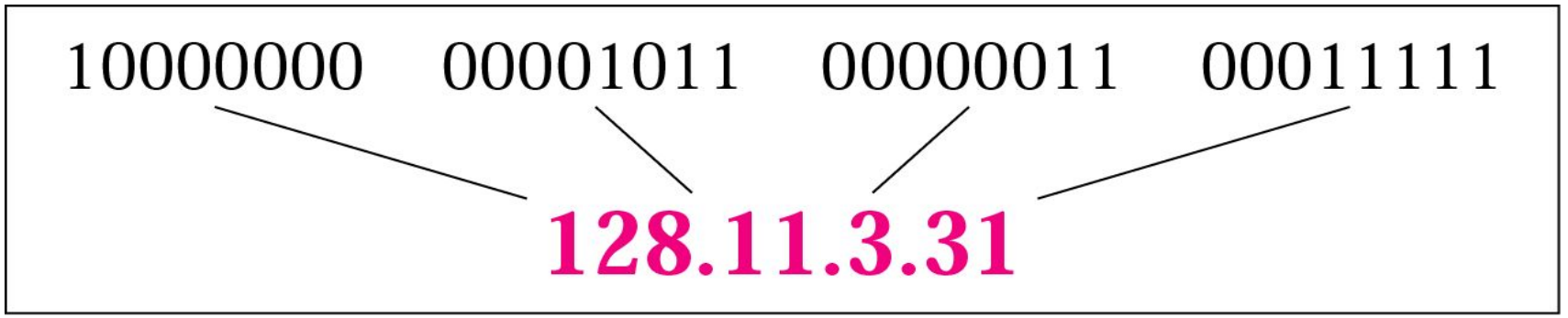
UNIT – 3

An IP address is a 32-bit address.

The IP addresses are unique and universal.

Figure 19.9 Dotted-decimal notation



Example 1

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

a. 10000001 00001011 00001011 11101111

b. 11111001 10011011 11111011 00001111

Solutio

We replace each group of 8 bits with its equivalent decimal number (see Appendix B) and add dots for separation:

a. 129.11.11.239

b. 249.155.251.15

Example 2

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

a. 111.56.45.78

b. 75.45.34.78

Solutio

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

a. 01101111 00111000 00101101 01001110
b. 01001011 00101101 00100010 01001110

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

Figure 19.10 Finding the class in binary notation

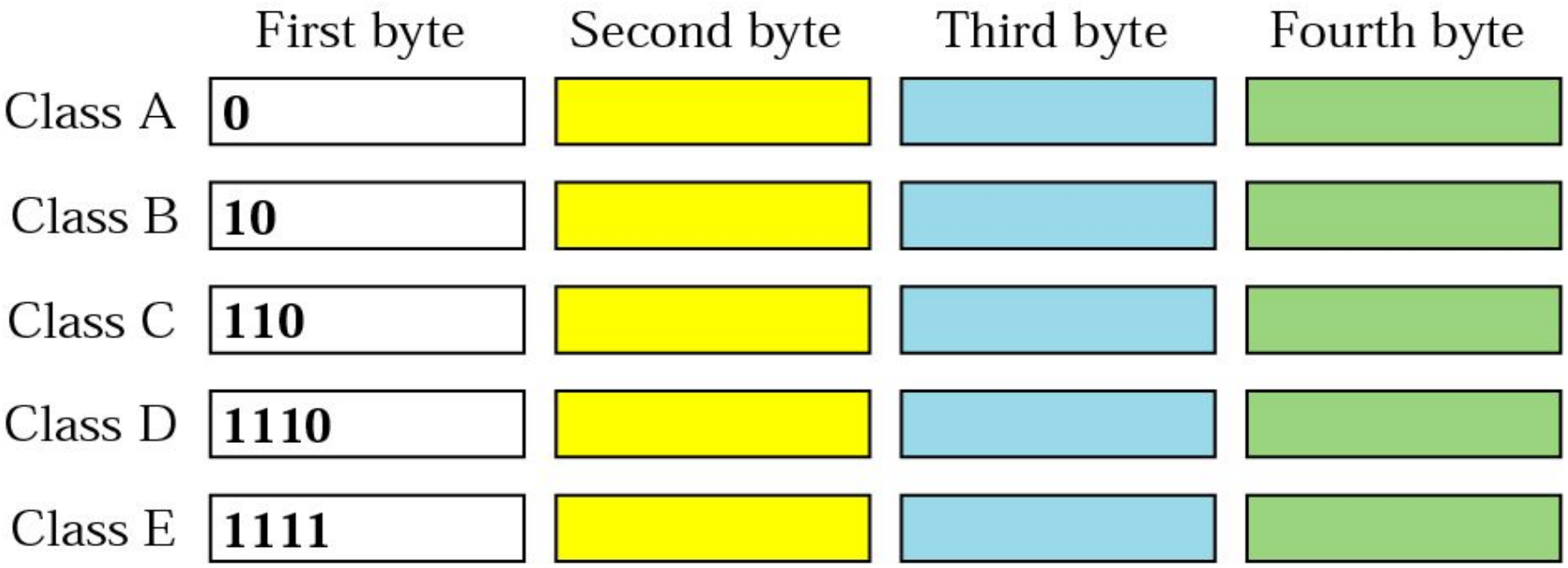
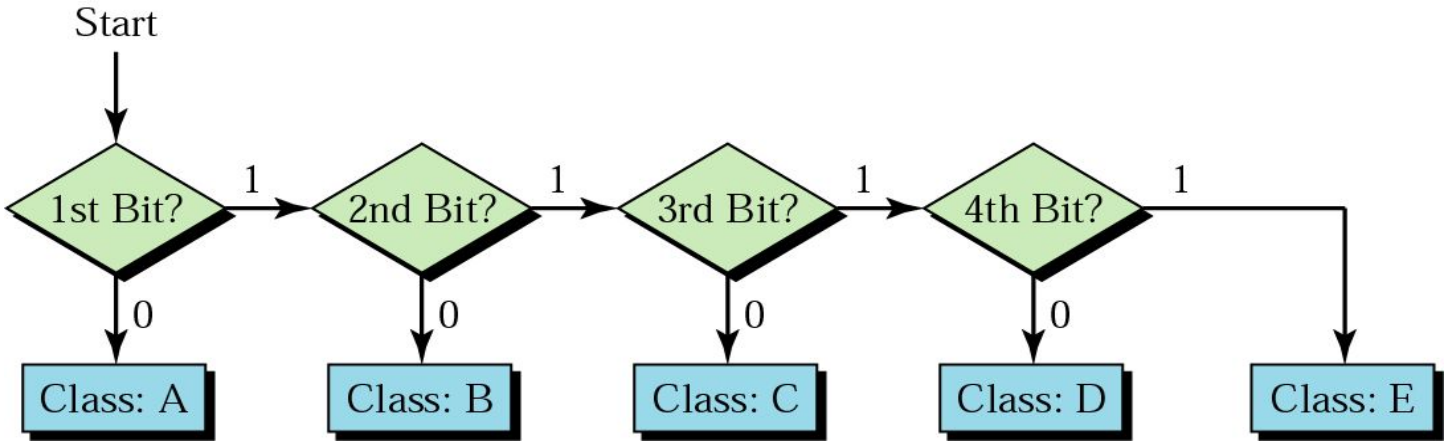


Figure 19.11 Finding the address class



Example 3

Find the class of each address:

a. **0**0000001 00001011 00001011 11101111

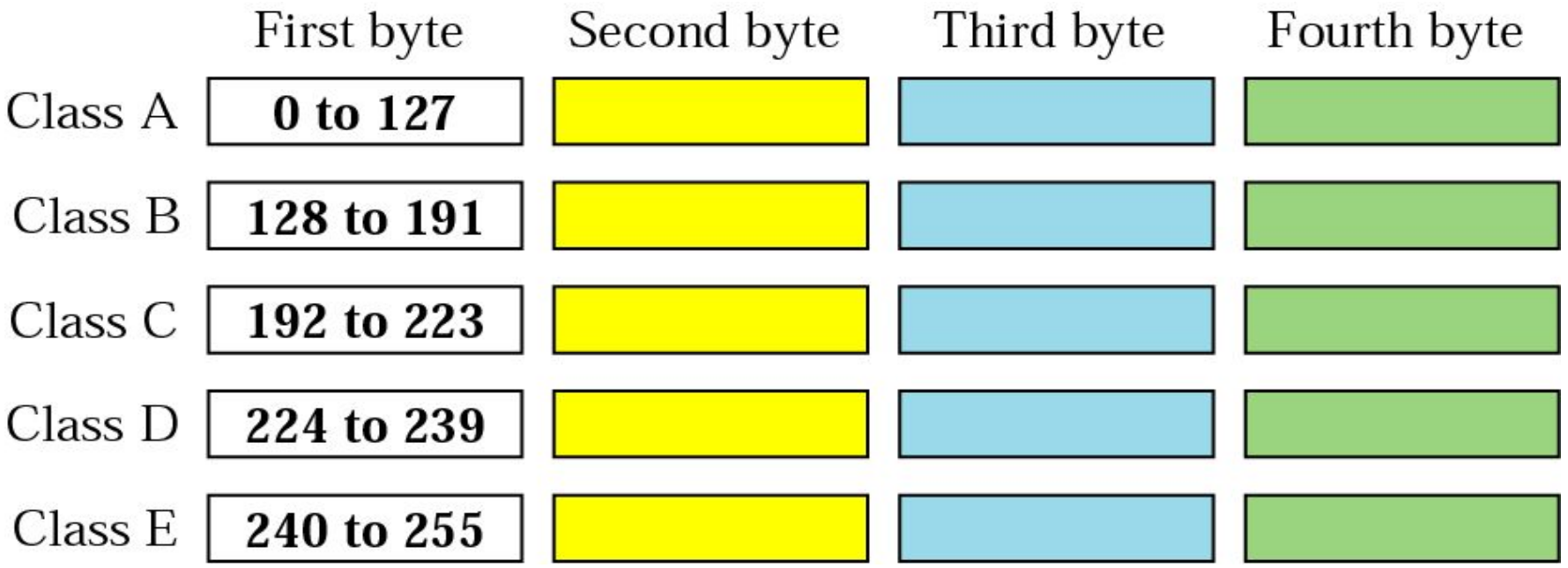
b. **1111**0011 10011011 11111011 00001111

Solutio

See the procedure in Figure 19.11.

a. The first bit is 0; this is a class A address.
b. The first 4 bits are 1s; this is a class E address.

Figure 19.12 Finding the class in decimal notation



Example 4

Find the class of each address:

a. **227**.12.14.87

b. **252**.5.15.111

c. **134**.11.78.56

Solution

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

Figure 19.13 Netid and hostid

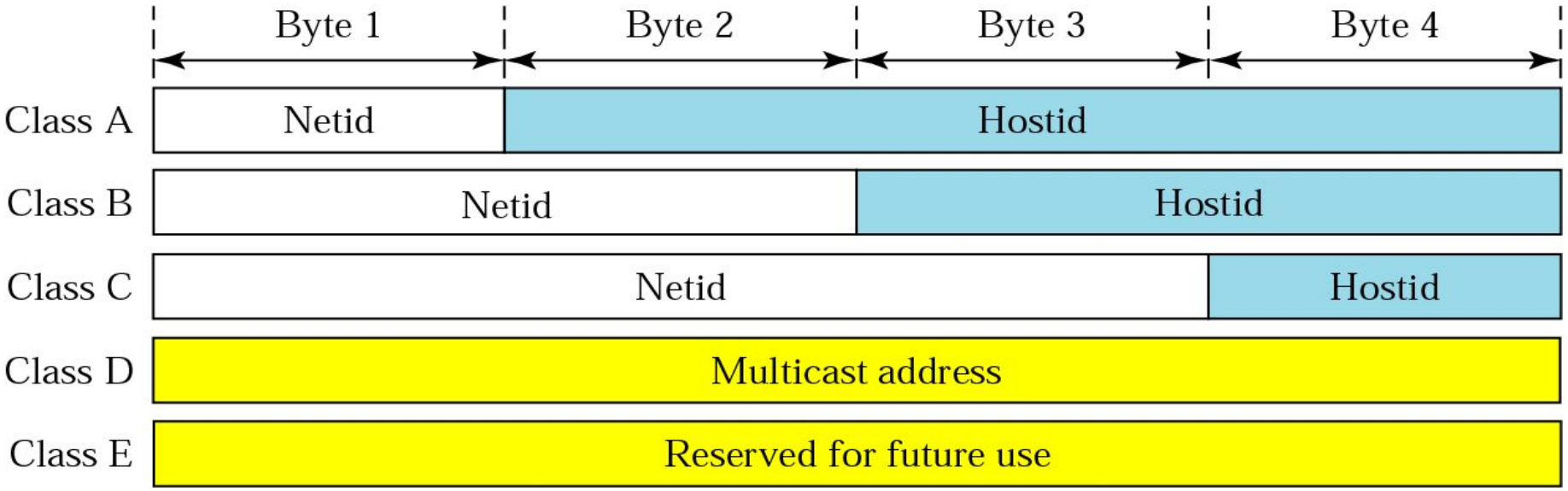
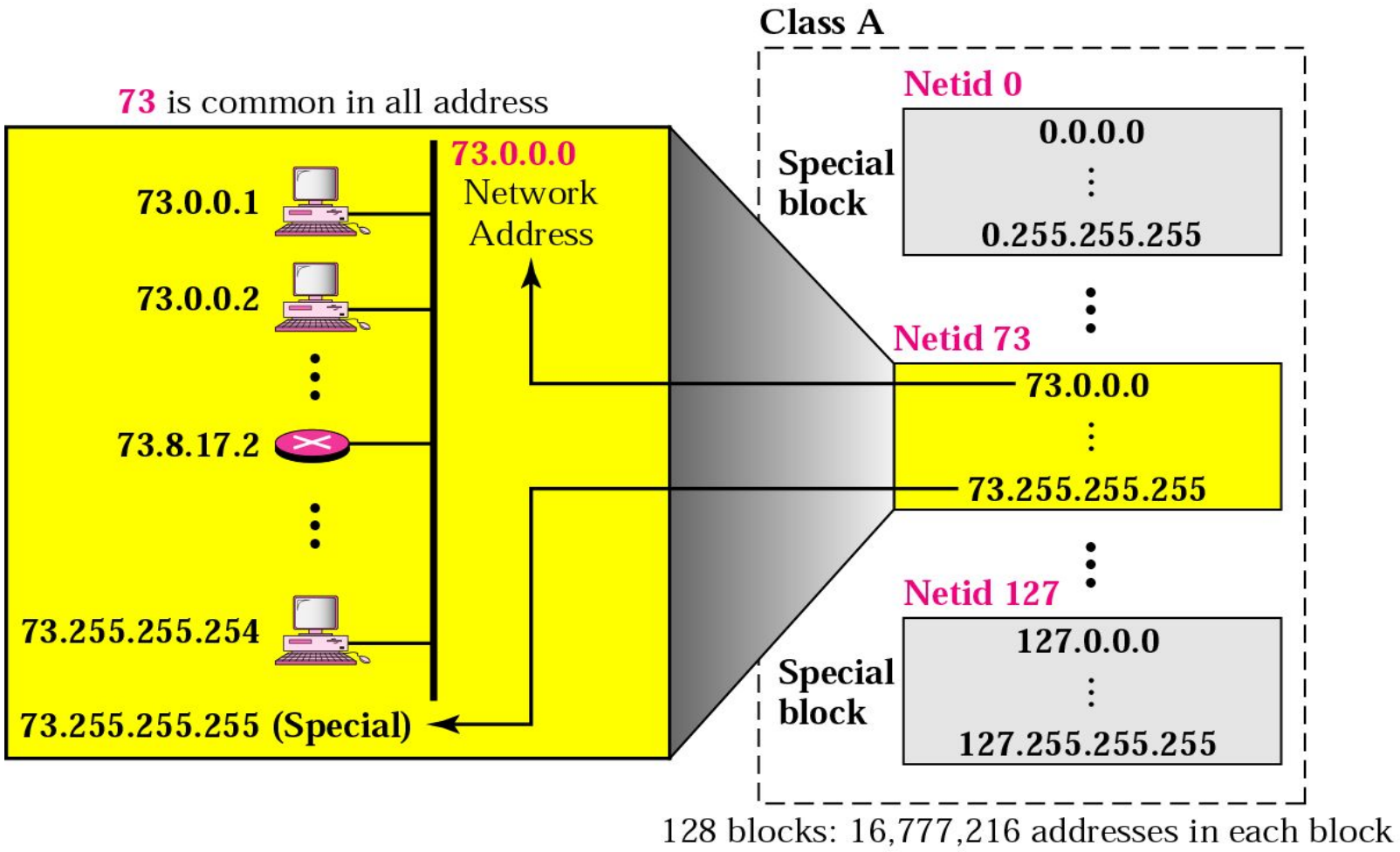
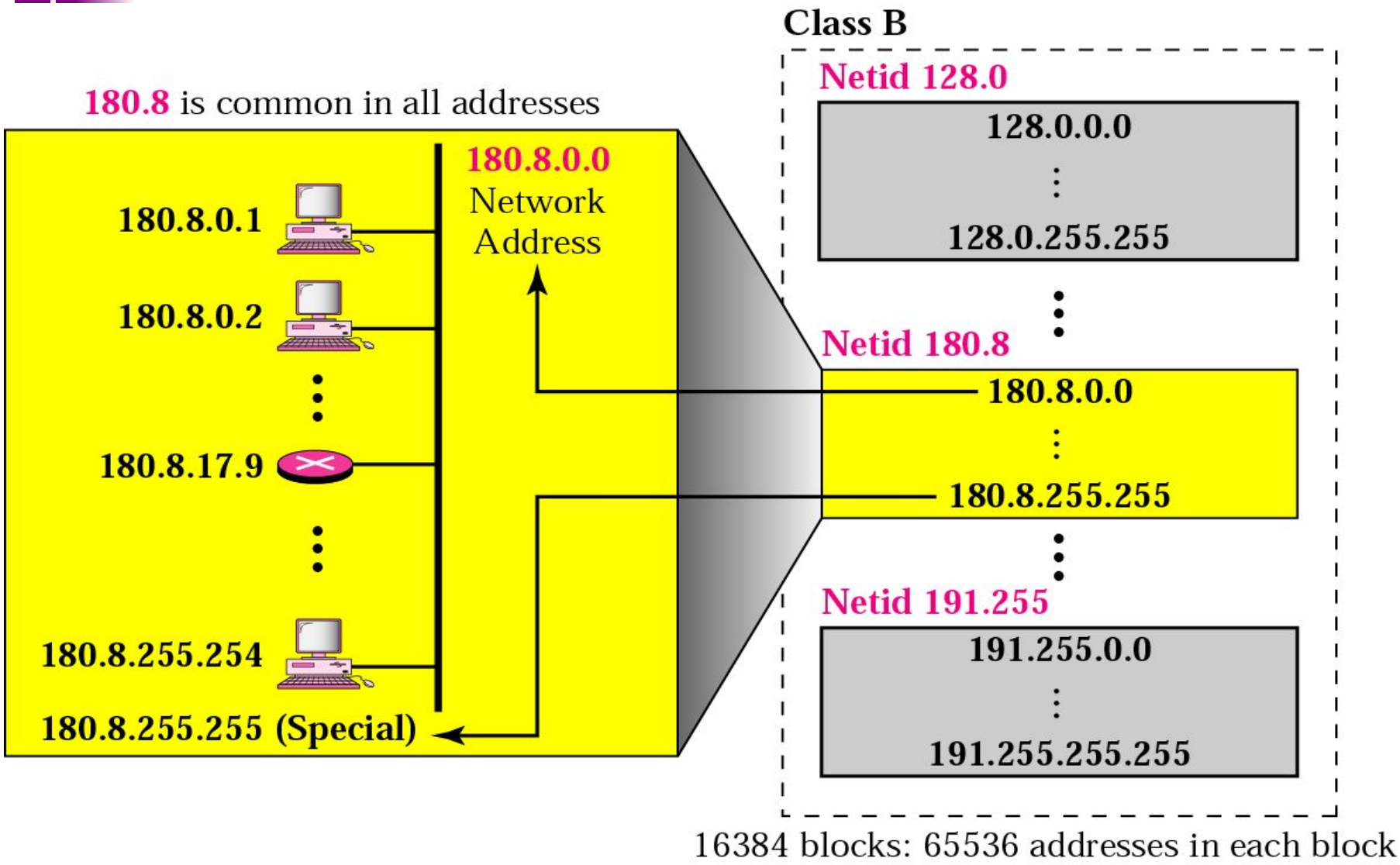


Figure 19.14 Blocks in class A



Millions of class A addresses are wasted.

Figure 19.15 Blocks in class B



Many class B addresses are wasted.

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

Figure 19.16 Blocks in class C

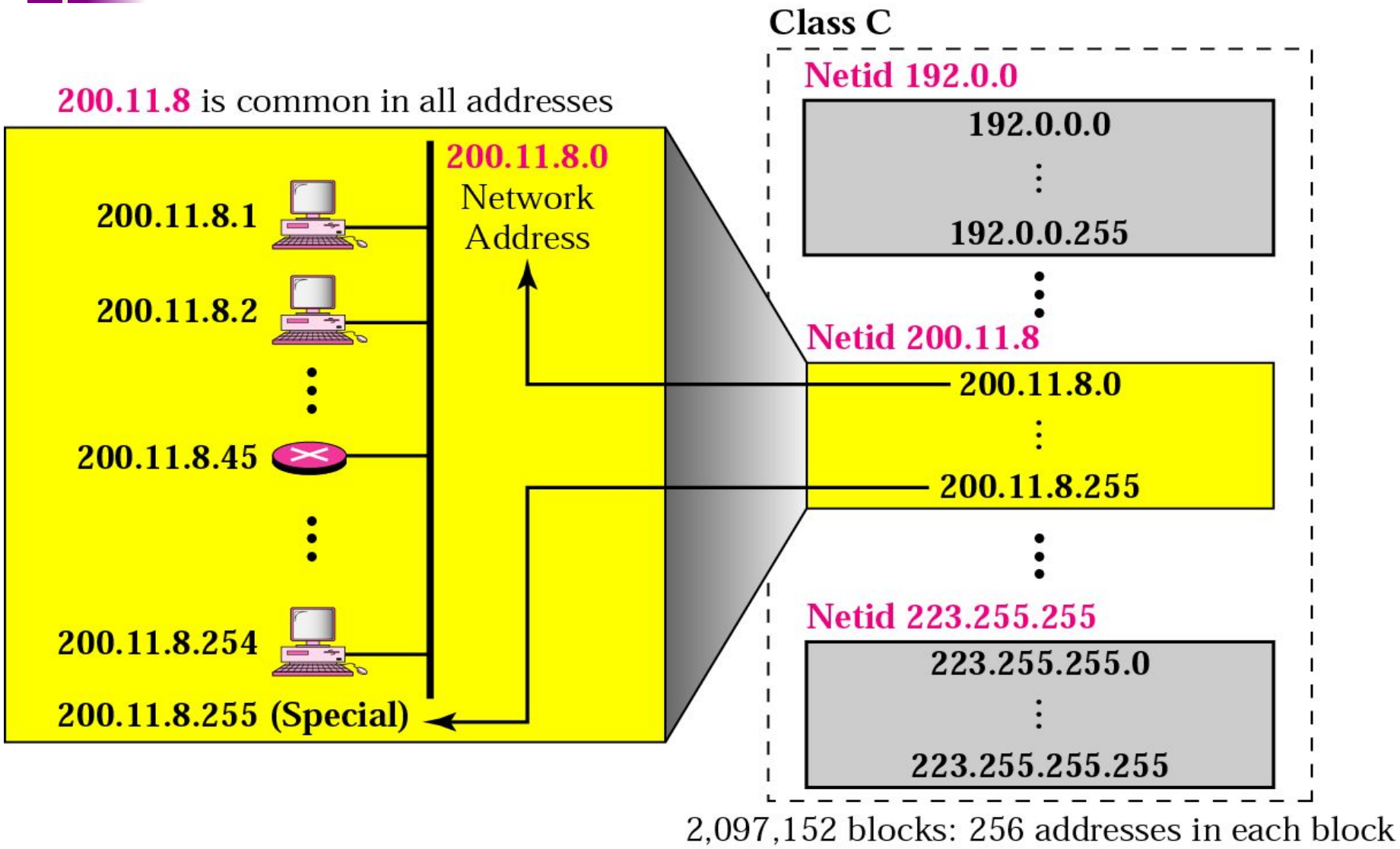
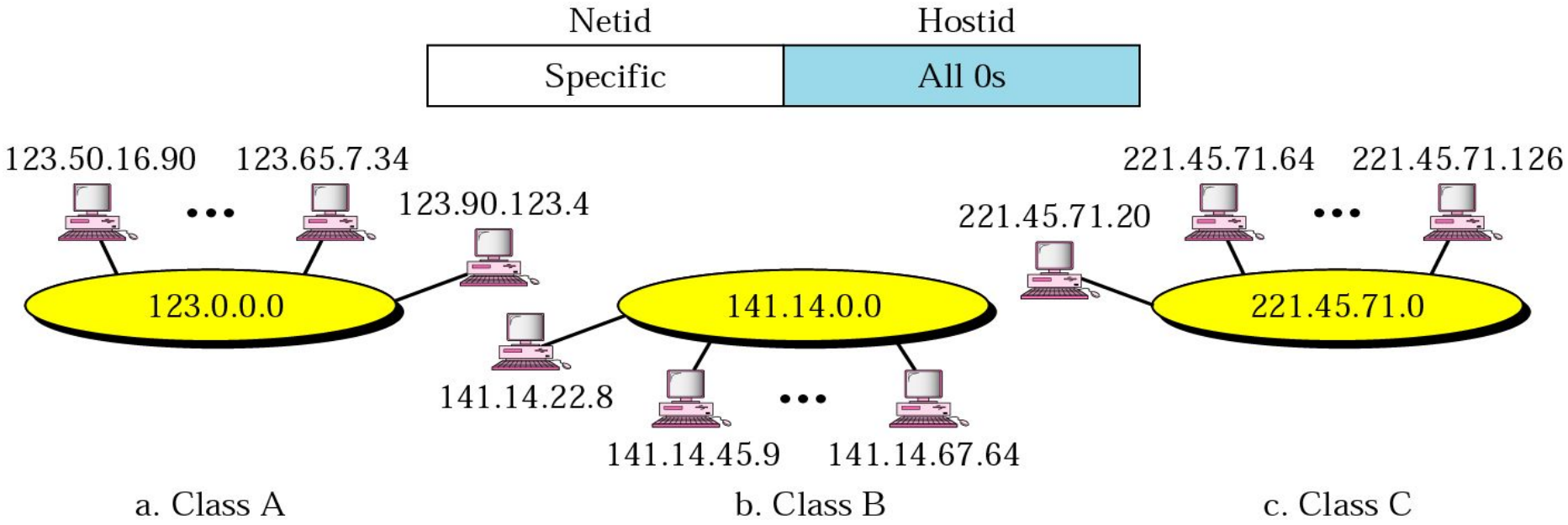


Figure 19.17 Network address



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

Example 5

Given the address 23.56.7.91, find the network address.

Solutio

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

Example 6

Given the address 132.6.17.85, find the network address.

Solutio

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

Example 7

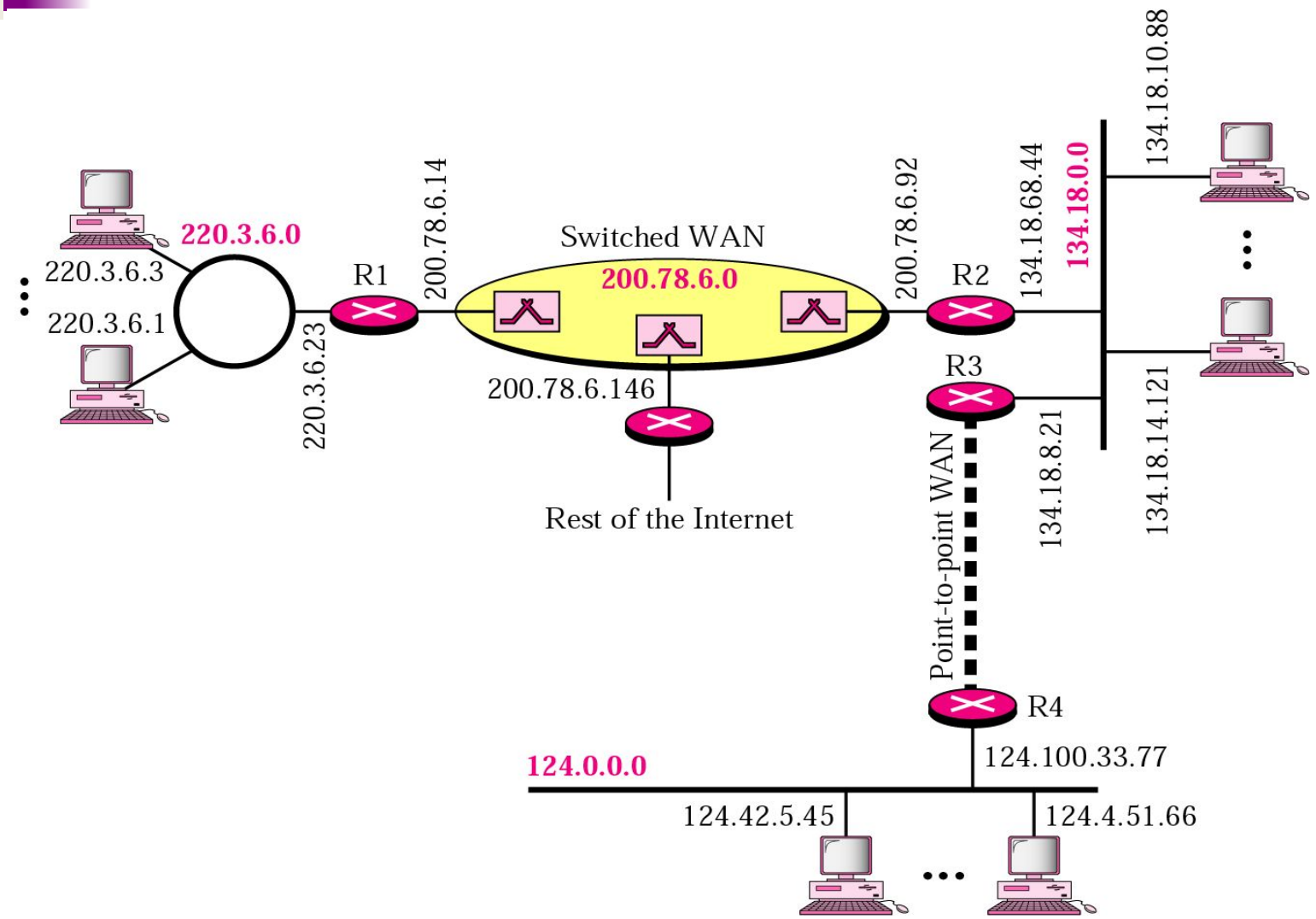
Given the network address 17.0.0.0, find the class.

Solutio

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

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

Figure 19.18 Sample internet



IP addresses are designed with two levels of hierarchy.

Figure 19.19 A network with two levels of hierarchy

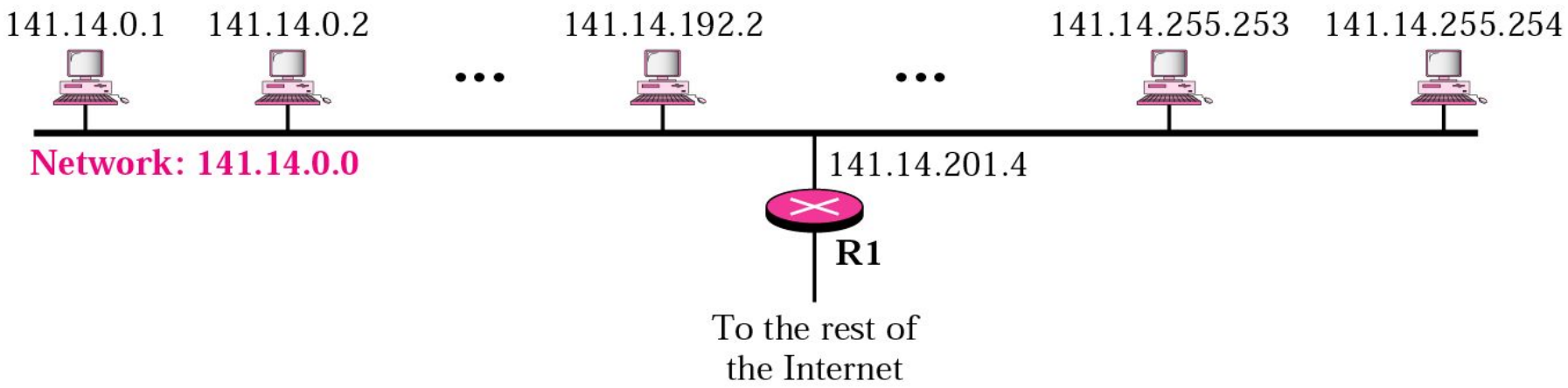


Figure 19.20 A network with three levels of hierarchy (subnetted)

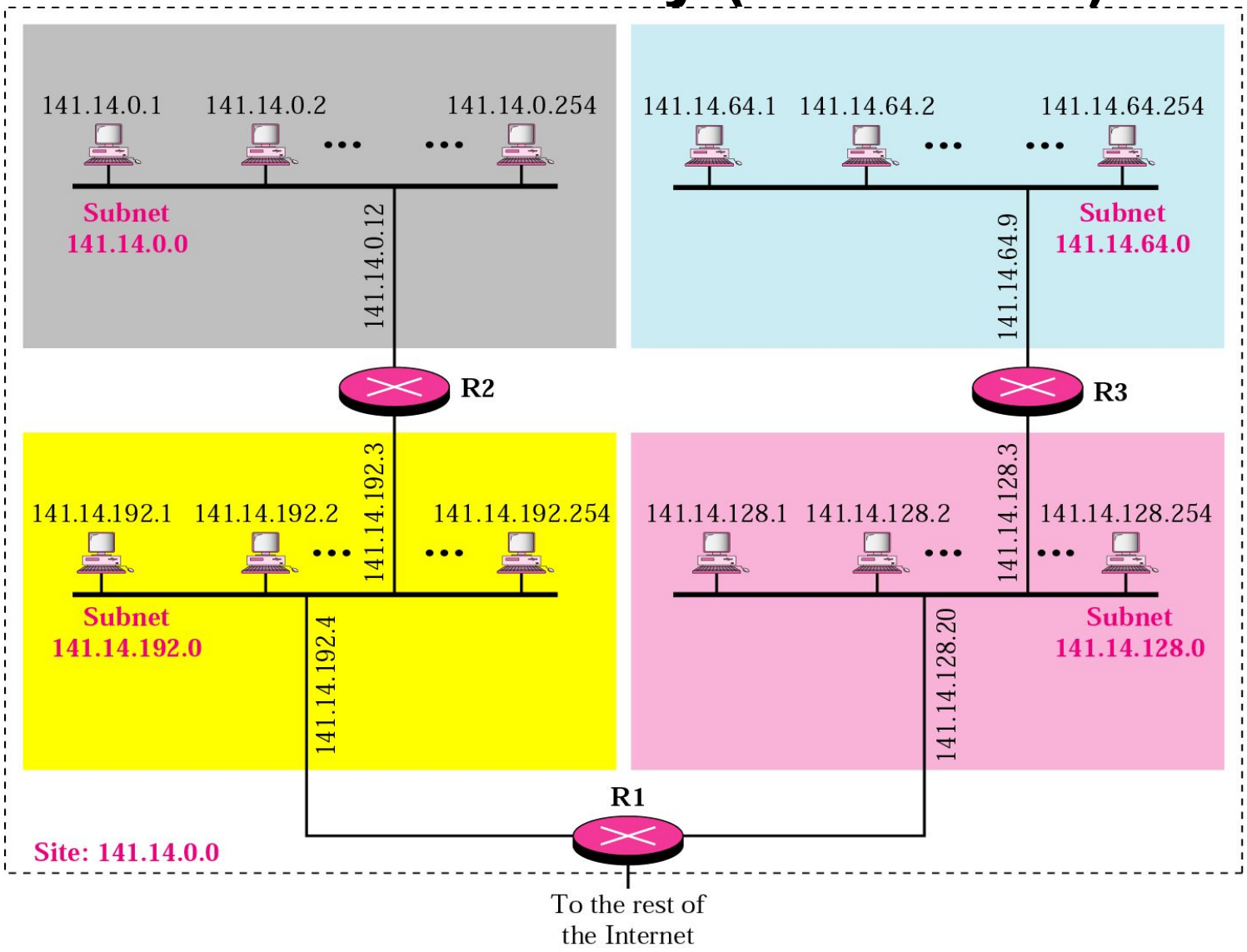
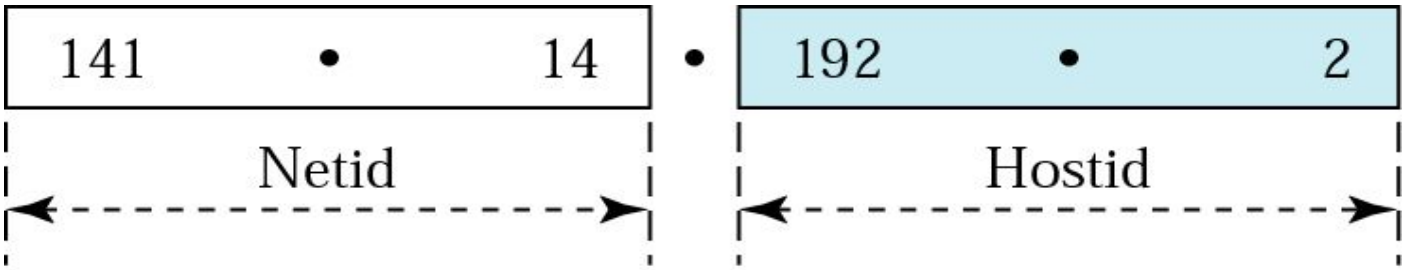
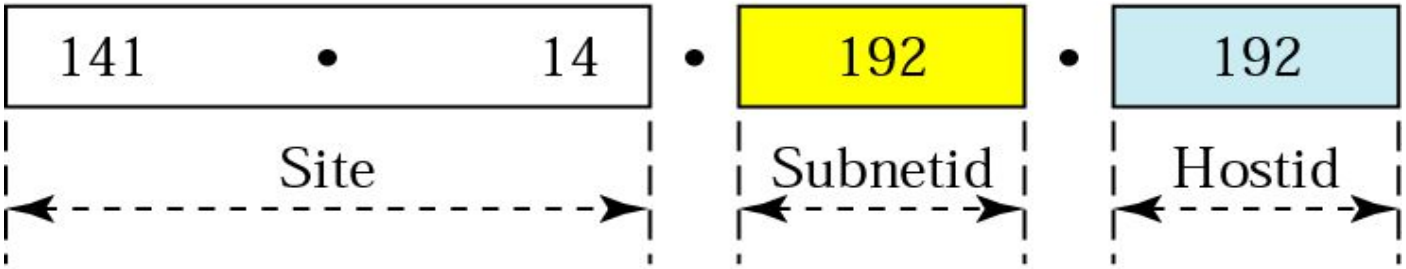


Figure 19.21 Addresses in a network with and without subnetting



a. Without subnetting



b. With subnetting

Figure 19.22 Hierarchy concept in a telephone number

The network address can be found by applying the default mask to any address in the block (including itself).
It retains the netid of the block and sets the hostid to 0s.

Table 19.1 Default

Class	In Binary	In Dotted-Decimal	Using Slash
A	11111111 00000000 00000000 00000000	255.0.0.0	/8
B	11111111 11111111 00000000 00000000	255.255.0.0	/16
C	11111111 111111111 11111111 00000000	255.255.255.0	/24

Example 8

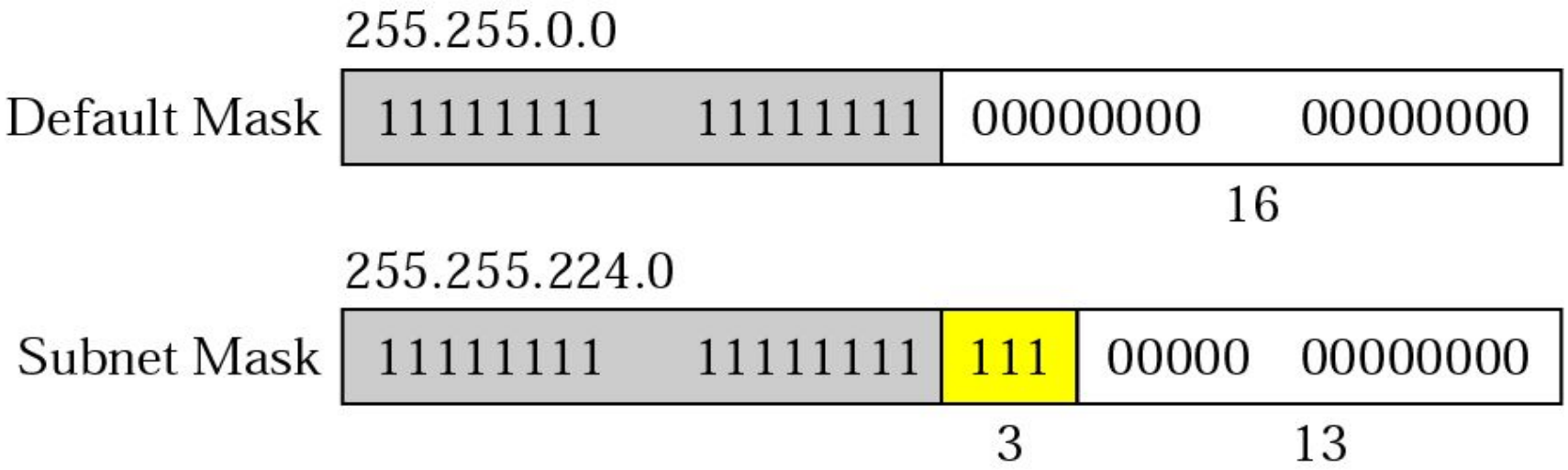
A router outside the organization receives a packet with destination address 190.240.7.91. Show how it finds the network address to route the packet.

Solutio

The router follows three steps:

1. The router looks at the first byte of the address to find the class. It is class B.
2. The default mask for class B is 255.255.0.0. The router ANDs this mask with the address to get 190.240.0.0.
3. The router looks in its routing table to find out how to route the packet to this destination. Later, we will see what happens if this destination does not exist.

Figure 19.23 Subnet mask



Example 9

A router inside the organization receives the same packet with destination address 190.240.33.91. Show how it finds the subnetwork address to route the packet.

Solutio

The router follows three steps:

1. **The router must know the mask. We assume it is /19, as shown in Figure 19.23.**
2. **The router applies the mask to the address, 190.240.33.91. The subnet address is 190.240.32.0.**
3. **The router looks in its routing table to find how to route the packet to this destination. Later, we will see what happens if this destination does not exist.**

IP Network Addressing

- INTERNET world's largest public data network, doubling in size every nine months
- IPv4, defines a 32-bit address - 2^{32} (4,294,967,296) IPv4 addresses available
- The first problem is concerned with the eventual depletion of the IP address space.
- Traditional model of classful addressing does not allow the address space to be used to its maximum potential.

Techniques to reduce address shortage in IPv4

- Subnetting
- Classless Inter Domain Routing (CIDR)
- Network Address Translation (NAT)

Subnetting

- Three-level hierarchy: network, subnet, and host.
- The extended-network-prefix is composed of the classful network-prefix and the subnet-number
- The extended-network-prefix has traditionally been identified by the subnet mask

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.

Assume the blocks of IPs are sequentially assigned. 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 (log2 256) bits are needed to define each host. The prefix length is then 32 − 8 = 24. The addresses are

Example 19.10 (continued)

Group 2

For this group, each customer needs 128 addresses. This means that 7 (log2 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 × 128 = 16,384

Example 19.10 (continued)

Group 3

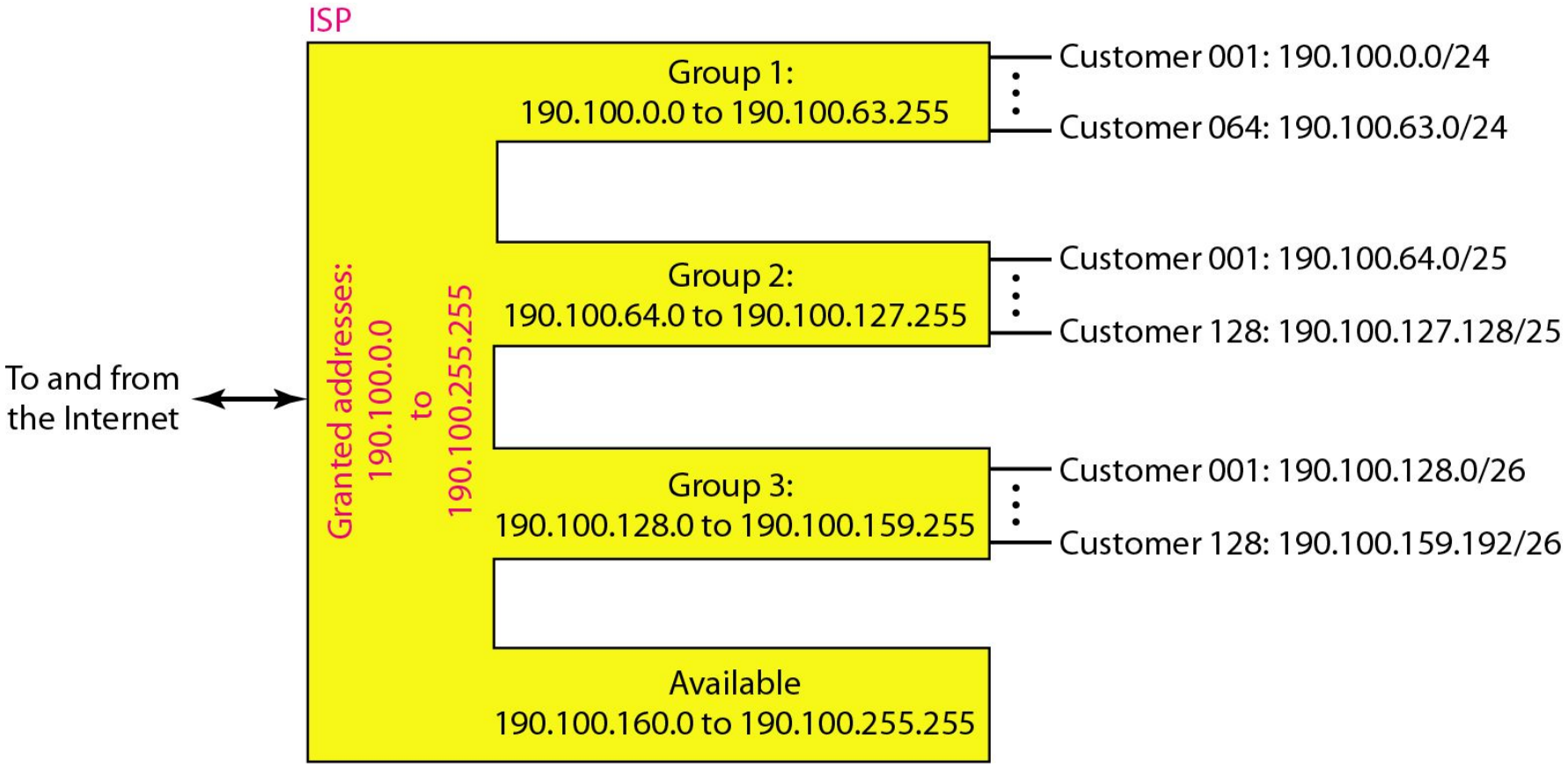
For this group, each customer needs 64 addresses. This means that 6 ($\log_2 64$) bits are needed to each host. The prefix length is then 32 − 6 = 26. The addresses are

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*



Another Example on Subnetting

An ISP needs to allocate three subnets: Subnet 1, Subnet 2, and Subnet 3 with its acquired IP block of 223.1.17.0/24. Subnet 1 is required to support 63 interfaces, Subnet 2 is to support at least 40 interfaces, and Subnet 3 is to support at least 95 interfaces. In addition, values of IP addresses have the relationship: Subnet 1 < Subnet 2 < Subnet 3.

Provide three network addresses (of the form a.b.c.d/x) that satisfy these constraints.

Subnetting

223.1.17.0/24, ip addresses are 2^(32-24) = 256

Subnet 1 needs 2^6=64, 223.1.17.0/26
last address: 223.1.17.63
Subnet 2 needs 2^6=64, 223.1.17.64/26
last address: 223.1.17.127

Subnet 3 needs 2^7 = 128, 223.1.17.128/25

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}

Home used wireless router usually uses 192.168.1.0/24 or 192.168.0.0/24 IP block

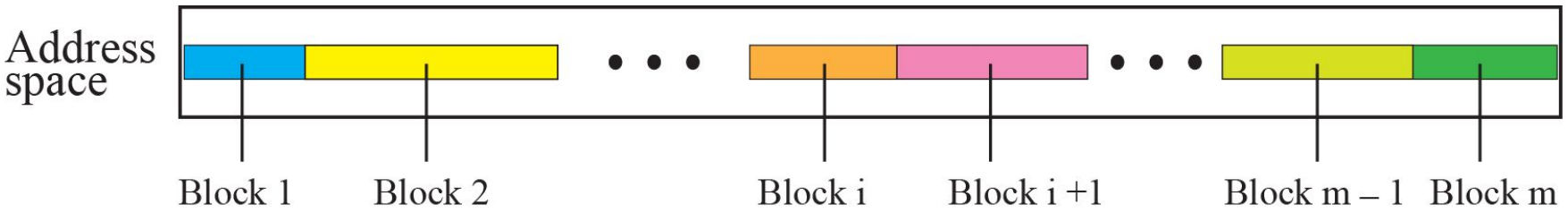
CLASSLESS ADDRESSING

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**

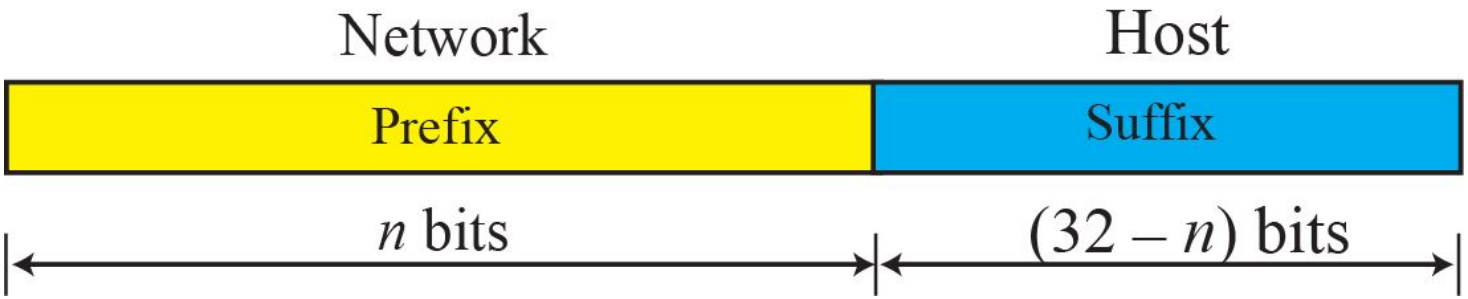
Figure 5.27 *Variable-length blocks in classless addressing*



Note

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

Figure 5.28 *Prefix and suffix*



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?

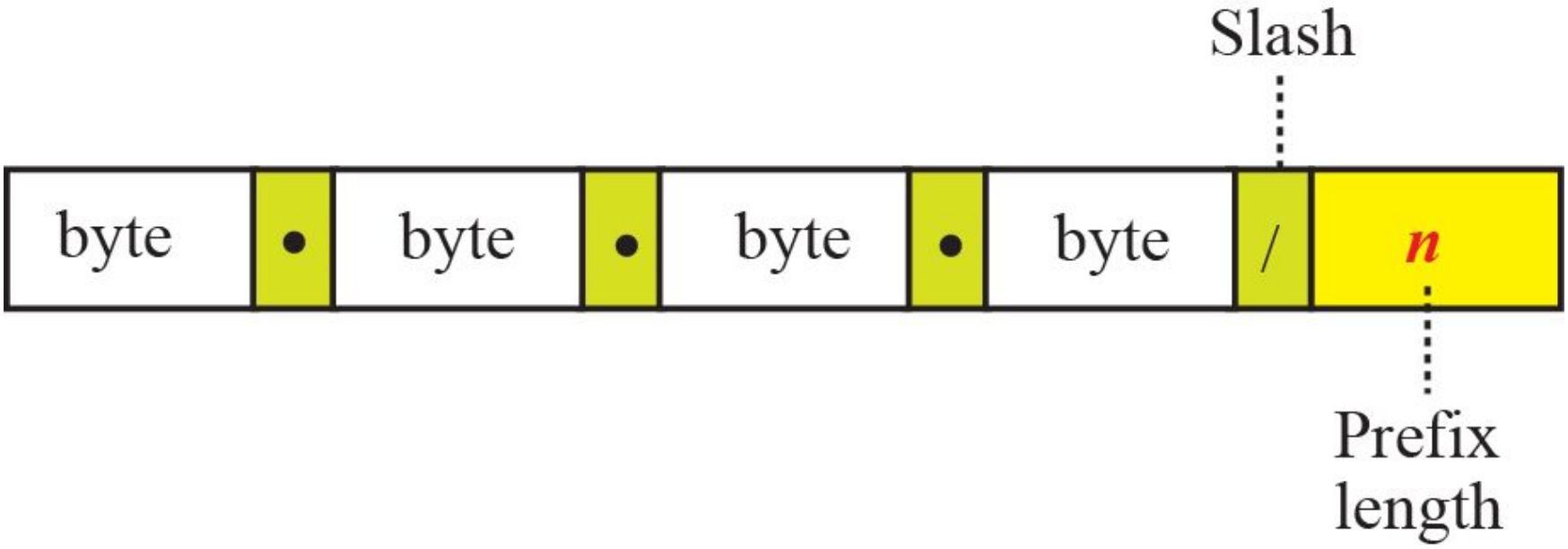
Solution

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 = 2^{10}).

b. The prefix length for the block is calculated as n = 32 − $\log_2 1024$ = 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.

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).

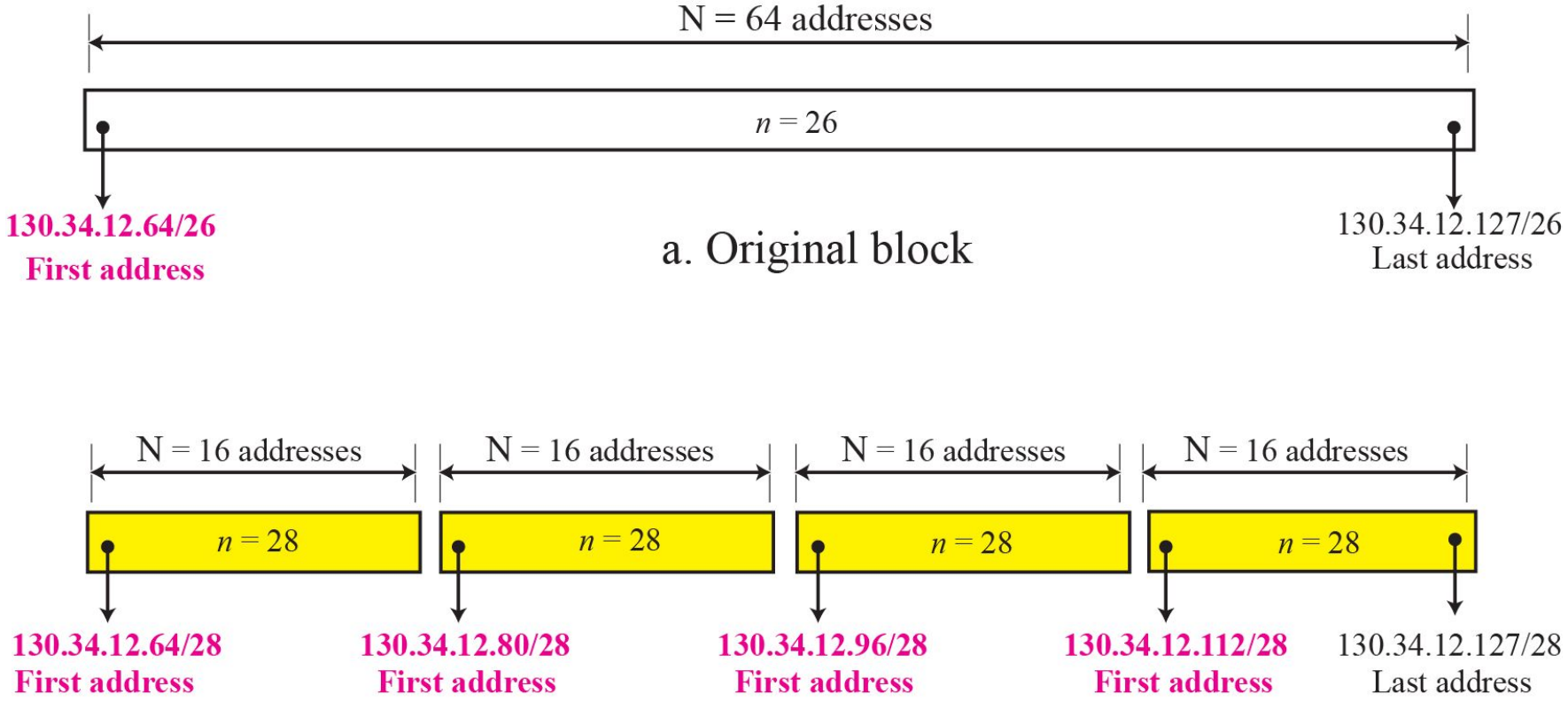
2. The subnetwork mask for each subnetwork is:

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

Example 5.32 *Continued*

3. We grant 16 addresses to each subnet starting from the first available address. Figure 5.30 shows the sub block 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

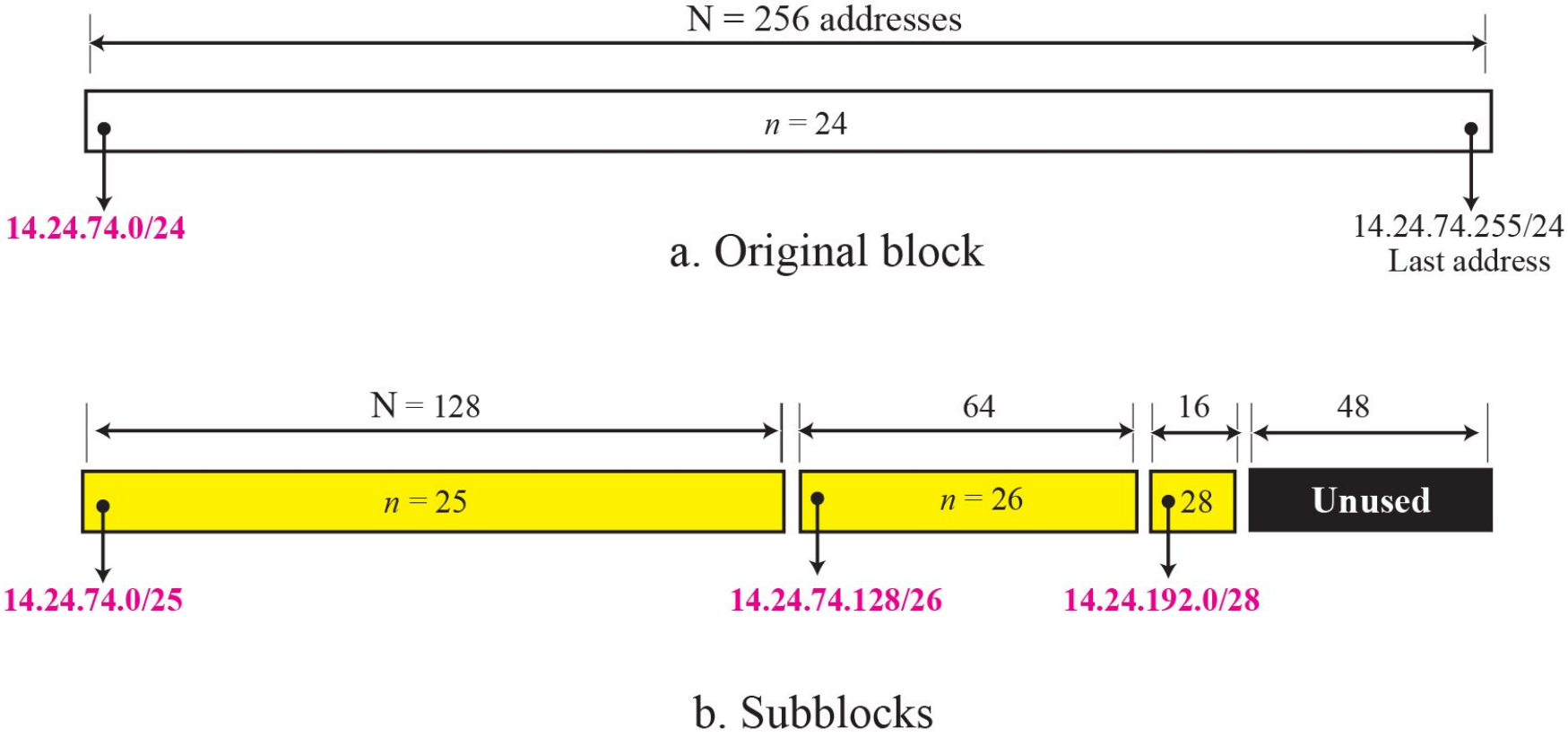
b. The number of addresses in the second subblock is not a power of 2 either. We allocate 64 addresses. The subnet mask is 26. The first address in this block is 14.24.74.128/26; the last address is 14.24.74.191/26.

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:

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

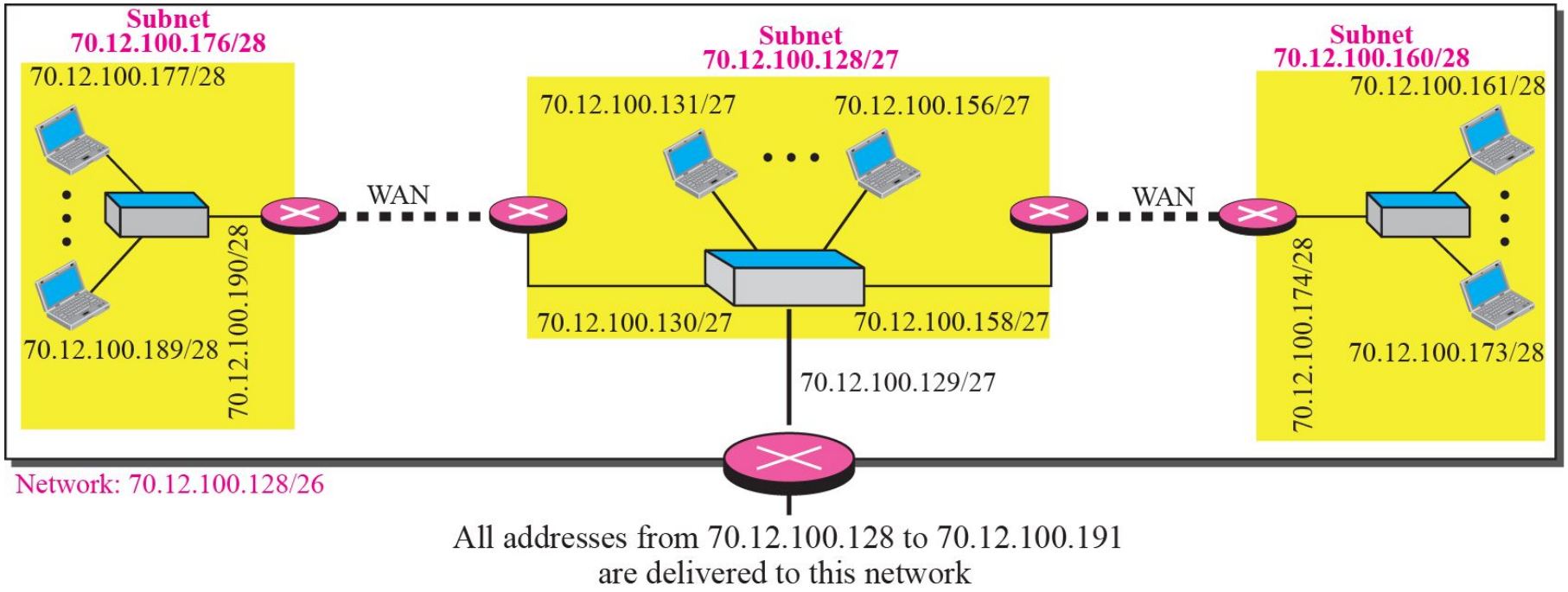
2. We can find the prefix length for each subnetwork:

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

Example 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 *Solution to 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*

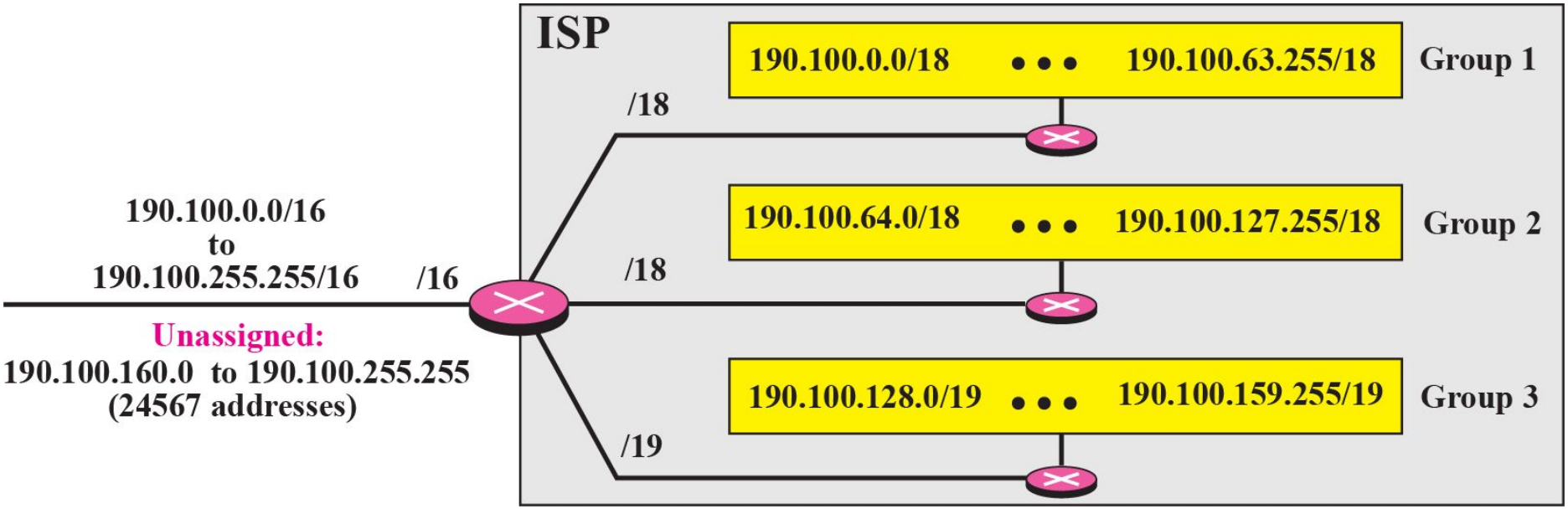
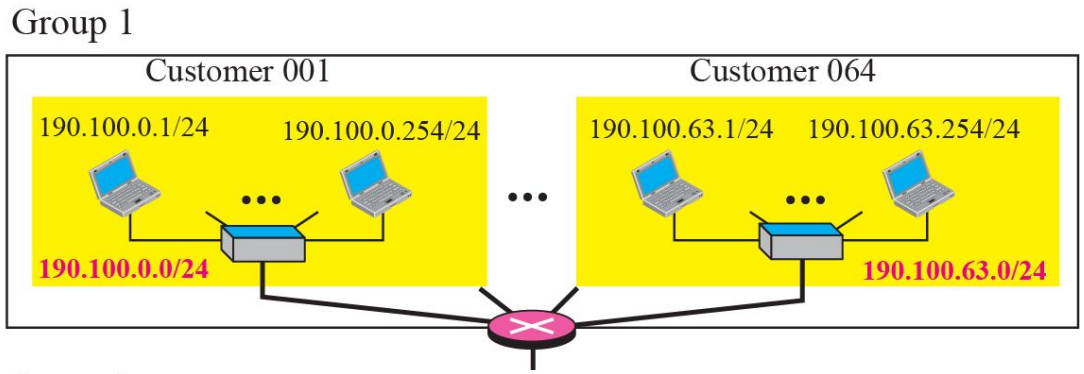


Figure 5.34 *Solution to Example 5.35: second step*

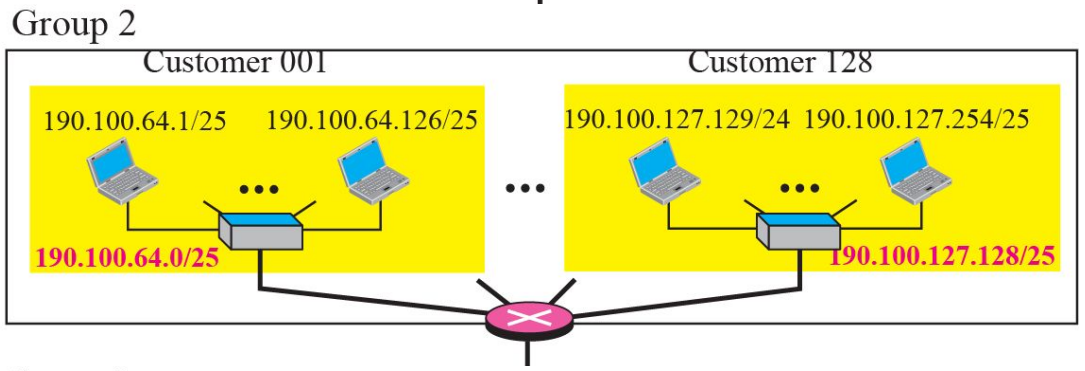
Group: $n = 18$

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



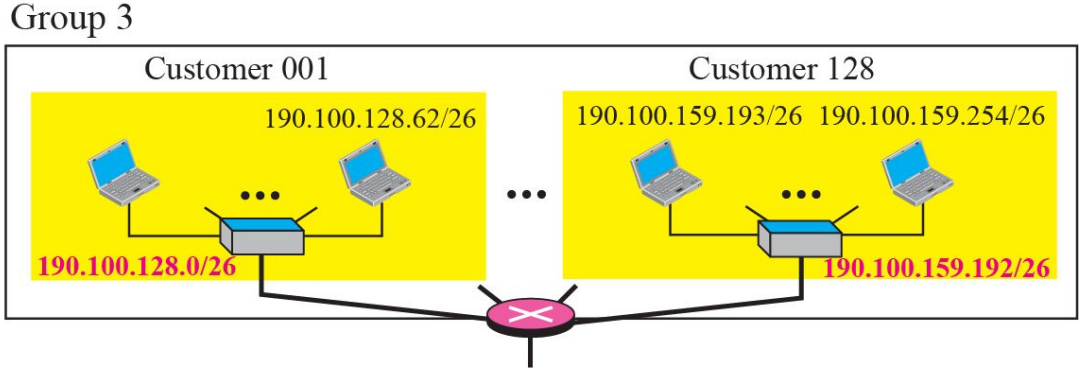
Group: $n = 18$

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



Group: $n = 19$

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



NAT: Network Address Translation

- 16-bit port-number field:
  - 60,000 simultaneous connections with a single LAN-side address!
- NAT is controversial:
  - violates end-to-end argument
    - Internal computers not visible to outside
    - Outside hosts have trouble to request service from local computers, e.g., P2P, video conference, web hosting.
  - address shortage should instead be solved by IPv6

Figure 19.10 *A NAT implementation*

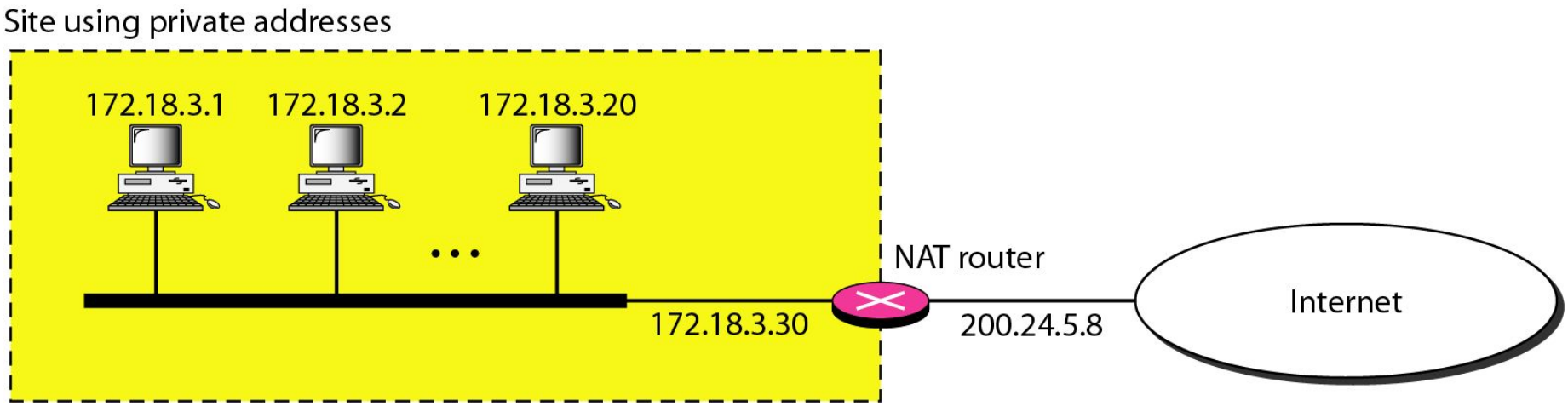


Figure 19.11 *Addresses in a NAT*

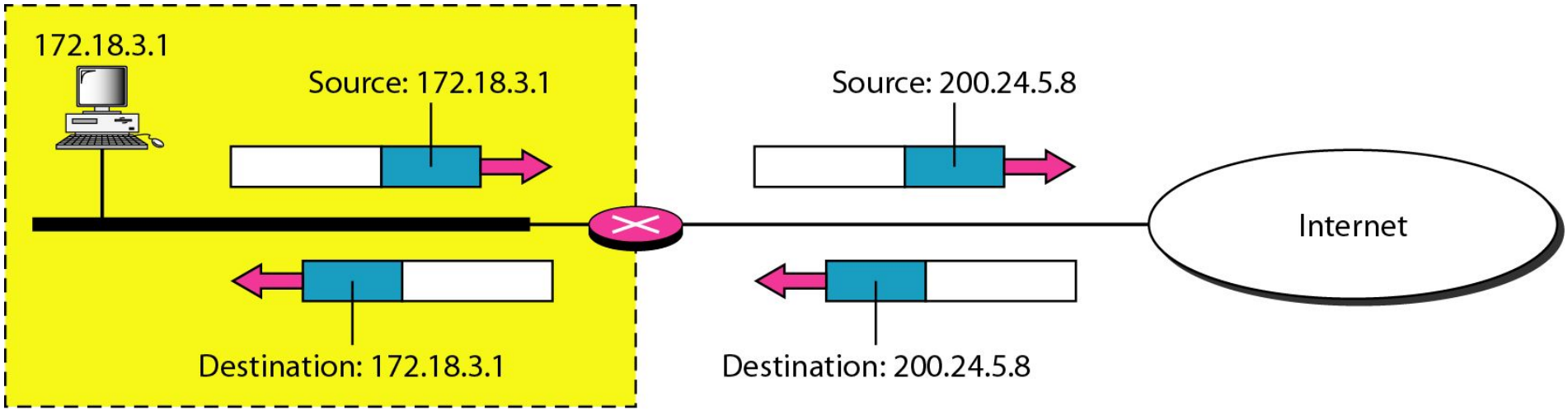


Figure 5.40 *Address resolution*

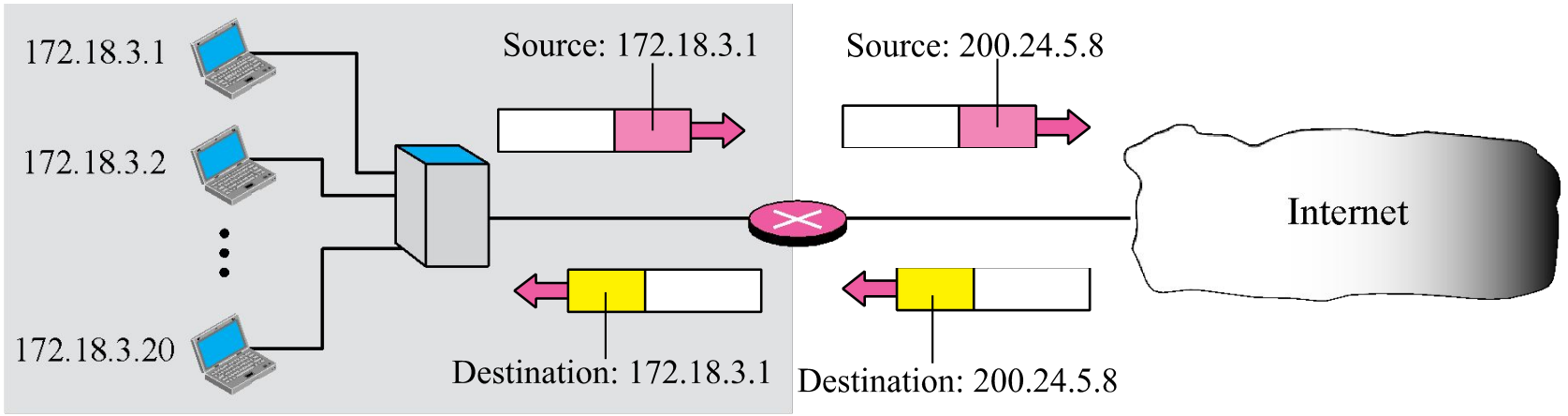


Table 19.4 ***Five-column translation table***

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

20-2 IPv4

The Internet Protocol version 4 (IPv4) is the delivery mechanism used by the TCP/IP protocols.

Figure 20.4 *Position of IPv4 in TCP/IP protocol suite*

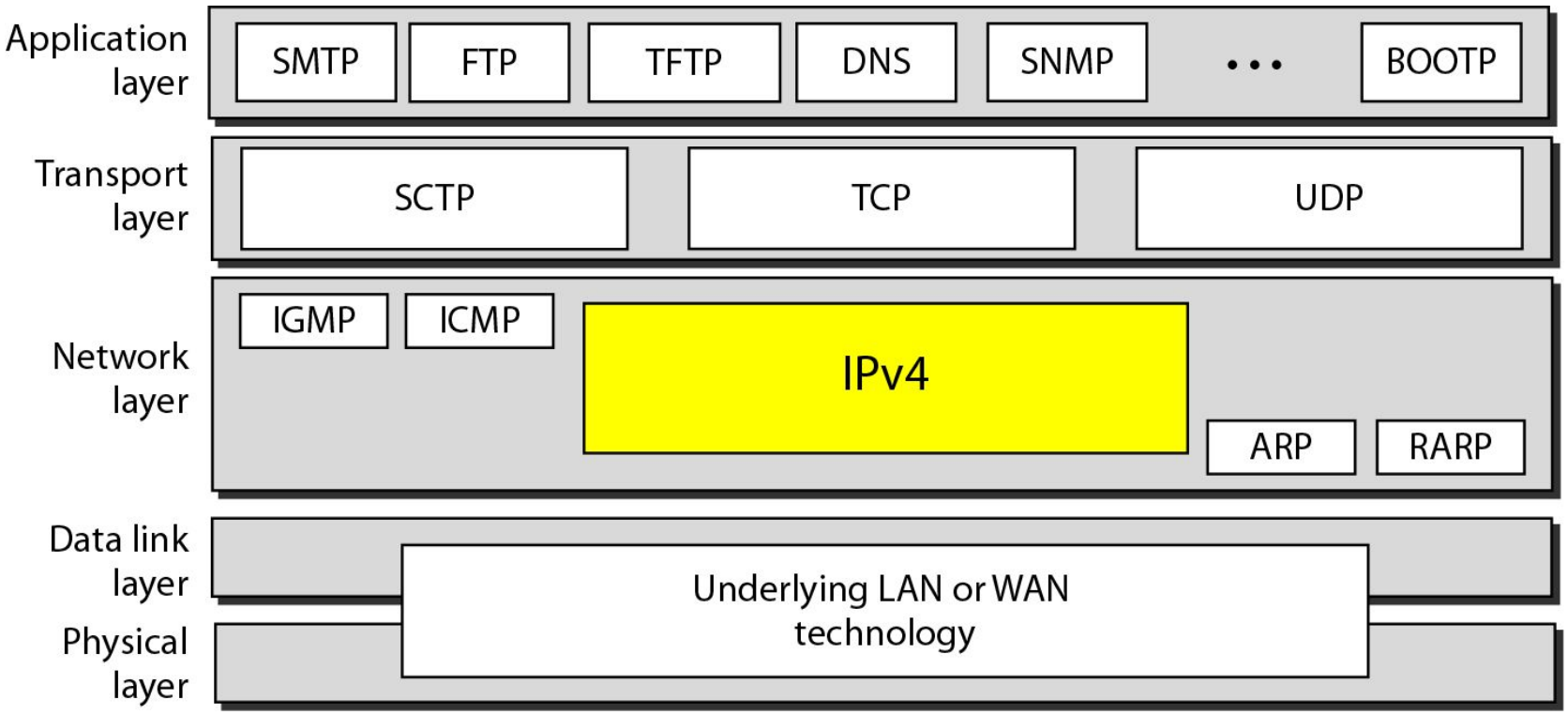
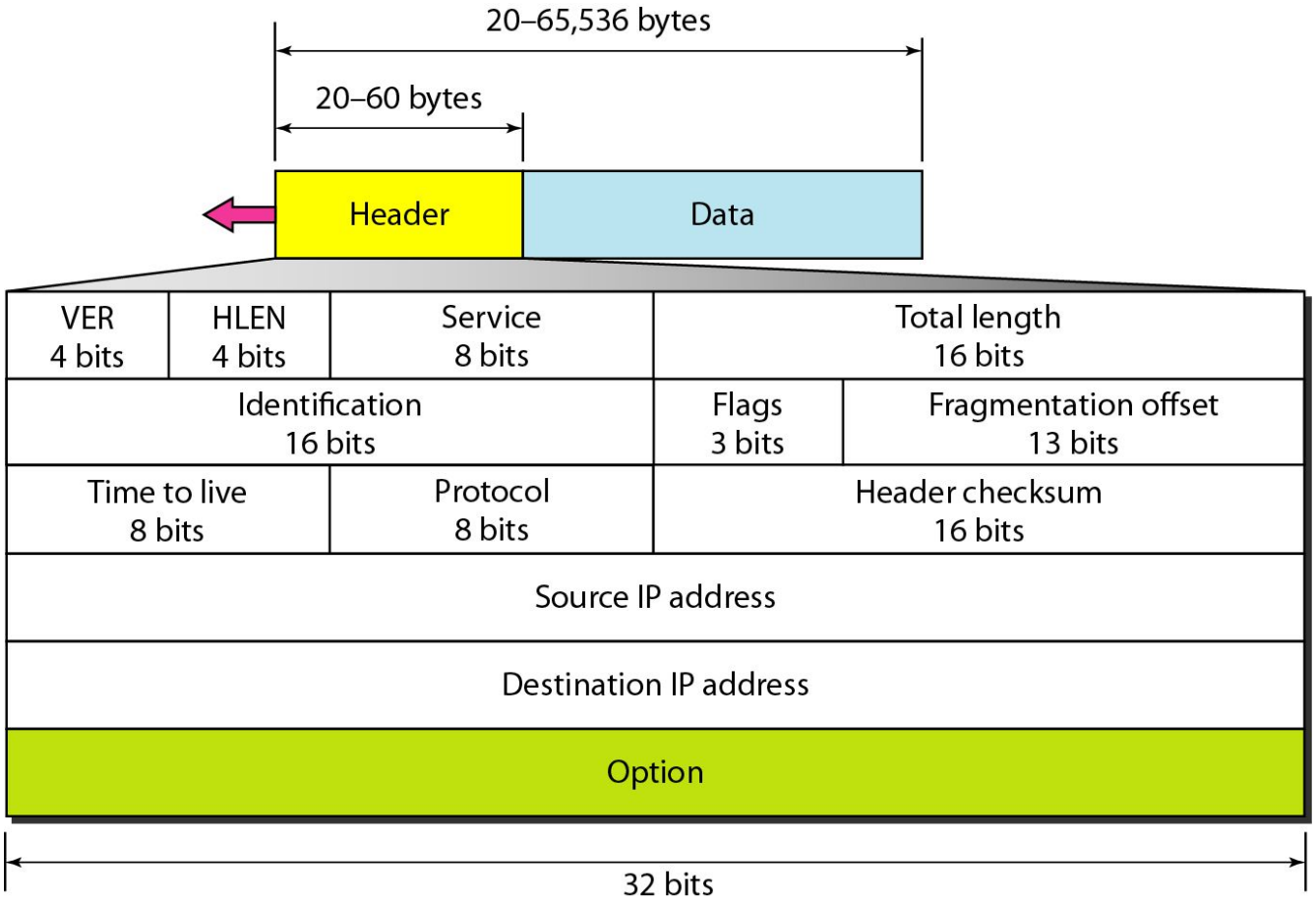


Figure 20.5 *IPv4 datagram format*



Version (VER): This 4-bit field defines the version of the IPv4 protocol. Currently the version is 4. However, version 6 may totally replace version 4 in the future.

Header length (HLEN): This 4-bit field defines the total length of the datagram header . This field is needed because the length of the header is variable (between 20 and 60 bytes). When there are no options, the header length is 20 bytes, and the value of this field is 5 (5 x 4 = 20). When the option field is at its maximum size, the value of this field is 15 (15 x 4 = 60).

Services: IETF has changed the interpretation and name of this 8-bit field. This field, previously called service type, is now called differentiated services.

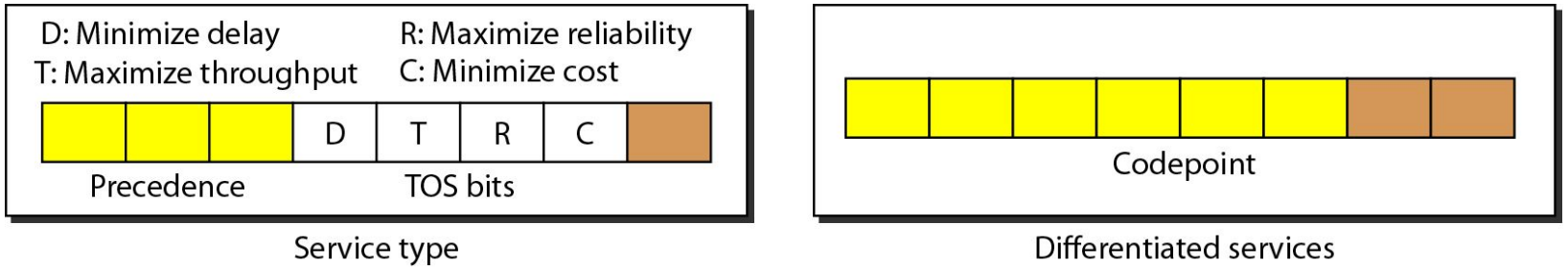
Service Type

In this interpretation, the first 3 bits are called precedence bits. The next 4 bits are called type of service (TOS) bits, and the last bit is not used.

a. Precedence is a 3-bit subfield ranging from 0 (000 in binary) to 7 (111 in binary). The precedence defines the priority of the datagram in issues such as congestion. If a router is congested and needs to discard some datagrams, those datagrams with lowest precedence are discarded first.

b. TOS bits is a 4-bit subfield with each bit having a special meaning. Although a bit can be either 0 or 1, one and only one of the bits can have the value of 1 in each datagram. The bit patterns and their interpretations are given in Table . With only 1 bit set at a time, we can have five different types of services.

Figure 20.6 *Service type or differentiated services*



Note

The precedence subfield was part of version 4, but never used.

Table 20.1 *Types of service*

TOS Bits	Description
0000	Normal (Default)
0001	Minimize cost
0010	Maximize reliability
0100	Maximize throughput
1000	Minimize delay

Table 20.2 *Default types of service*

Protocol	*TOS Bits*	*Description*
ICMP	0000	Normal
BOOTP	0000	Normal
NNTP	0001	Minimize cost
IGP	0010	Maximize reliability
SNMP	0010	Maximize reliability
TELNET	1000	Minimize delay
FTP (data)	0100	Maximize throughput
FTP (control)	1000	Minimize delay
TFTP	1000	Minimize delay
SMTP (command)	1000	Minimize delay
SMTP (data)	0100	Maximize throughput
DNS (UDP query)	1000	Minimize delay
DNS (TCP query)	0000	Normal
DNS (zone)	0100	Maximize throughput

Table 20.3 ***Values (in decimal) for code points***

Value	*Protocol*
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

Differentiated Services:

In this interpretation, the first 6 bits make up the codepoint subfield, and the last 2 bits are not used. The codepoint subfield can be used in two different ways.

a. When the 3 rightmost bits are Os, the 3 leftmost bits are interpreted the same as the precedence bits in the service type interpretation. In other words, it is compatible with the old interpretation.

b. When the 3 rightmost bits are not all Os, the 6 bits define 64 services based on the priority assignment by the Internet or local authorities.

Total length: This is a 16-bit field that defines the total length (header plus data) of the IPv4 datagram in bytes.

Length of data =total length - header length

Identification: This field is used in fragmentation (discussed in the next section).

Flags: This field is used in fragmentation (discussed in the next section).

Fragmentation offset: This field is used in fragmentation.

Time to live:A datagram has a limited lifetime in its travel through an internet. This field was originally designed to hold a timestamp, which was decremented by each visited router. The datagram was discarded when the value became zero. When a source host sends the datagram, it stores a number in this field. This value is approximately 2 times the maximum number of routes between any two hosts. Each router that processes the datagram decrements this number by 1. If this value, after being decremented, is zero, the router discards the datagram.

Protocol: This 8-bit field defines the higher-level protocol that uses the services of the IPv4 layer. An IPv4 datagram can encapsulate data from several higher-level protocols such as TCP, UDP, ICMP, and IGMP.

Identification: This 16-bit field identifies a datagram originating from the source host. The combination of the identification and source IPv4 address must uniquely define a datagram as it leaves the source host. When a datagram is fragmented, the value in the identification field is copied to all fragments. In other words, all fragments have the same identification number, the same as the original datagram. The identification number helps the destination in reassembling the datagram. It knows that all fragments having the same identification value must be assembled into one datagram.

Flags. This is a 3-bit field. The first bit is reserved. The second bit is called the *do not fragment bit. If its value is 1, the machine must not fragment the datagram.* If its value is 0, the datagram can be fragmented if necessary. The third bit is called the *more fragment bit. If its value is 1, it means the datagram is not the last fragment;* there are more fragments after this one. If its value is 0, it means this is the last or only fragment.

Note

The total length field defines the total length of the datagram including the header.

Figure 20.7 *Encapsulation of a small datagram in an Ethernet frame*

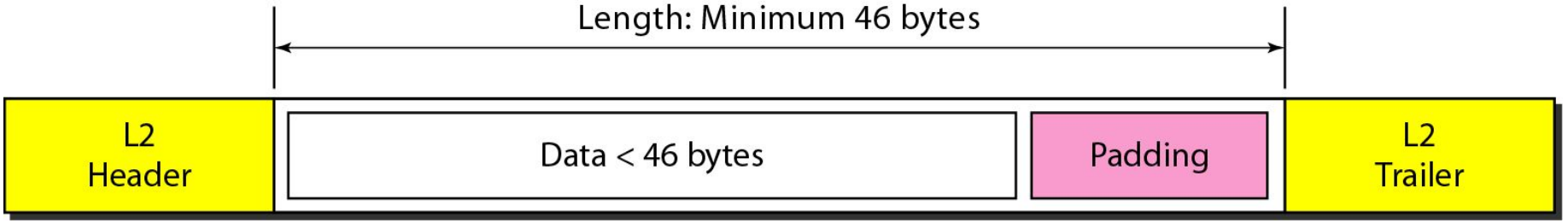


Figure 20.8 *Protocol field and encapsulated data*

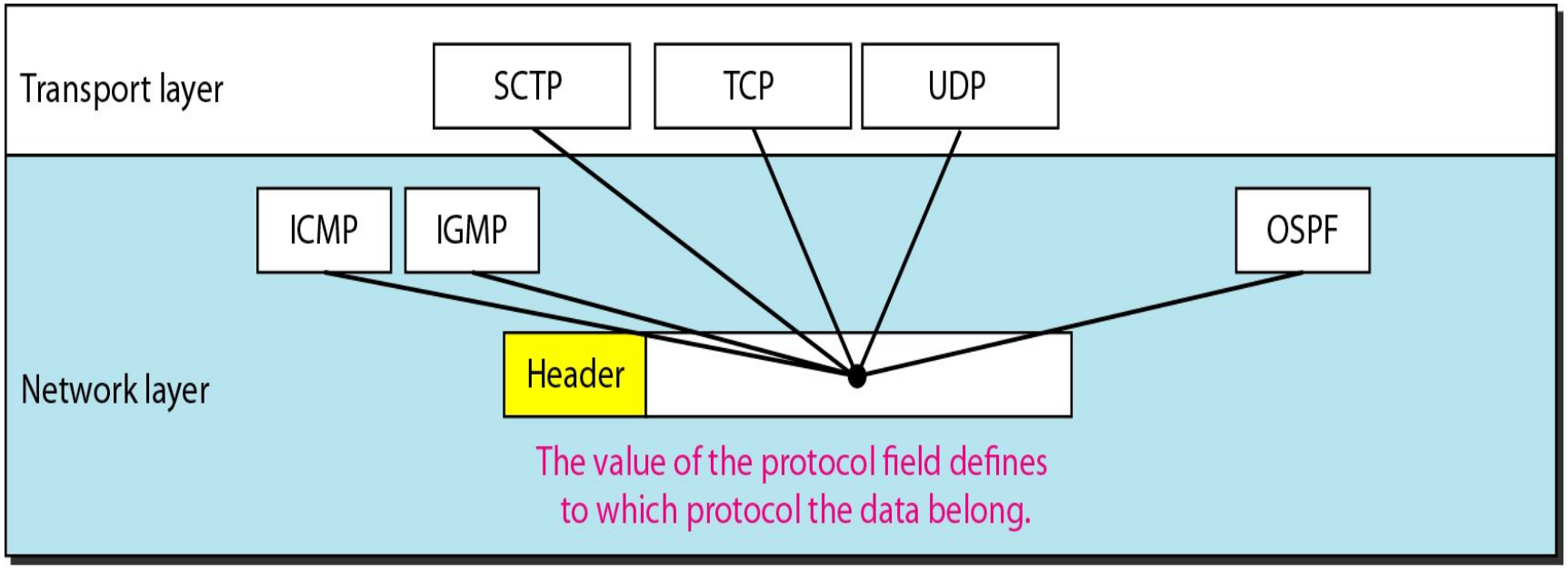


Table 20.4 ***Protocol values***

Value	*Protocol*
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

Example 20.1

An IPv4 packet has arrived with the first 8 bits as shown:

01000010

The receiver discards the packet. Why?

Solution

There is an error in this packet. The 4 leftmost bits (0100) show the version, which is correct. The next 4 bits (0010) show an invalid header length ($2 \times 4 = 8$). The minimum number of bytes in the header must be 20. The packet has been corrupted in transmission.

Example 20.2

In an IPv4 packet, the value of HLEN is 1000 in binary. How many bytes of options are being carried by this packet?

Solution

The HLEN value is 8, which means the total number of bytes in the header is 8 × 4, or 32 bytes. The first 20 bytes are the base header, the next 12 bytes are the options.

Example 20.3

In an IPv4 packet, the value of HLEN is 5, and the value of the total length field is 0x0028. How many bytes of data are being carried by this packet?

Solution

The HLEN value is 5, which means the total number of bytes in the header is 5 × 4, or 20 bytes (no options). The total length is 40 bytes, which means the packet is carrying 20 bytes of data (40 − 20).

Example 20.4

An IPv4 packet has arrived with the first few hexadecimal digits as shown.

0x45000028000100000102 . . .

How many hops can this packet travel before being dropped? The data belong to what upper-layer protocol?

Solution

To find the time-to-live field, we skip 8 bytes. The time-to-live field is the ninth byte, which is 01. This means the packet can travel only one hop. The protocol field is the next byte (02), which means that the upper-layer protocol is IGMP.

Figure 20.9 *Maximum transfer unit (MTU)*

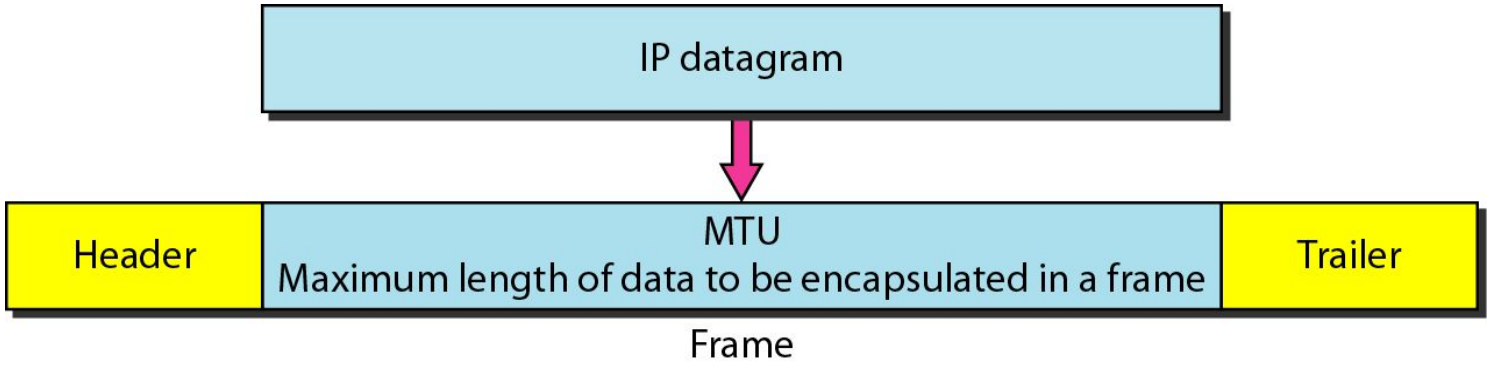


Table 20.5 *MTUs for some networks*

Protocol	*MTU*
Hyperchannel	65,535
Token Ring (16 Mbps)	17,914
Token Ring (4 Mbps)	4,464
FDDI	4,352
Ethernet	1,500
X.25	576
PPP	296

Figure 20.10 *Flags used in fragmentation*

Figure 20.11 *Fragmentation example*

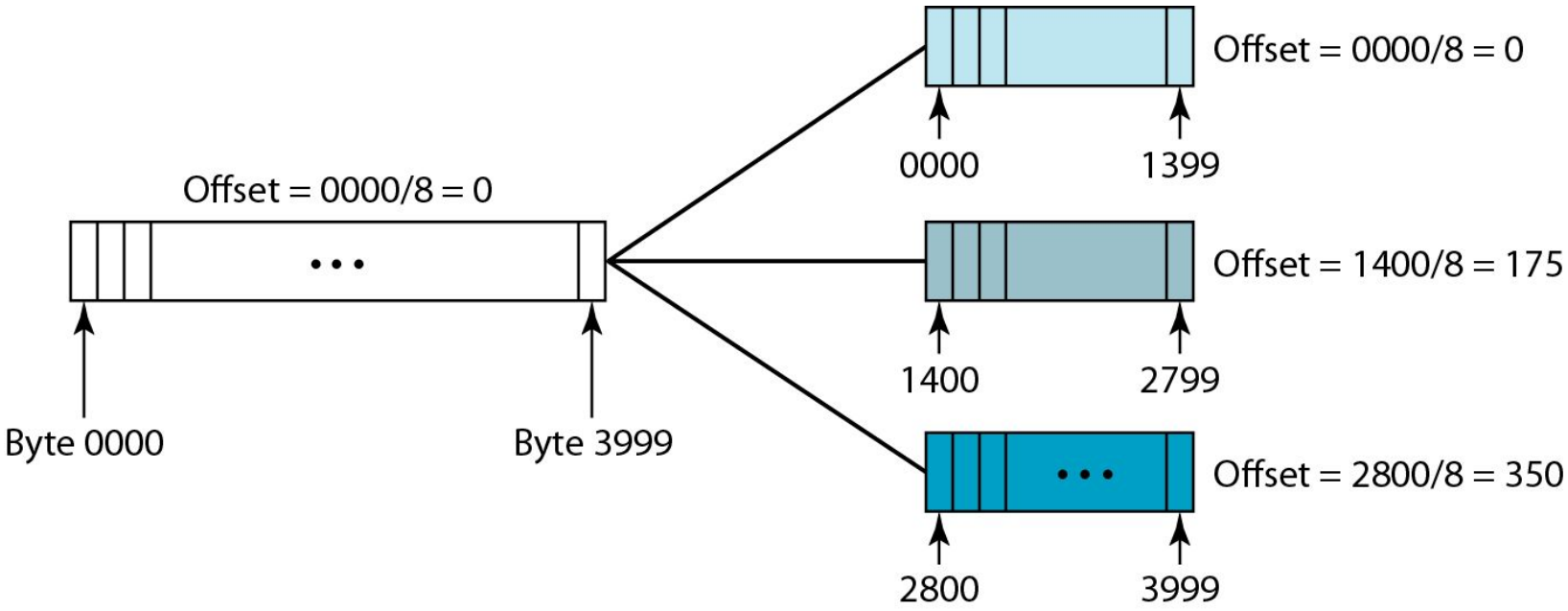
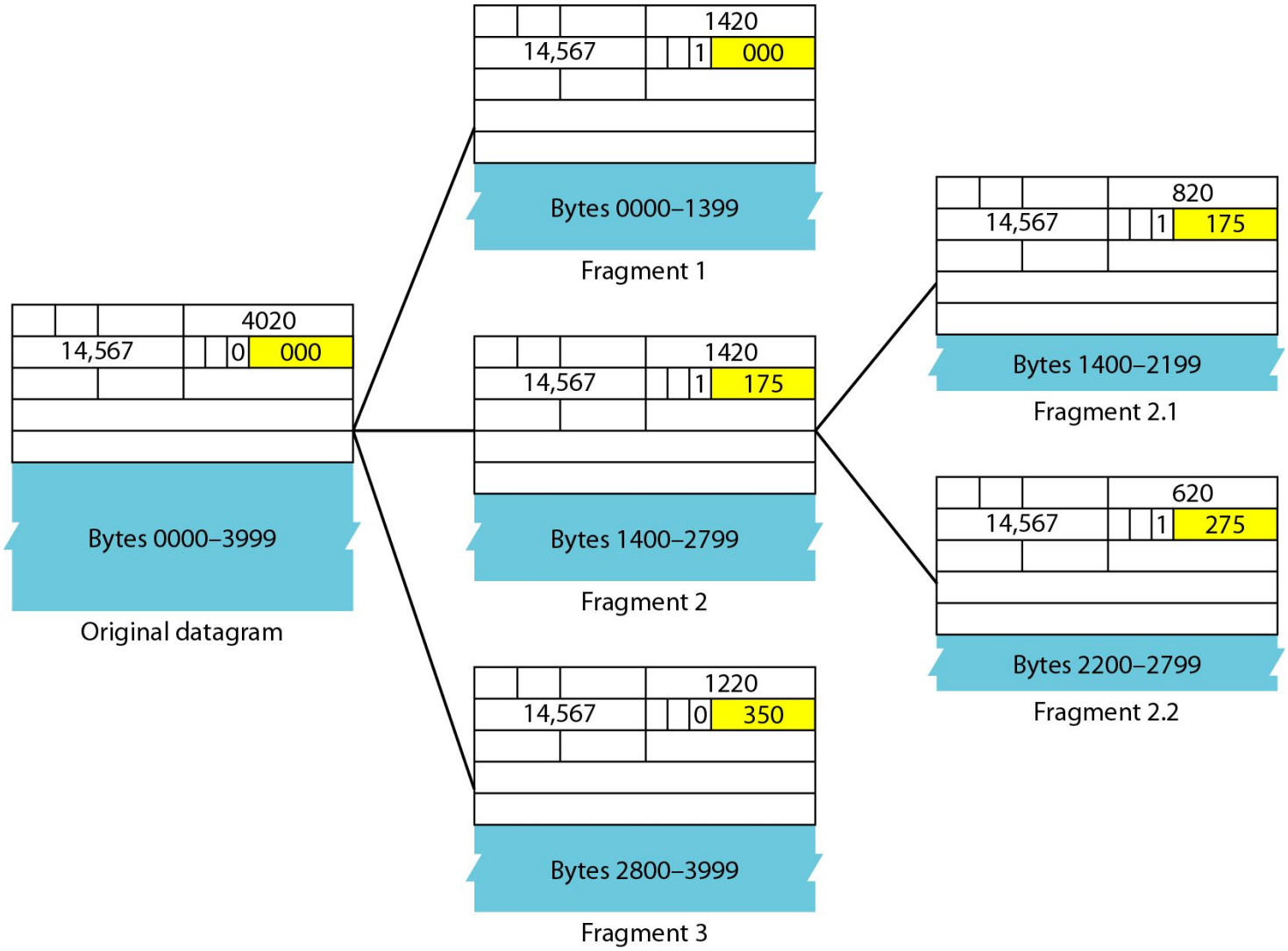


Figure 20.12 *Detailed fragmentation example*



Example 20.5

A packet has arrived with an M bit value of 0. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?

Solution

If the M bit is 0, it means that there are no more fragments; the fragment is the last one. However, we cannot say if the original packet was fragmented or not. A non-fragmented packet is considered the last fragment.

Example 20.6

A packet has arrived with an M bit value of 1. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?

Solution

If the M bit is 1, it means that there is at least one more fragment. This fragment can be the first one or a middle one, but not the last one. We don't know if it is the first one or a middle one; we need more information (the value of the fragmentation offset).

Example 20.7

A packet has arrived with an M bit value of 1 and a fragmentation offset value of 0. Is this the first fragment, the last fragment, or a middle fragment?

Solution

Because the M bit is 1, it is either the first fragment or a middle one. Because the offset value is 0, it is the first fragment.

Example 20.8

A packet has arrived in which the offset value is 100. What is the number of the first byte? Do we know the number of the last byte?

Solution

To find the number of the first byte, we multiply the offset value by 8. This means that the first byte number is 800. We cannot determine the number of the last byte unless we know the length.

Example 20.9

A packet has arrived in which the offset value is 100, the value of HLEN is 5, and the value of the total length field is 100. What are the numbers of the first byte and the last byte?

Solution

The first byte number is 100 × 8 = 800. The total length is 100 bytes, and the header length is 20 bytes (5 × 4), which means that there are 80 bytes in this datagram. If the first byte number is 800, the last byte number must be 879.

Example 20.10

Figure 20.13 shows an example of a checksum calculation for an IPv4 header without options. The header is divided into 16-bit sections. All the sections are added and the sum is complemented. The result is inserted in the checksum field.

Figure 20.13 *Example of checksum calculation in IPv4*

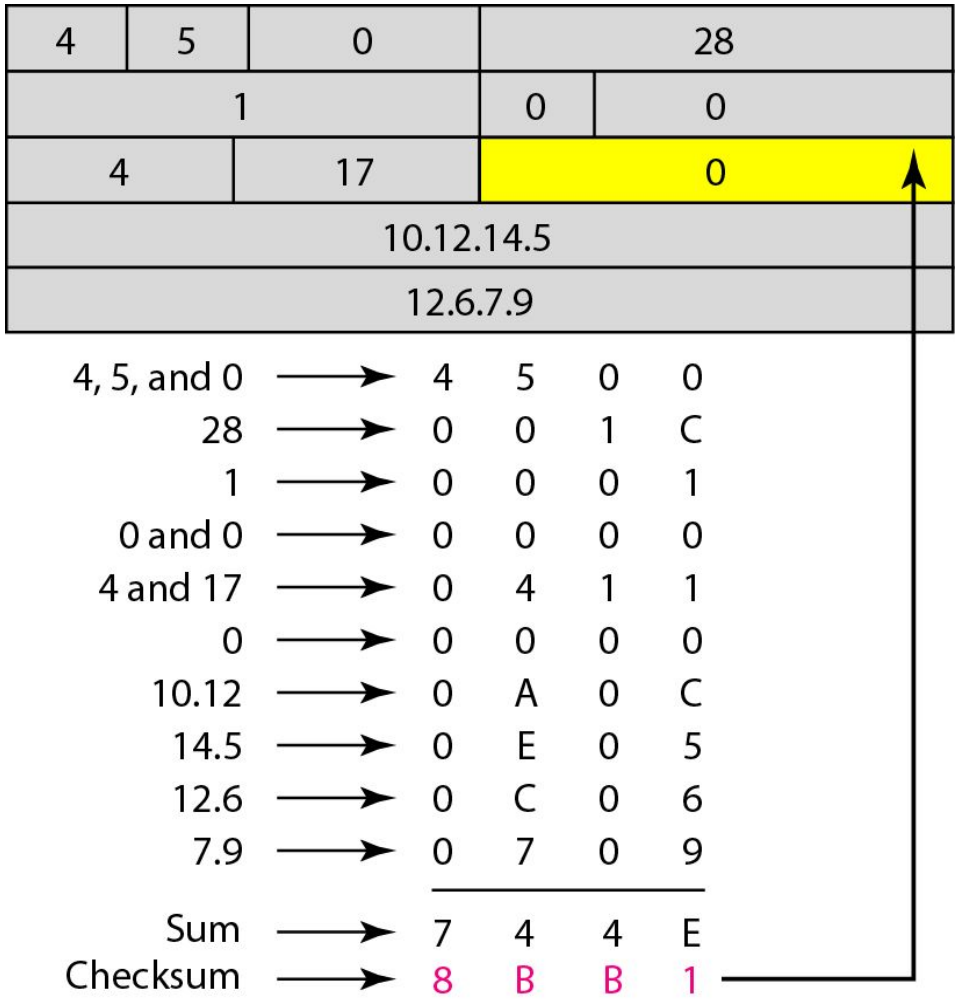
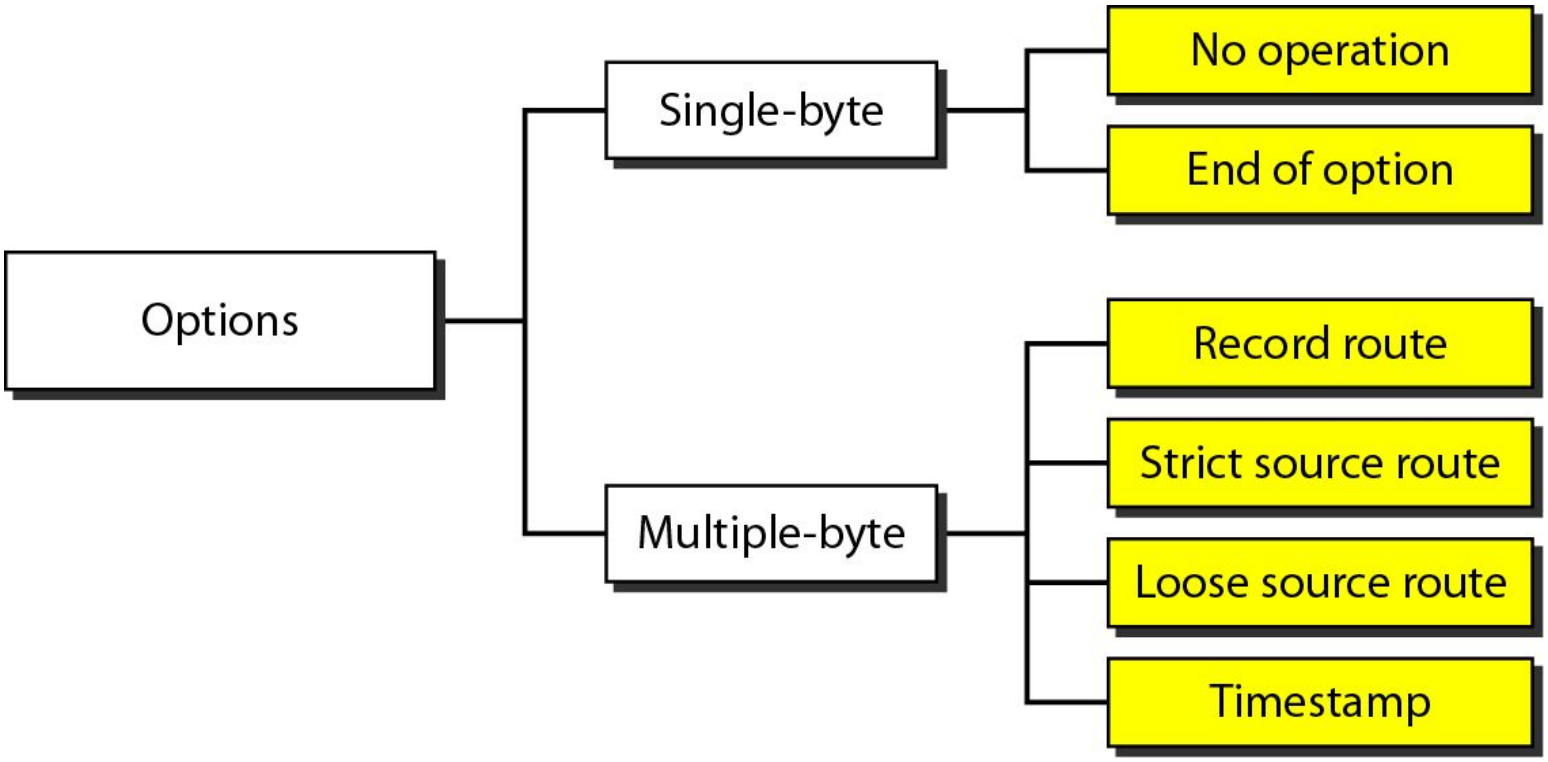


Figure 20.14 *Taxonomy of options in IPv4*



20-3 IPv6

The network layer protocol in the TCP/IP protocol suite is currently IPv4. Although IPv4 is well designed, data communication has evolved since the inception of IPv4 in the 1970s. IPv4 has some deficiencies that make it unsuitable for the fast-growing Internet.

Features of IPv6

- Larger Address Space
- Aggregation-based address hierarchy
  - Efficient backbone routing
- Efficient and Extensible IP datagram
- Stateless Address Autoconfiguration
- Security (IPsec mandatory)
- Mobility

Why IPv6?

- Deficiency of IPv4
- Address space exhaustion
- New types of service ☐ Integration
  - Multicast
  - Quality of Service
  - Security
  - Mobility (MIPv6)
- Header and format limitations

Advantages of IPv6 over IPv4

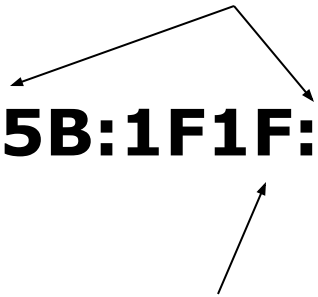
- Larger address space
- Better header format
- New options
- Allowance for extension
- Support for resource allocation
- Support for more security
- Support for mobility

128-bit IPv6 Address

3FFE:085B:1F1F:0000:0000:0000:00A9:1234

8 groups of 16-bit hexadecimal numbers separated by “:”

Leading zeros can be removed



3FFE:85B:1F1F::A9:1234

:: = all zeros in one or more group of 16-bit hexadecimal numbers

Figure 20.15 *IPv6 datagram header and payload*

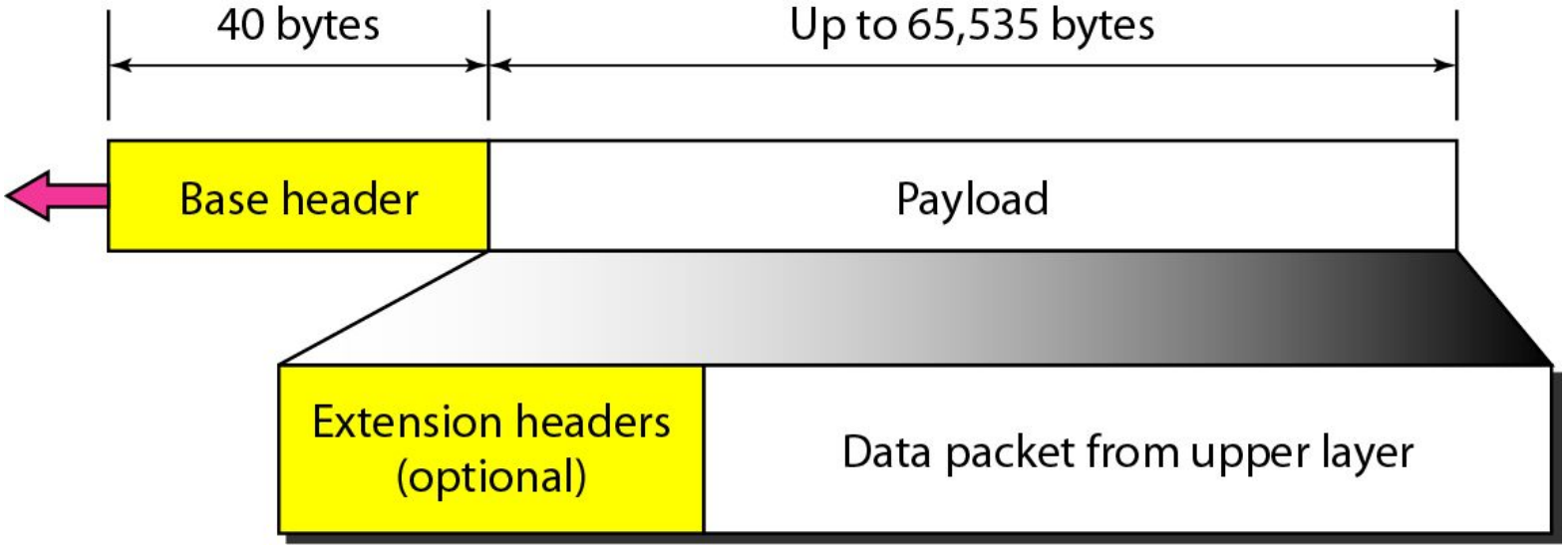
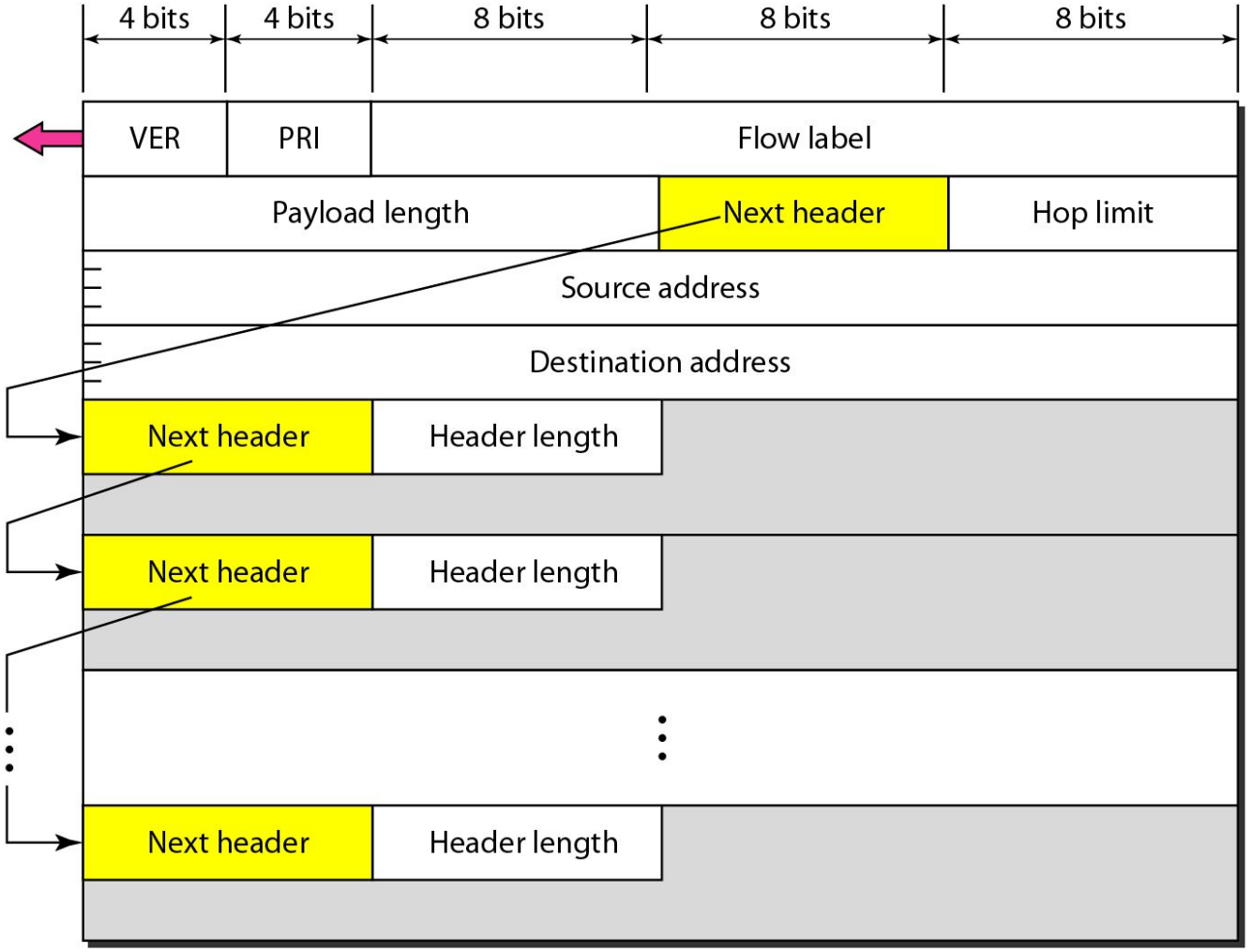
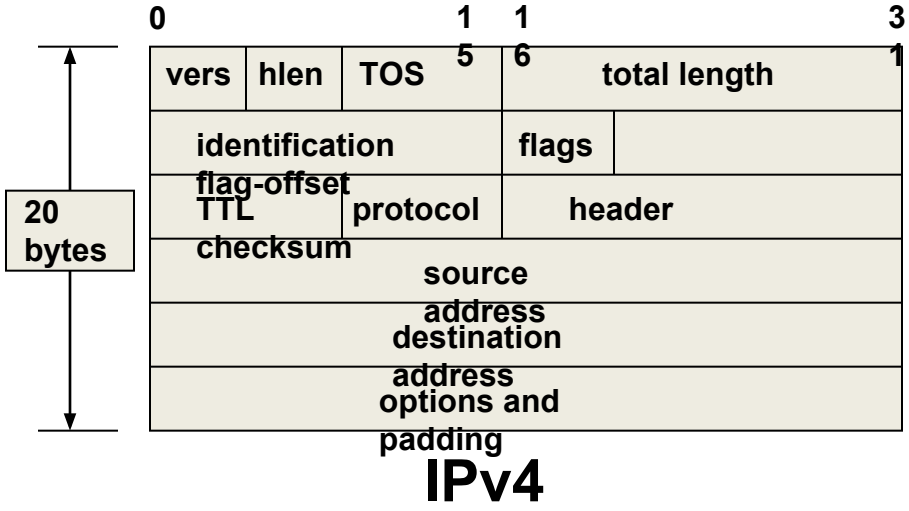


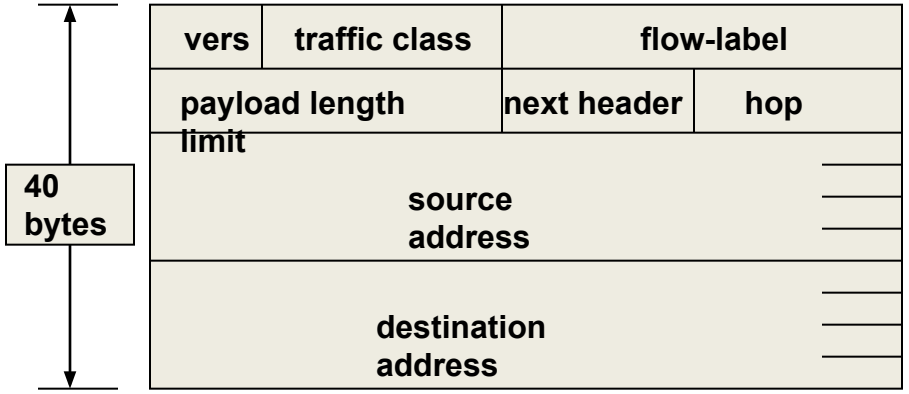
Figure 20.16 *Format of an IPv6 datagram*



Header comparison



IPv4



IPv6

Removed (6)

- ID, flags, flag offset
- TOS, hlen
- header checksum

Changed (3)

- total length => payload
- protocol => next header
- TTL => hop limit

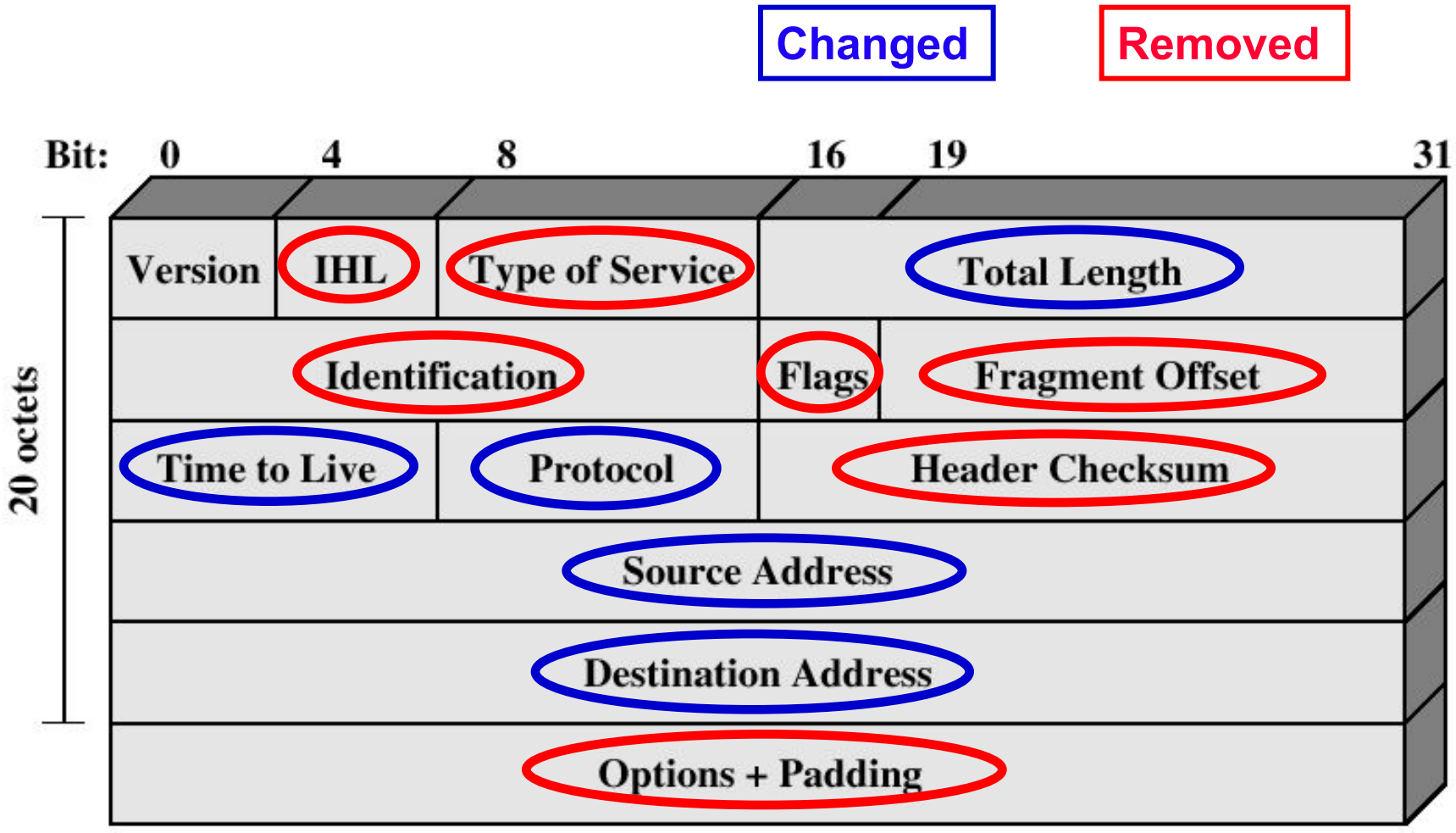
Added (2)

- traffic class
- flow label

Expanded

- address 32 to 128 bits

Header: from IPv4 to IPv6



Major Improvements of IPv6 Header

- No option field: Replaced by extension header. Result in a fixed length, 40-byte IP header.
- No header checksum: Result in fast processing.
- No fragmentation at intermediate nodes: Result in fast IP forwarding.

Traffic Class

- The 8-bit field in the IPv6 header is available for use by originating nodes and/or forwarding routers to identify and distinguish between different classes or priorities of IPv6 packets.
  - E.g., used as the codepoint in DiffServ
- General requirements
  - Service interface must provide means for upper-layer protocol to supply the value of traffic class
  - Value of traffic class can be changed by source, forwarder, receiver
  - An upper-layer protocol should not assume the value of traffic class in a packet has not been changed.

IPv6 Flow Label

- Related sequence of packets
- Needing special handling
- Identified by src & dest addr + flow label
- Router treats flow as sharing attributes
  - E.g. path, resource allocation, discard requirements, accounting, security
- May treat flows differently
  - Buffer sizes, different forwarding precedence, different quality of service
- Alternative to including all info. in every header

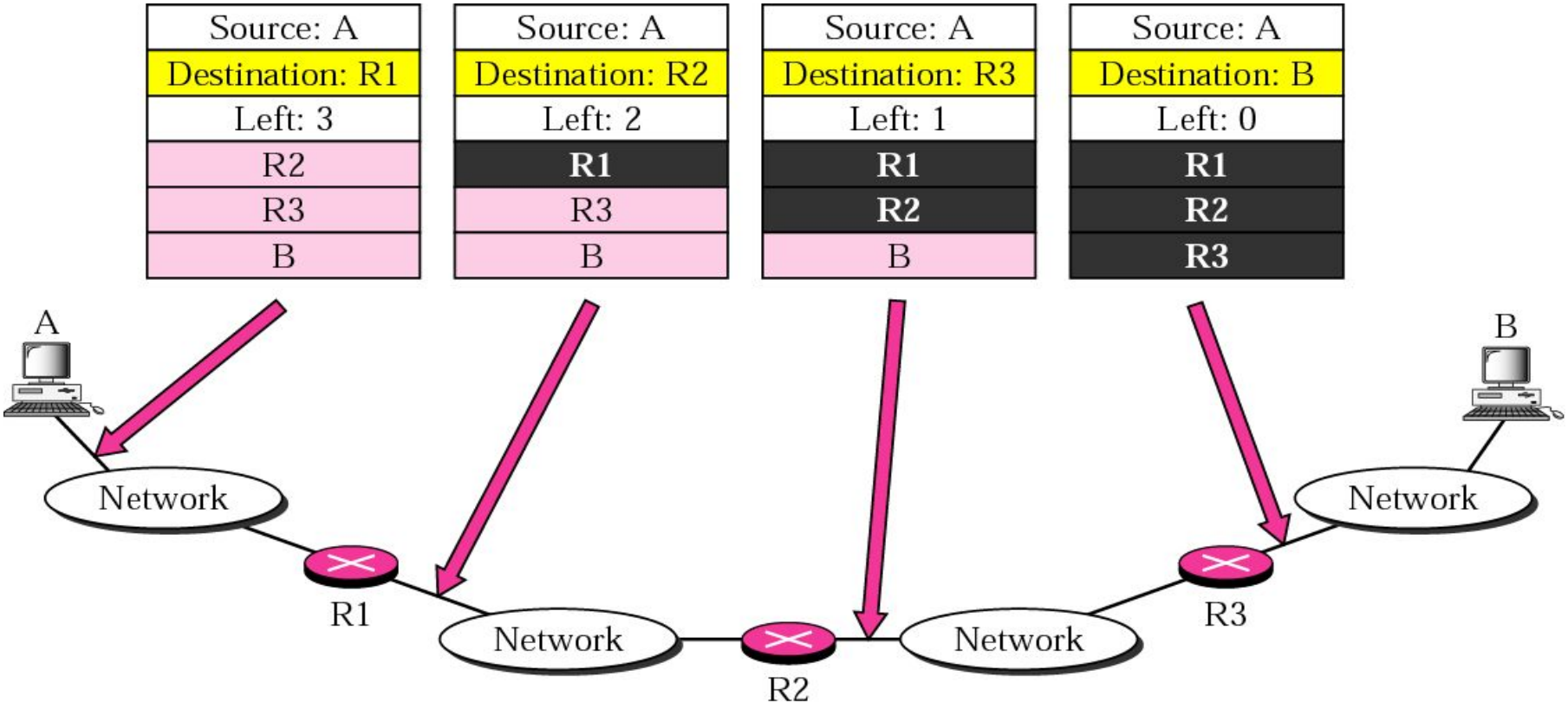
Payload Length

- 16-bit unsigned integer. Length of the IPv6 payload, i.e., the rest of the packet following this IPv6 header, in octets.
- Note that any extension headers present are considered part of the payload, i.e., included in the length count.

Routing Header

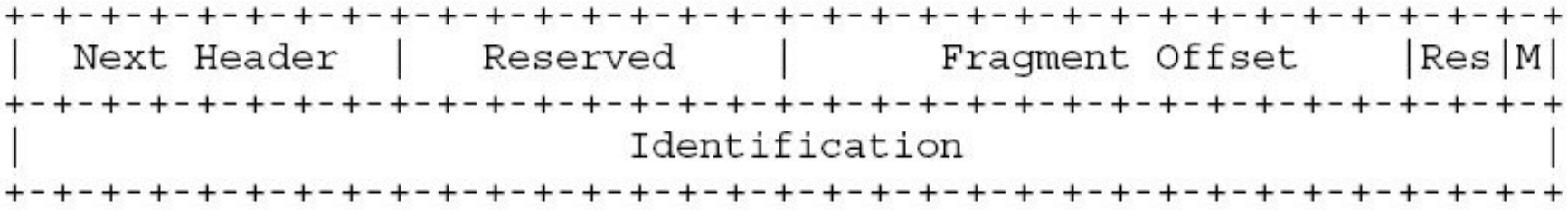
- List of one or more intermediate nodes to visit
- Header includes
  - Next Header
  - Header extension length
  - Routing type (e.g. type 0 = Source Routing)
  - Segments left

Source Routing Example



Fragment Header (1)

- Fragment Offset: 8-bit unsigned integer
  - The offset, in 8-octet units, of the data following this header, relative to the start of the Fragmentable Part of the original packet
  - Unfragmentable part: IPv6 header + any extension headers that must be processed by nodes en route



Fragment Header (2)

- M flag: 1=more fragments, 0=last fragment
- Identification: combined with the src & dest addr uniquely identifies the original packet

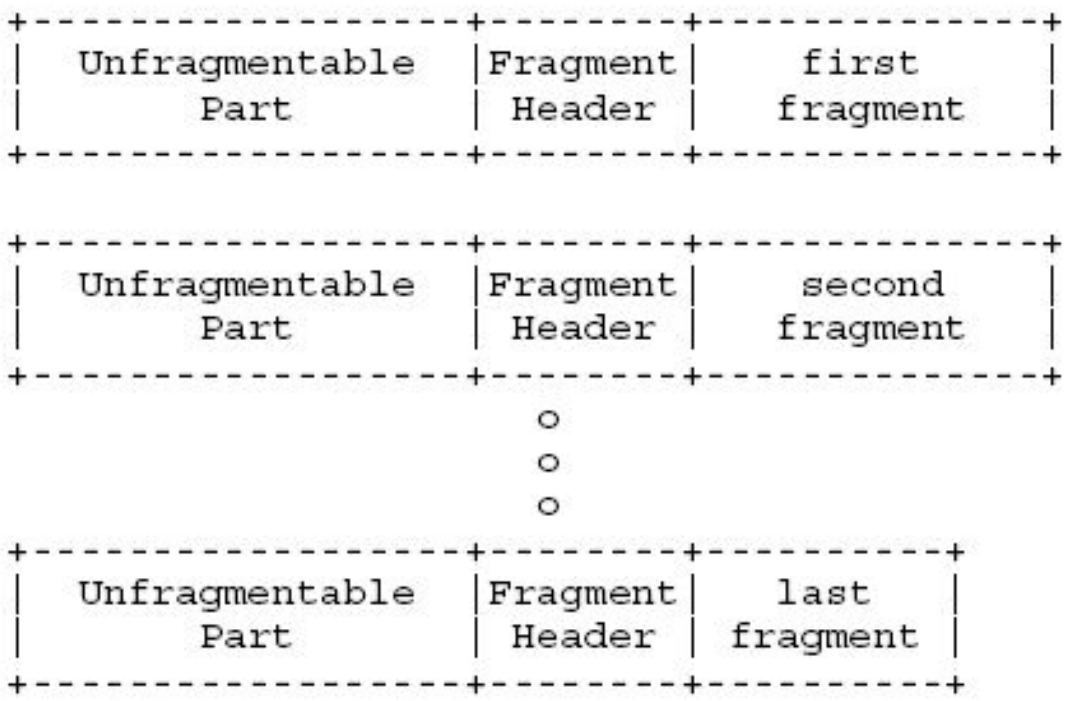


Table 20.6 *Next header codes for IPv6*

Code	*Next Header*
0	Hop-by-hop option
2	ICMP
6	TCP
17	UDP
43	Source routing
44	Fragmentation
50	Encrypted security payload
51	Authentication
59	Null (no next header)
60	Destination option

Table 20.7 ***Priorities for congestion-controlled traffic***

Priority	*Meaning*
0	No specific traffic
1	Background data
2	Unattended data traffic
3	Reserved
4	Attended bulk data traffic
5	Reserved
6	Interactive traffic
7	Control traffic

Table 20.8 ***Priorities for noncongestion-controlled traffic***

Priority	*Meaning*
8	Data with greatest redundancy
. . .	. . .
15	Data with least redundancy

Table 20.9 *Comparison between IPv4 and IPv6 packet headers*

Comparison
1. The header length field is eliminated in IPv6 because the length of the header is fixed in this version.
2. The service type field is eliminated in IPv6. The priority and flow label fields together take over the function of the service type field.
3. The total length field is eliminated in IPv6 and replaced by the payload length field.
4. The identification, flag, and offset fields are eliminated from the base header in IPv6. They are included in the fragmentation extension header.
5. The TTL field is called hop limit in IPv6.
6. The protocol field is replaced by the next header field.
7. The header checksum is eliminated because the checksum is provided by upper-layer protocols; it is therefore not needed at this level.
8. The option fields in IPv4 are implemented as extension headers in IPv6.

Figure 20.17 *Extension header types*

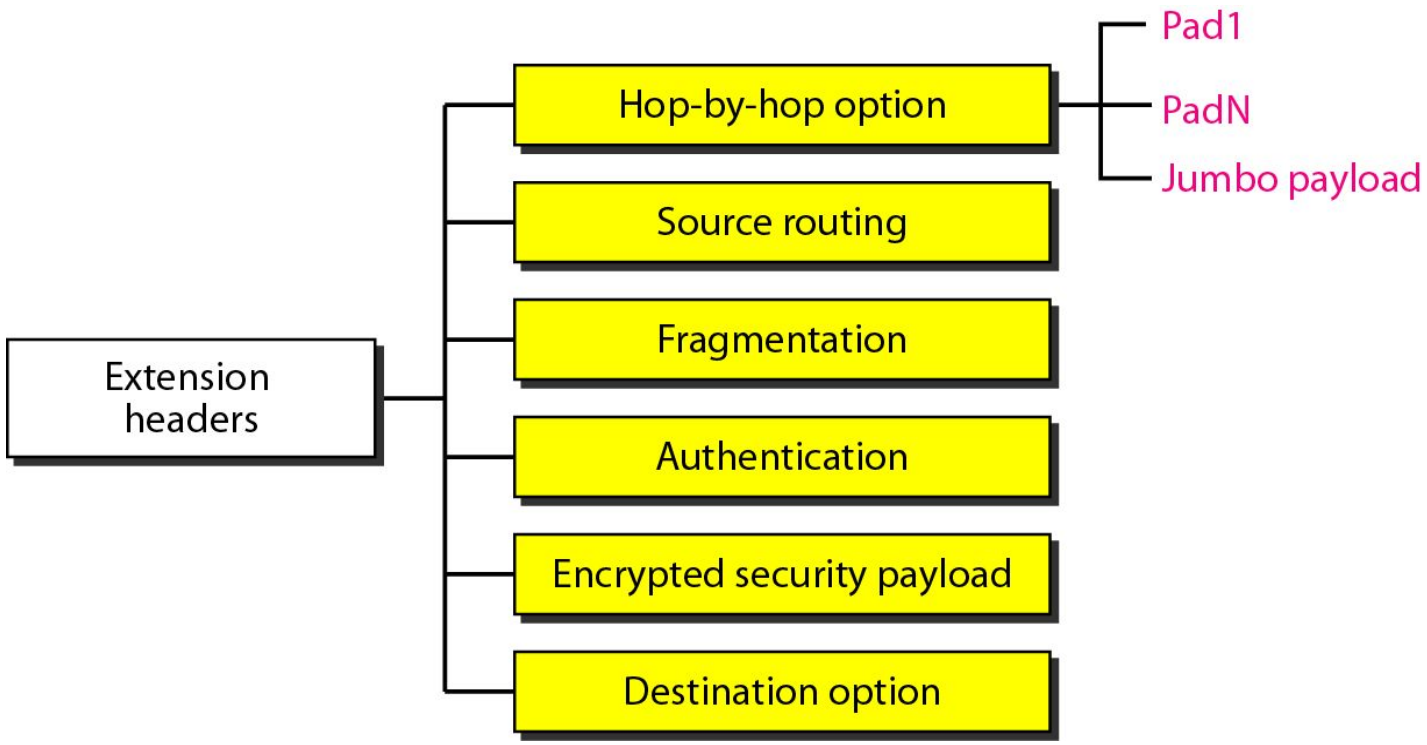


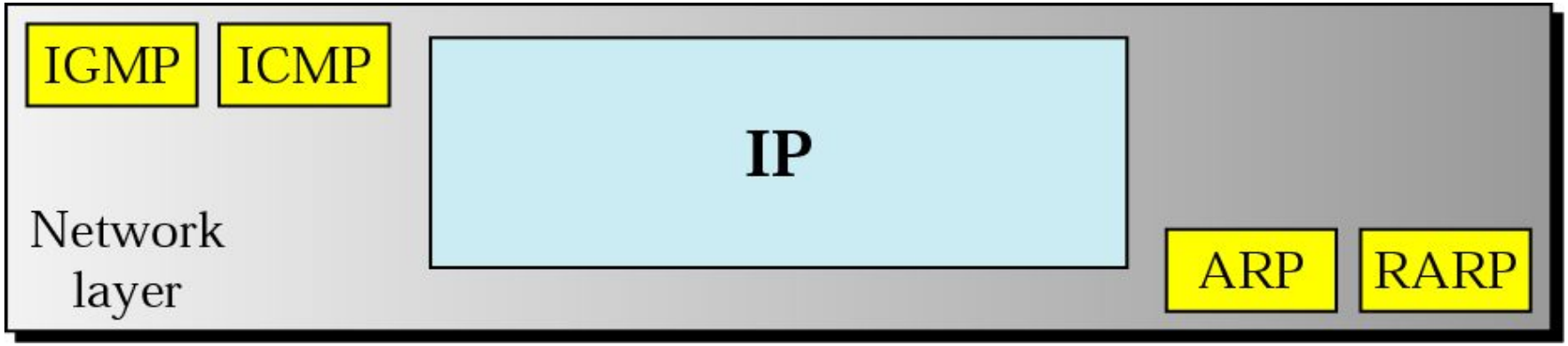
Table 20.10 *Comparison between IPv4 options and IPv6 extension headers*

Comparison
1. The no-operation and end-of-option options in IPv4 are replaced by Pad1 and PadN options in IPv6.
2. The record route option is not implemented in IPv6 because it was not used.
3. The timestamp option is not implemented because it was not used.
4. The source route option is called the source route extension header in IPv6.
5. The fragmentation fields in the base header section of IPv4 have moved to the fragmentation extension header in IPv6.
6. The authentication extension header is new in IPv6.
7. The encrypted security payload extension header is new in IPv6.

Conclusion

- IPv6 is NEW …
  - built on the experiences learned from IPv4
  - new features
  - large address space
  - new efficient header
  - autoconfiguration
- … and OLD
  - still IP
  - build on a solid base
  - started in 1995, a lot of implementations and tests done

Figure 20.1 Protocols at network layer



20.1 ARP

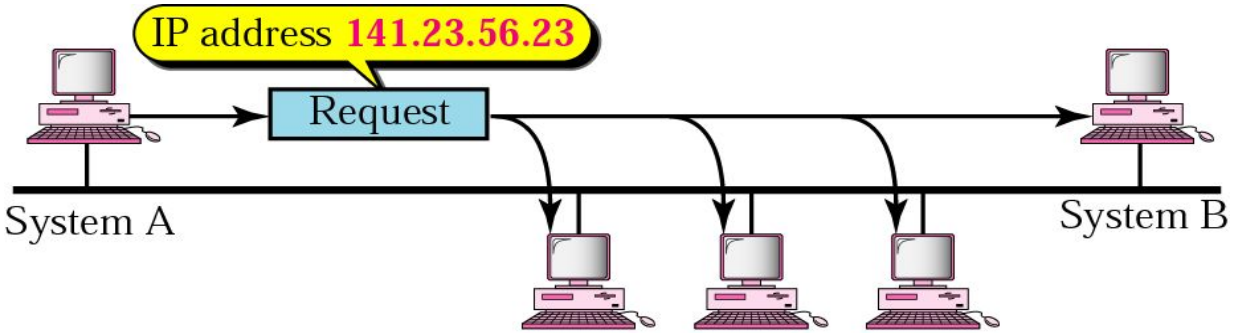
Mapping

Packet Format

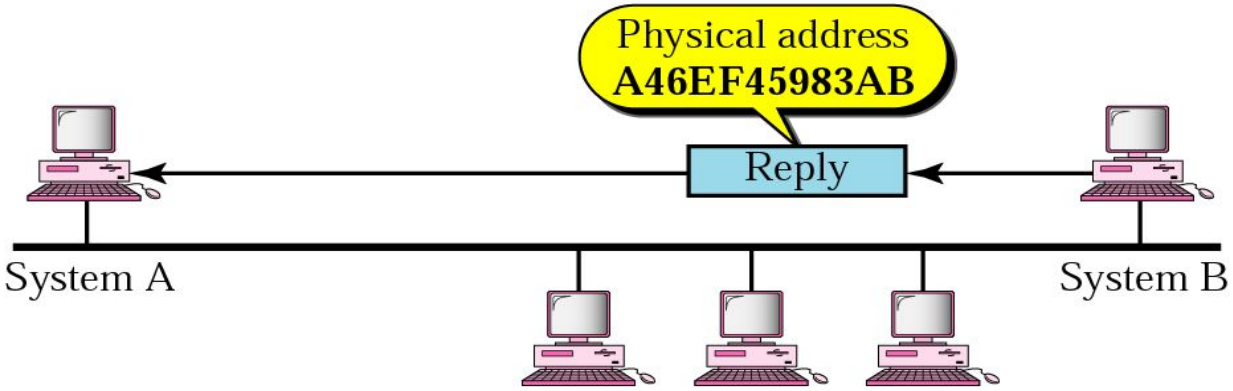
Encapsulation

Operation

Figure 20.2 ARP operation



a. ARP request is broadcast



b. ARP reply is unicast

Figure 20.3 ARP packet

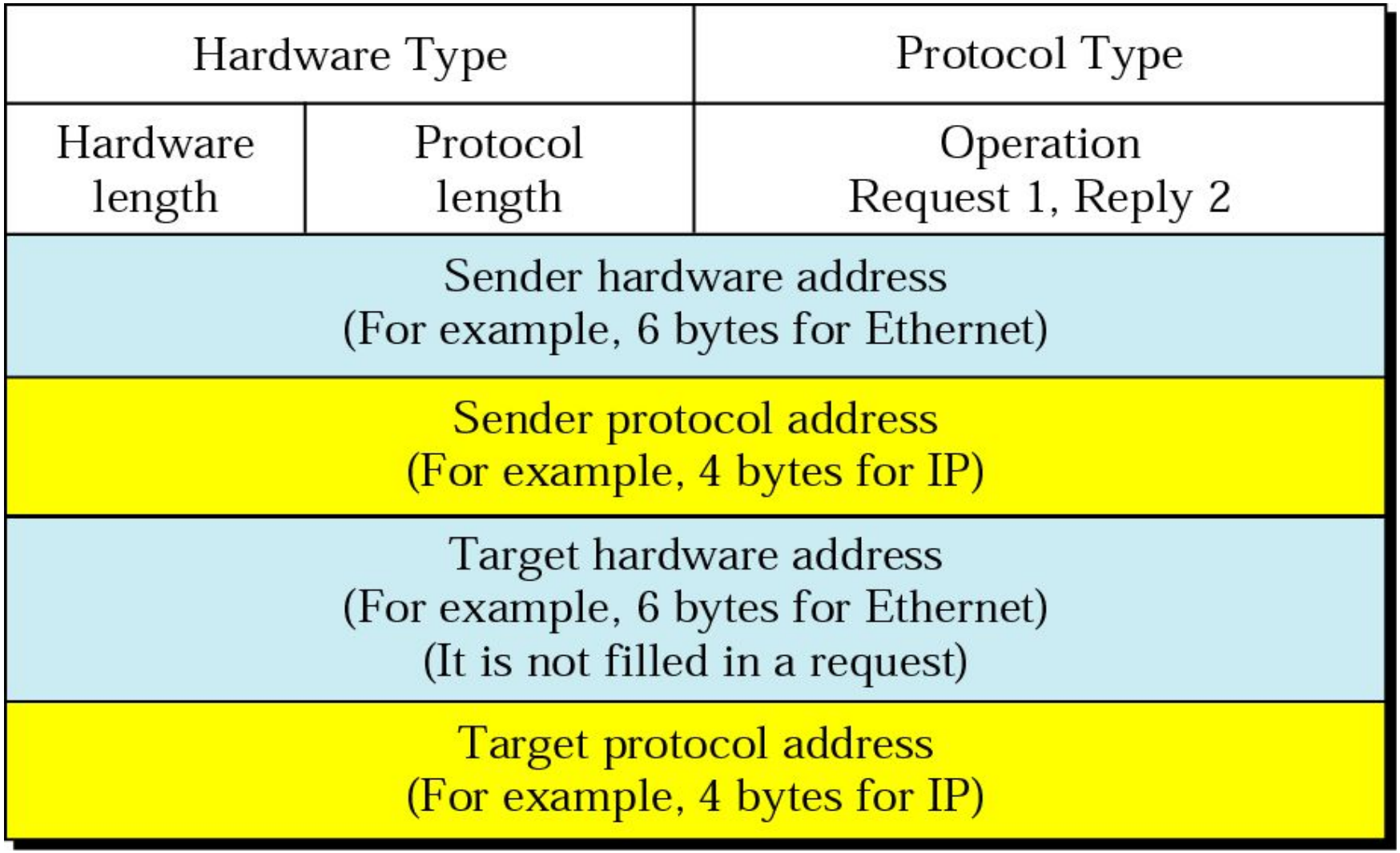


Figure 20.4 Encapsulation of ARP packet

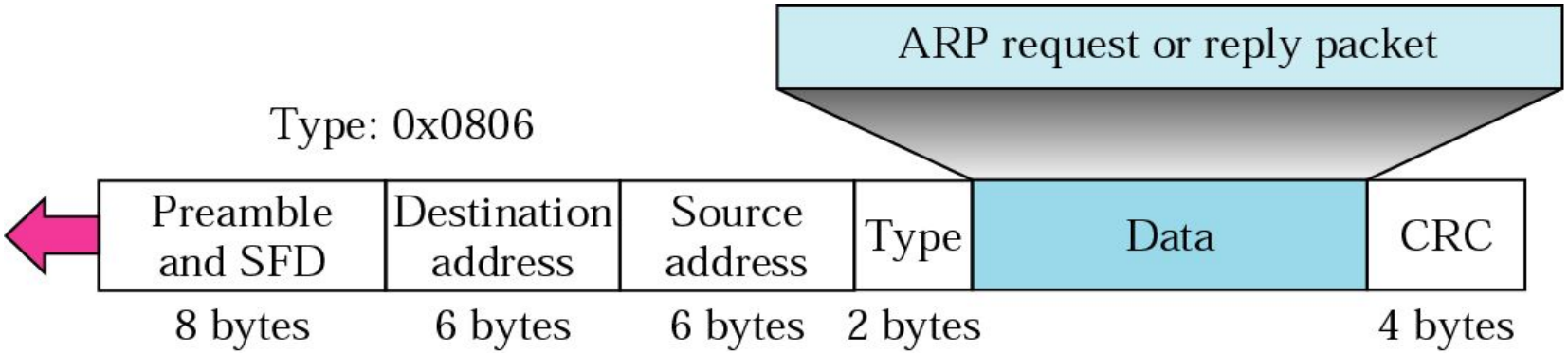
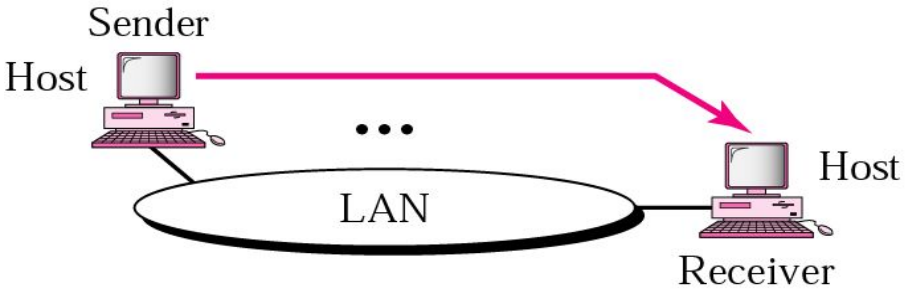
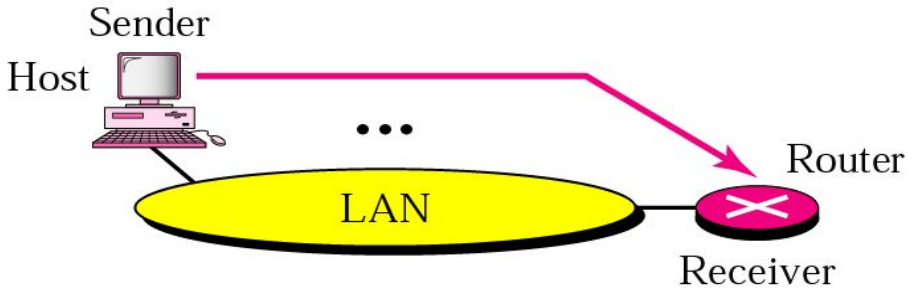


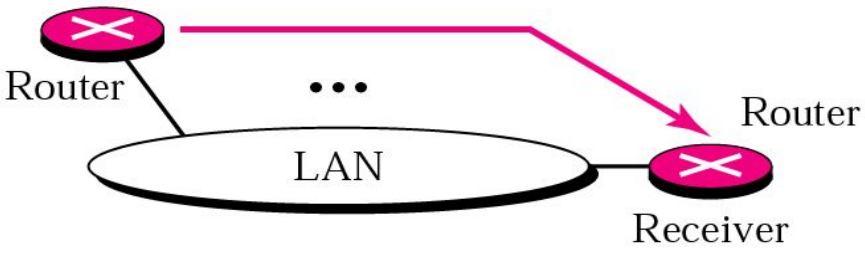
Figure 20.5 Four cases using ARP



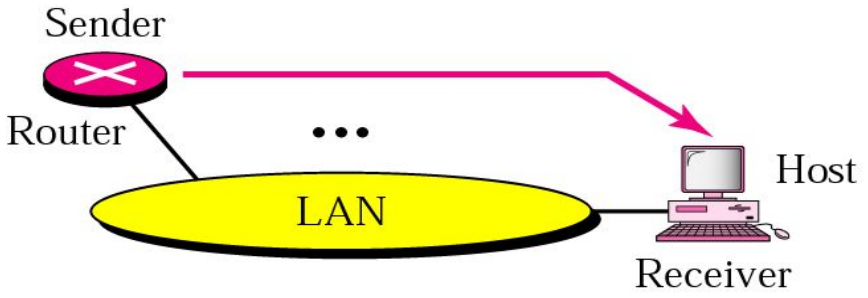
Case 1. A host has a packet to send to another host on the same network.



Case 2. A host wants to send a packet to another host on another network.
It must first be delivered to the appropriate router.



Case 3. A router receives a packet to be sent to a host on another network.
It must first be delivered to the appropriate router.



Case 4. A router receives a packet to be sent to a host on the same network.

Note:

An ARP request is broadcast; an ARP reply is unicast.

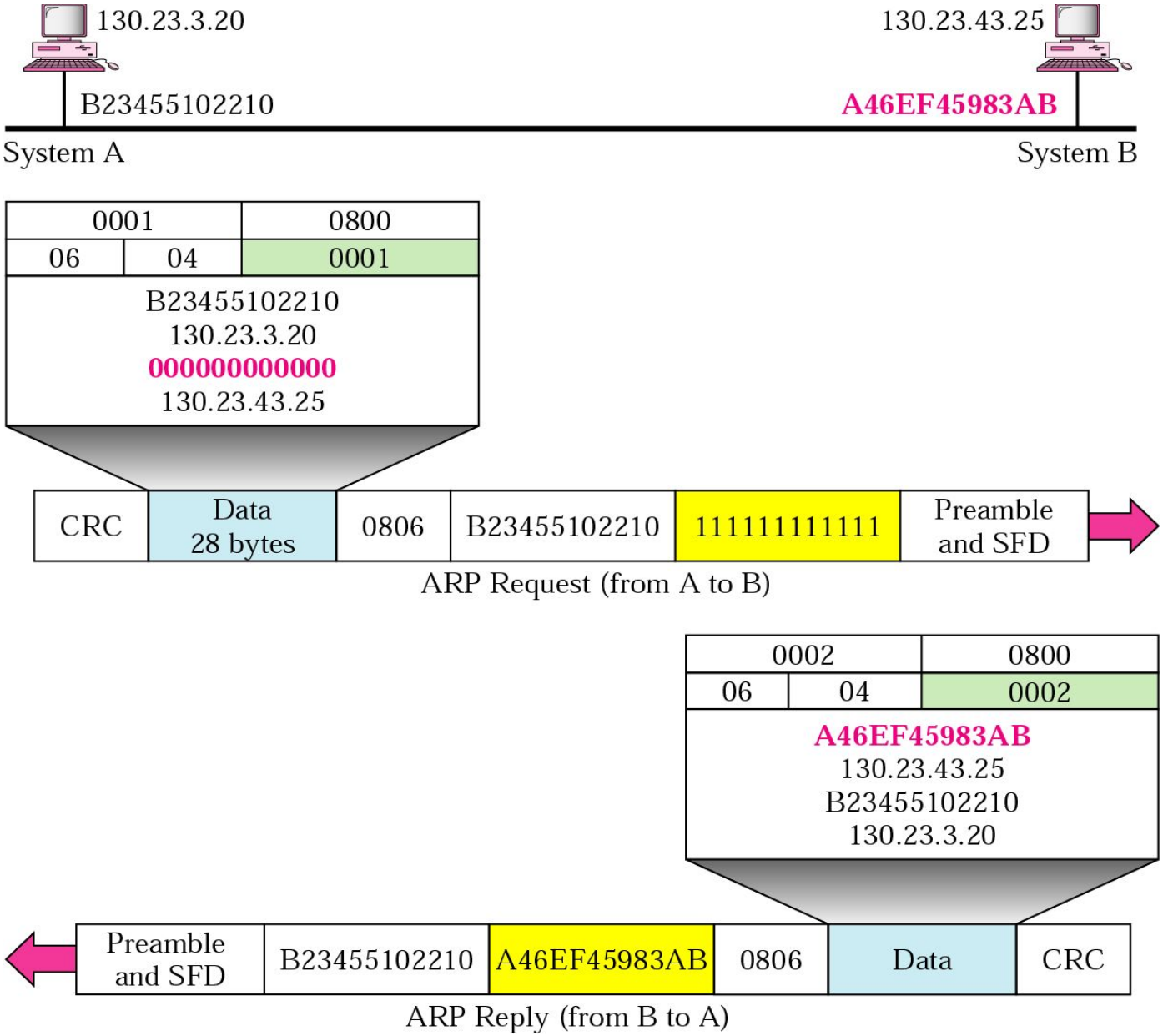
Example 1

A host with IP address 130.23.3.20 and physical address B23455102210 has a packet to send to another host with IP address 130.23.43.25 and physical address A46EF45983AB. The two hosts are on the same Ethernet network. Show the ARP request and reply packets encapsulated in Ethernet frames.

Solution

Figure 20.6 shows the ARP request and reply packets. Note that the ARP data field in this case is 28 bytes, and that the individual addresses do not fit in the 4-byte boundary. That is why we do not show the regular 4-byte boundaries for these addresses. Note that we use hexadecimal for every field except the IP addresses.

Figure 20.6 Example 1



Routing

Routing

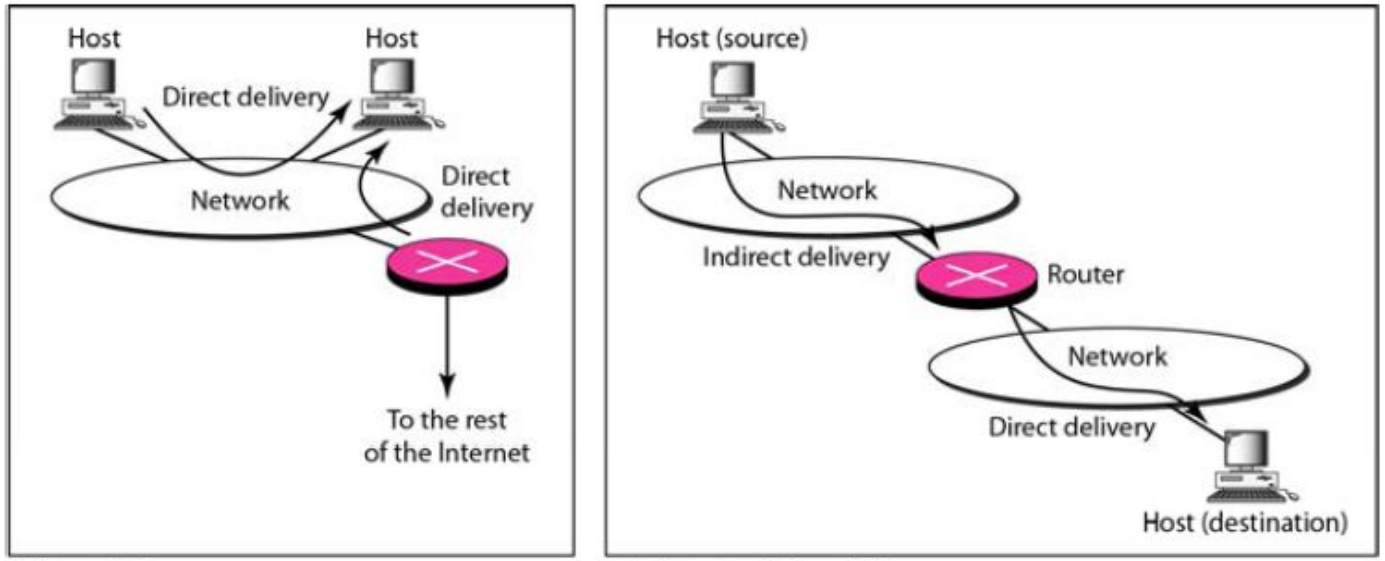
- Delivery: Network layer supervises the handling of packets by the underling physical network.
- Forwarding: the way the packets are delivered to the next station.
- Routing: the way routing tables are created to help in forwarding.

Delivery

- Direct Delivery: occurs when Source & Destination, both are on same network.
- Indirect Delivery: occurs when Source & Destination, both are on different network. Packet goes from router to router to reach the destination.

Direct & Indirect delivery

Figure 22.1 *Direct and indirect delivery*



a. Direct delivery

b. Indirect and direct delivery

Routing

- Every router has a table. When a packet comes to forward it to destination, route is consulted from routing table.
- Routing Techniques:
  - Next-hop
  - Route
  - Network-specific
  - Host-specific

Next-Hop Method & Route Method

Figure 22.2 *Route method versus next-hop method*

a. Routing tables based on route

Destination	Route
Host B	R1, R2, host B

Routing table for host A

Destination	Route
Host B	R2, host B

Routing table for R1

Destination	Route
Host B	Host B

Routing table for R2

b. Routing tables based on next hop

Destination	Next hop
Host B	R1

Destination	Next hop
Host B	R2

Destination	Next hop
Host B	---

Host-Specific & Network-Specific

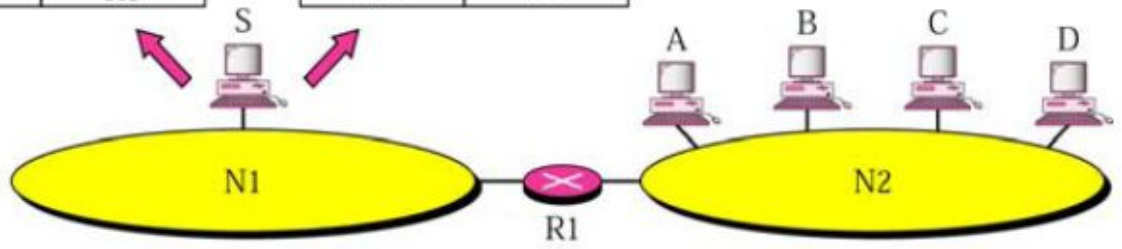
- Instead of having an entry for each host connected to the same physical network, we have only one entry to define the address of the network itself.

Routing table for host S based on host-specific routing

Destination	Next Hop
A	R1
B	R1
C	R1
D	R1

Routing table for host S based on network-specific routing

Destination	Next Hop
N2	R1



Forwarding Process

- In classless addressing, we need at least 4 columns in a routing table.
- Destination address does not give any information about the network, so mask is included.

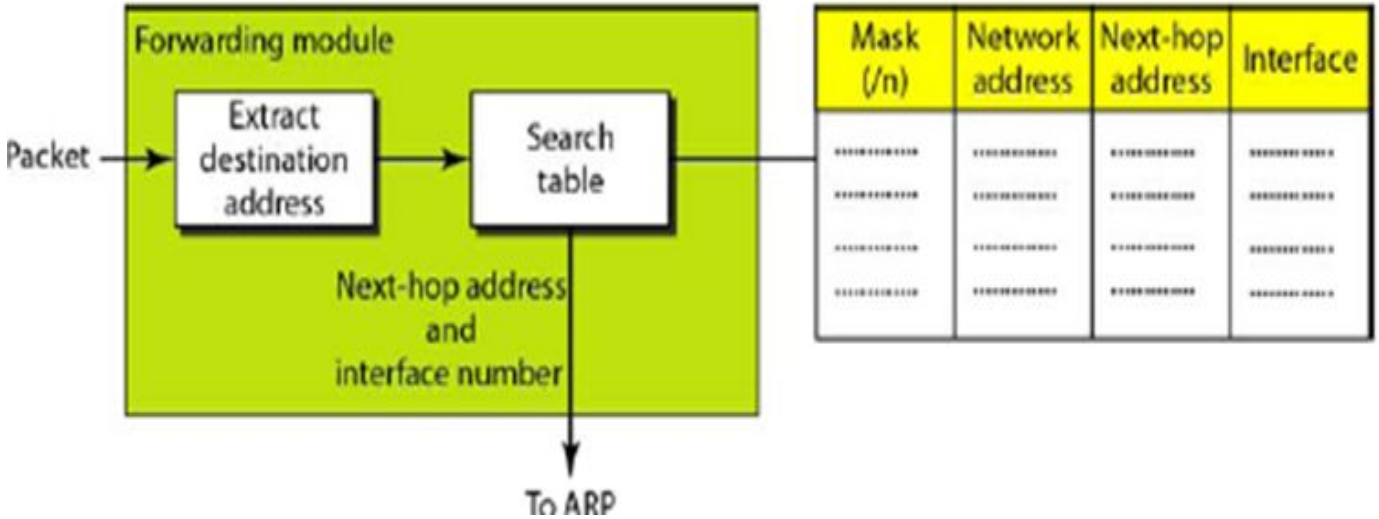
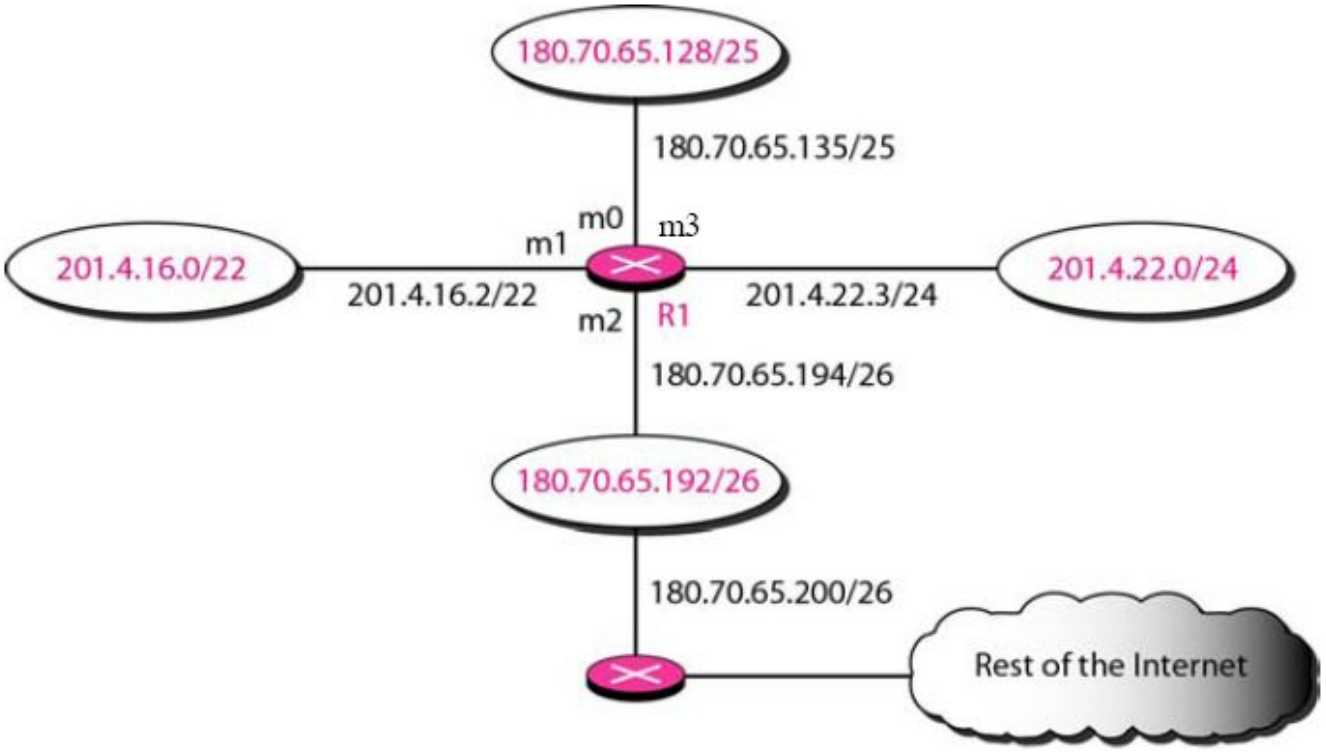


Figure 22.6 *Configuration for Example 22.1*



Example 22.1

Make a routing table for router R1, using the configuration in Figure 22.6.

Solution

Mask	*Network Address*	*Next Hop*	*Interface*
/26	180.70.65.192	—	m2
/25	180.70.65.128	—	m0
/24	201.4.22.0	—	m3
/22	201.4.16.0	....	m1
Any	Any	180.70.65.200	m2

Example 22.2

Show the forwarding process if a packet arrives at R1 in Figure 22.6 with the destination address 180.70.65.140.

Solution

1. *Applying the first mask (/26) does not match the corresponding network address.*
2. *The second mask (/25) is applied to the destination address. The result is 180.70.65.128, which matches the corresponding network address. The next-hop address and the interface number m0 are passed to ARP for further processing.*

Example 22.3

Show the forwarding process if a packet arrives at R1 in Figure 22.6 with the destination address 201.4.22.35.

Solution

The router performs the following steps:

1. ***Applying the first and the second masks does not match the corresponding network address.***
2. ***The third mask (/24) is applied to the destination address. The result is 201.4.22.0, which matches the corresponding network address. The destination address of the packet and the interface number m3 are passed to ARP.***

Example 22.4

Show the forwarding process if a packet arrives at R1 in Figure 22.6 with the destination address 18.24.32.78.

Solution

This time all masks are applied, one by one, to the destination address, but no matching network address is found.

When it reaches the end of the table, the module gives the next-hop address 180.70.65.200 and interface number m2 to ARP. This is probably an outgoing package that needs to be sent, via the default router, to someplace else in the Internet.

Hierarchical Routing

- Internet is divided into International, National, Regional, Local – ISP.
- Local can be further divided into smaller blocks.
- If block has been assigned a.b.c.d/n, then all customers of local ISP are identified as a.b.c.d/n to the rest of the Internet.

Geographical Routing

- Divide the whole address block into few larger blocks.
- E.g. A block to Asia, A block to Russia, etc.
- Routers of ISPs outside the Asia will have only one entry for the packets to Asia in their routing table.

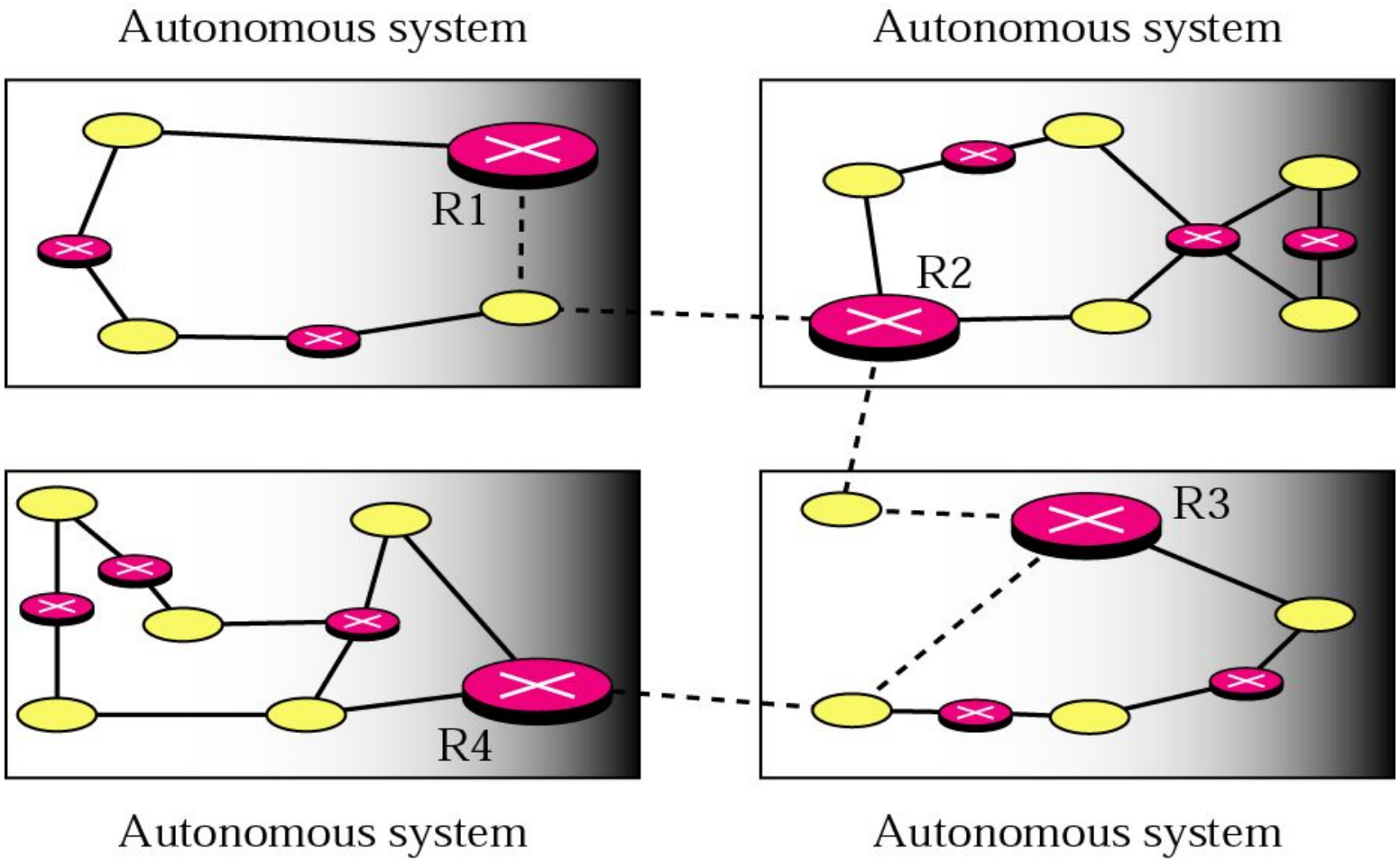
Routing Table

- Static: Information entered, maintained and updated manually by administrator.
  - Can be used in small internets that doesn’t change oftenly.
- Dynamic: updated periodicaly by using some routing protocol e.g. RIP, OSPF or BGP.

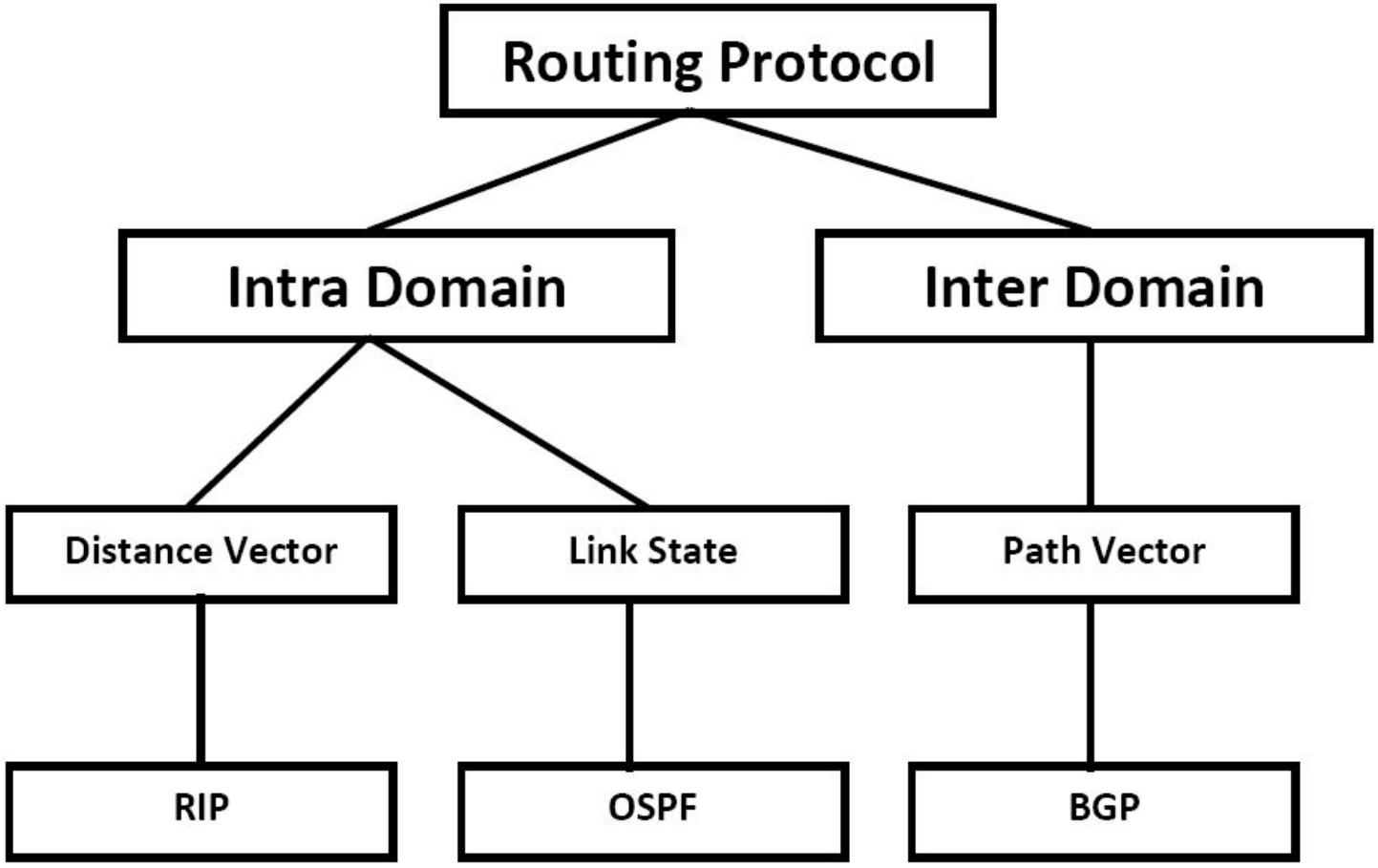
Unicast Routing

- As internet is large enough – difficult to handle the routing tables.
- So internet is divided into **Autonomous Systems (AS) .**
- AS is group of N/w or routers.
- Within an AS single administration is there.
- Routing inside AS: Intra Domain Routing
- Routing between AS: Inter Domain Routing

Autonomous Systems (AS)



Unicast Routing:
Routing Protocols



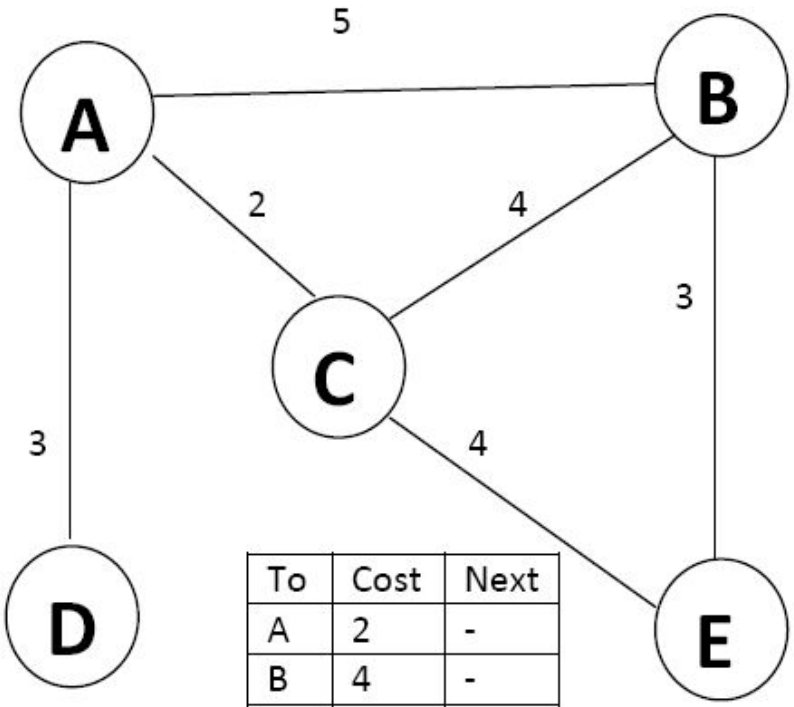
Distance Vector (Table) Routing

- Nodes use Bellman Ford's Algo to build routing table
- Least cost route between two nodes = route with min dist.
- Each node maintains the vector (table) of min dist to every node.
- Initialization:
  - Each node knows only the dist between itself & immediate neighbor.
  - Each node send a message to each neighbor to know the dist between itself & the neighbor.
  - The dist of any node which is not immediate neighbor = ∞

Distance Vector Routing

- Sharing:
  - Each node shares its routing table with its immediate neighbor – either periodically (periodic update) or when there is a change (triggered update).
  - A can know about E only when C shares its table with A.

Distance Vector Routing



To	Cost	Next
A	0	-
B	5	-
C	2	-
D	3	-
E	6	C

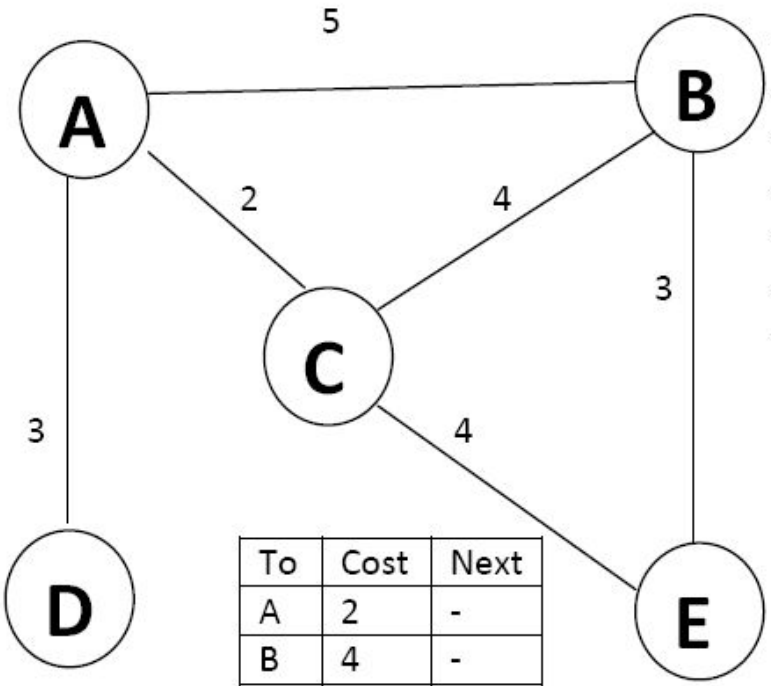
To	Cost	Next
A	5	-
B	0	-
C	4	-
D	8	A
E	3	-

To	Cost	Next
A	3	-
B	8	A
C	5	A
D	0	-
E	9	A

To	Cost	Next
A	2	-
B	4	-
C	0	-
D	5	A
E	4	-

To	Cost	Next
A	6	C
B	3	-
C	4	-
D	9	C
E	0	-

Distance Vector Routing Initialization



To	Cost	Next
A	0	-
B	5	-
C	2	-
D	3	-
E	∞	

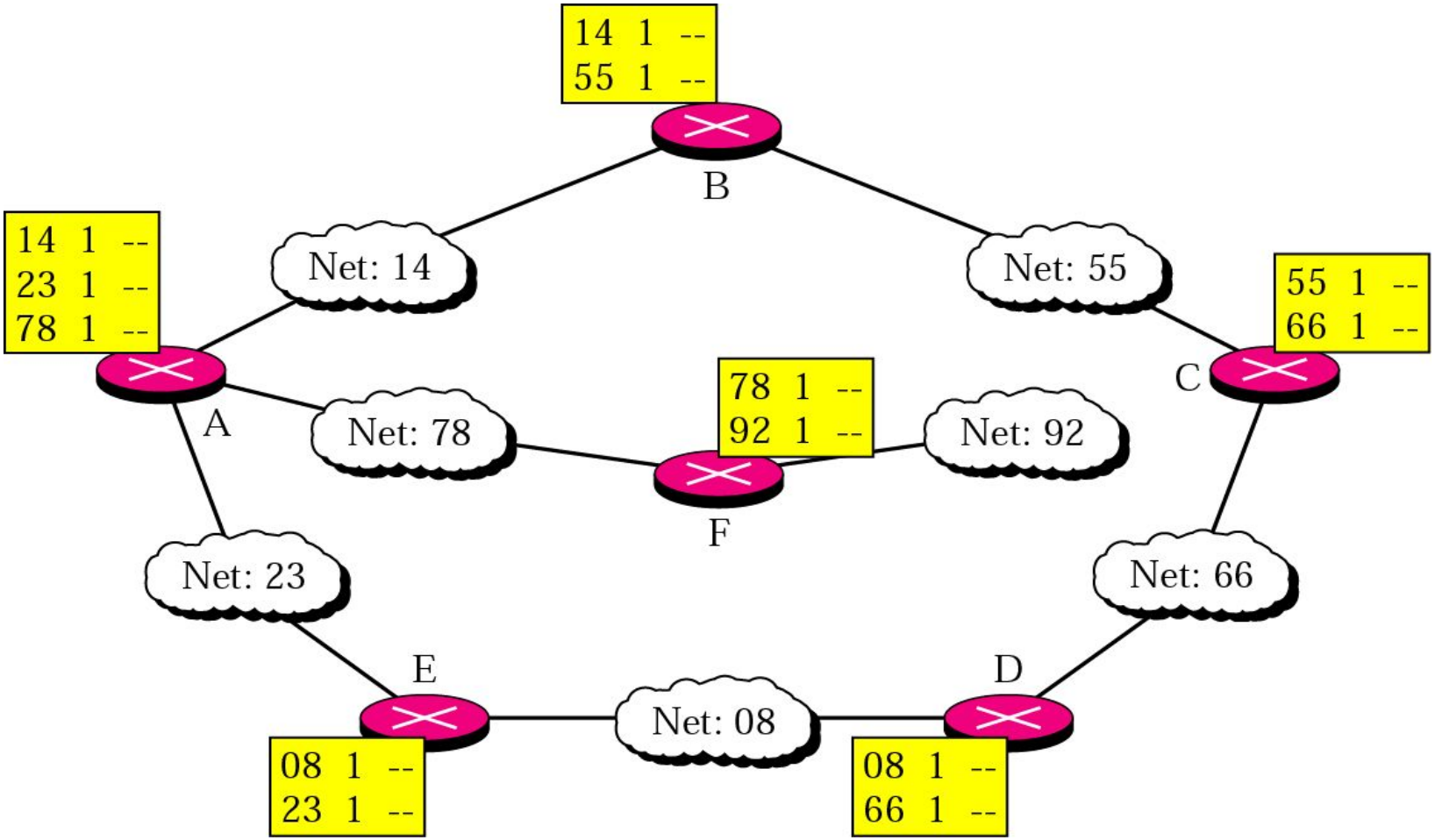
To	Cost	Next
A	5	-
B	0	-
C	4	-
D	∞	
E	3	-

To	Cost	Next
A	3	-
B	∞	
C	∞	
D	0	-
E	∞	

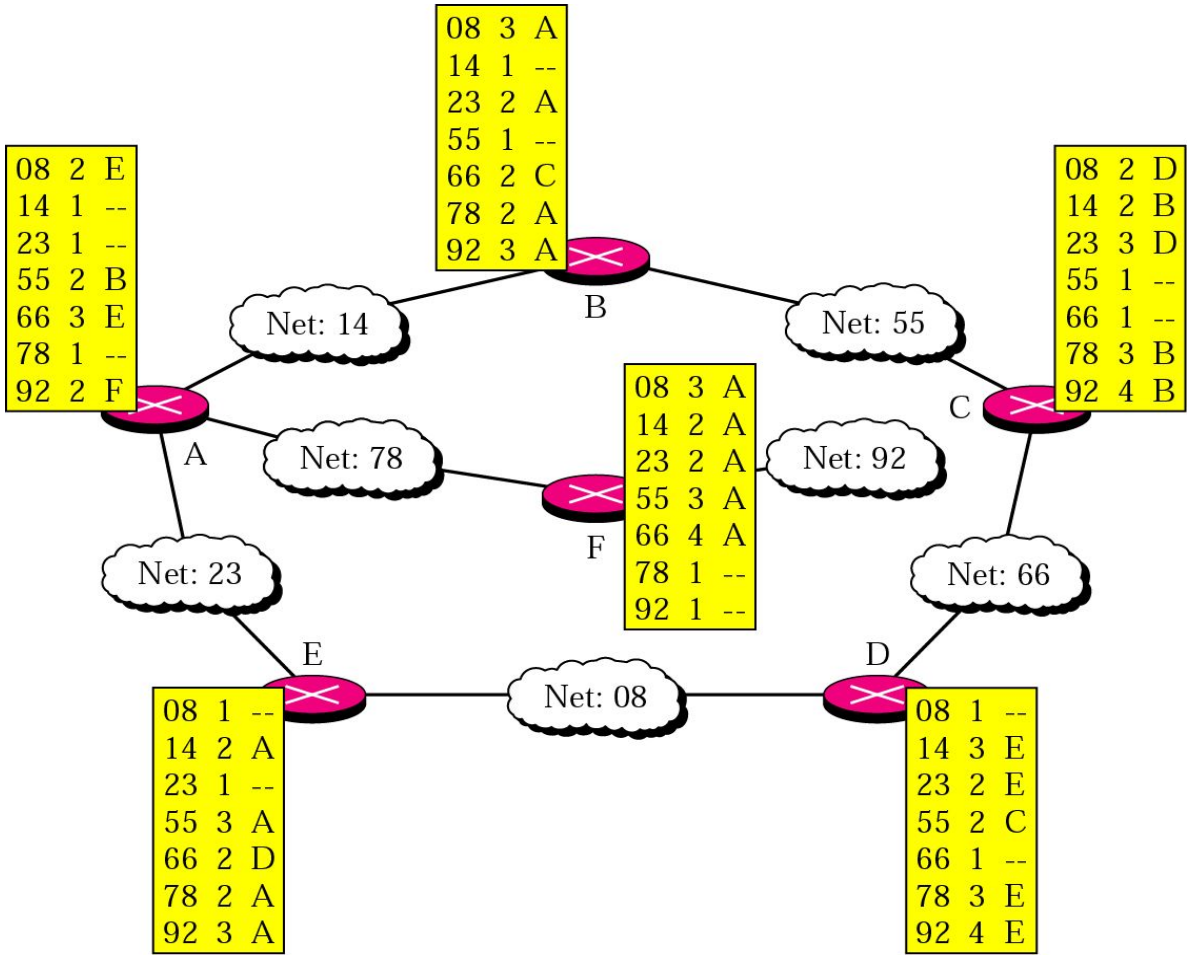
To	Cost	Next
A	2	-
B	4	-
C	0	-
D	∞	
E	4	-

To	Cost	Next
A	∞	
B	3	B
C	4	C
D	∞	
E	0	D

Initial routing tables in a small autonomous system



Final routing tables for above Figure



Routing Information Protocol (RIP)

- Based on DV routing
  - In an AS, only routers are dealt with. Each router has a routing table.
  - Destination in a routing table is a network; means first column defines a network address.
  - Metric (distance) used in RIP is hop count.
  - Infinite is defined as 16. means any route can have max 15 hops.
  - Next = defines the address of next router along the path to reach the destination.

Link State Routing

- Nodes use Dijkastra's algo to build routing table.
- Unlike DV, in LS, each node uses entire topology but routing table for each node is unique.
- As each person is having same map of city but everyone is opting different route to reach his destination.
- Topology must be dynamic= represent latest state of links and router (any link or router may be down).

Link State Routing contd..

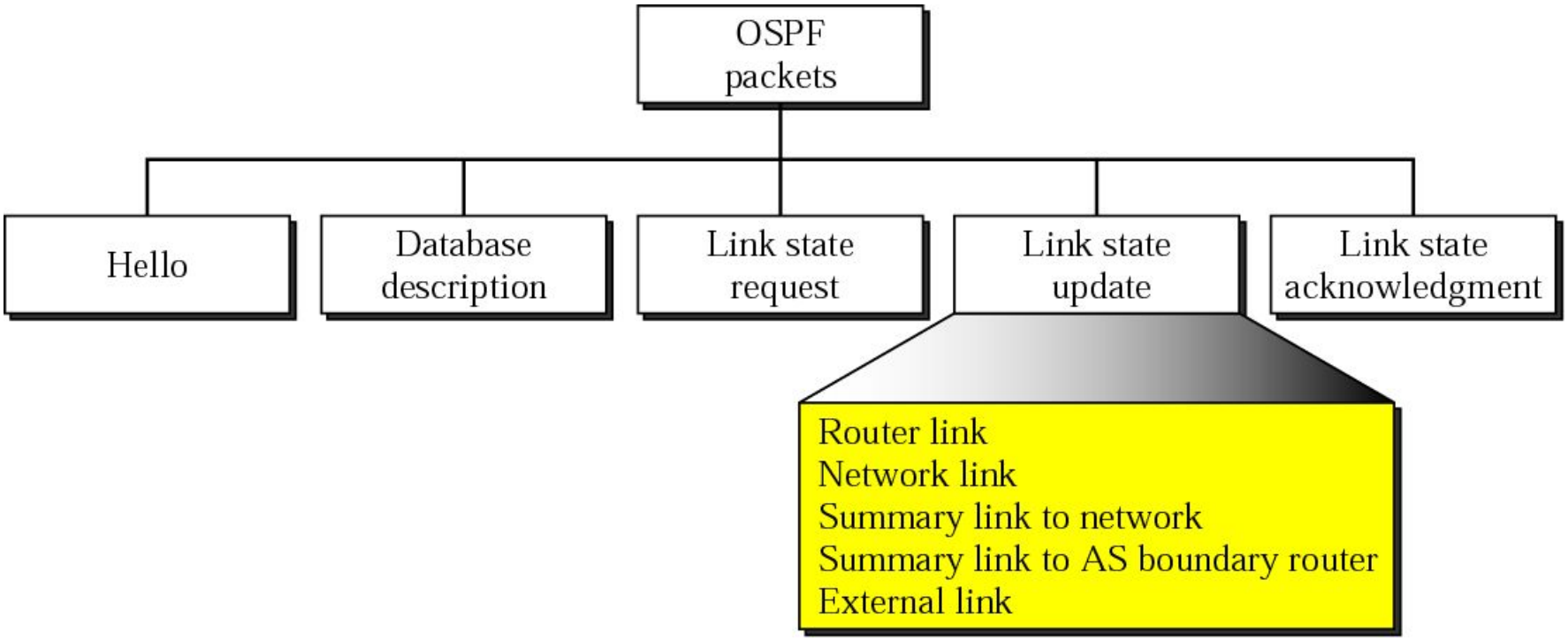
Each router must do the following:

1. Discover its neighbors, learn their network address.
2. Measure the delay or cost to each of its neighbors.
3. Construct a packet (LSP) telling all it has just learned.
4. Send this packet to all other routers i.e. flooding.
5. Compute the shortest path to every other router.
6. Calculation of routing table on the basis of shortest path tree.

Link State Routing contd..

- LSP:
  - Has min four fields:
    - *Node id* & *list of links* – to make topology.
    - *Seq. no.* distinguish new LSPs from old one.
    - *Age* is TTL.
  - Is generated
    - When there is a change in topology.
    - On periodic basis.

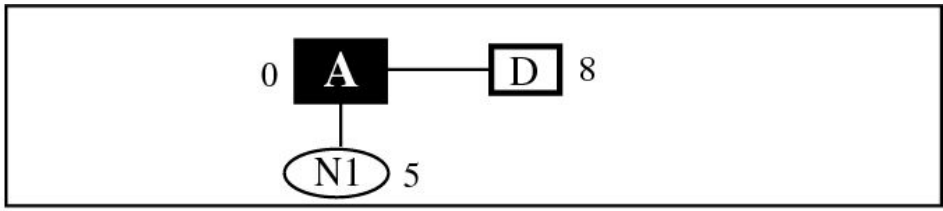
Types of LSPs



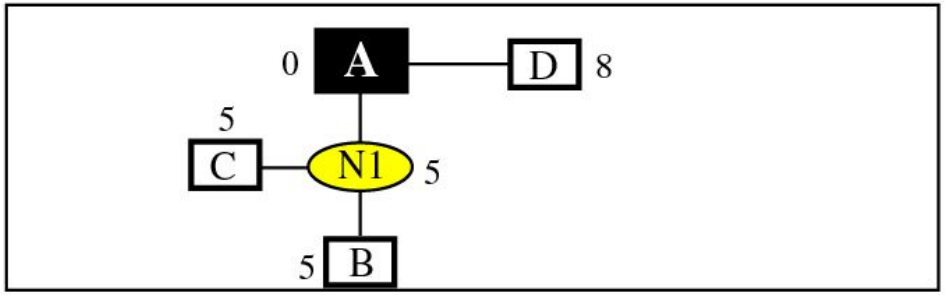
Link State Routing contd..

Shortest path calculation

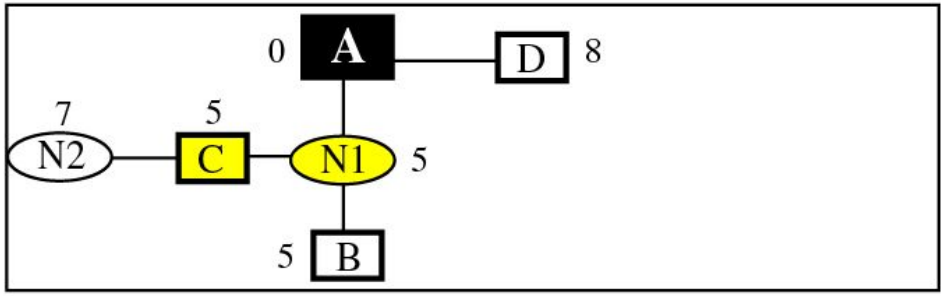
a. Start with A



b. Make A permanent, add its neighbors



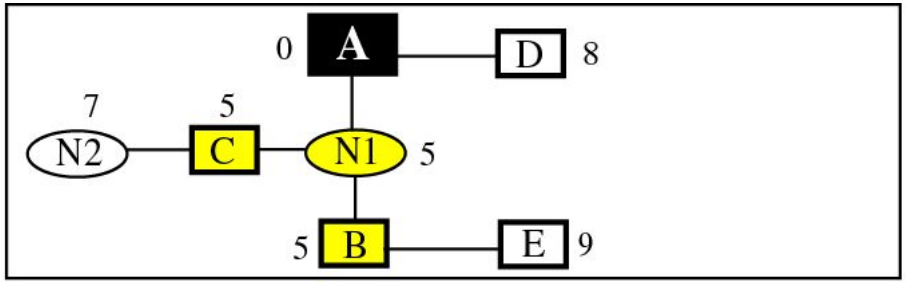
c. Make N1 permanent, add its neighbors



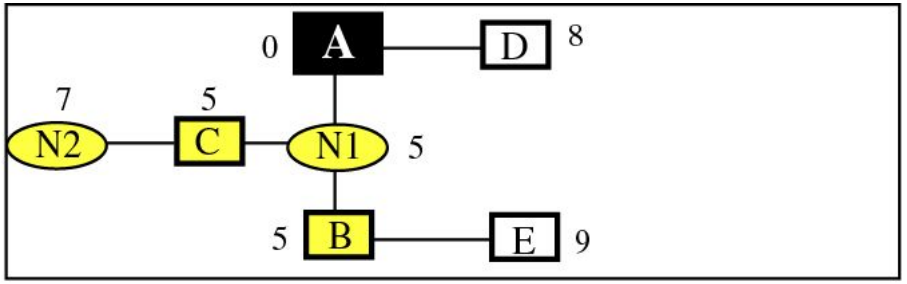
d. Make C permanent, add its neighbors

Link State Routing contd..

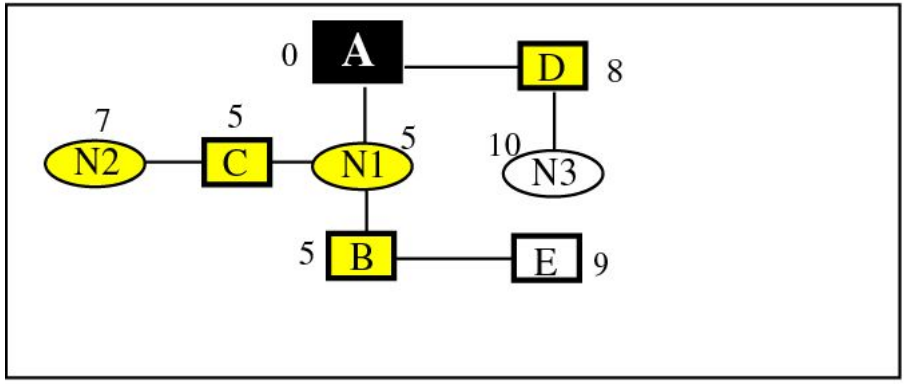
Shortest path calculation



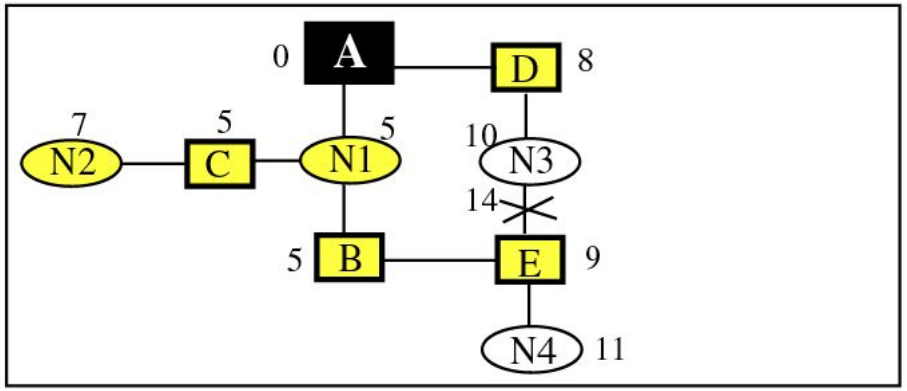
e. Make B permanent, add its neighbors



f. Make N2 permanent



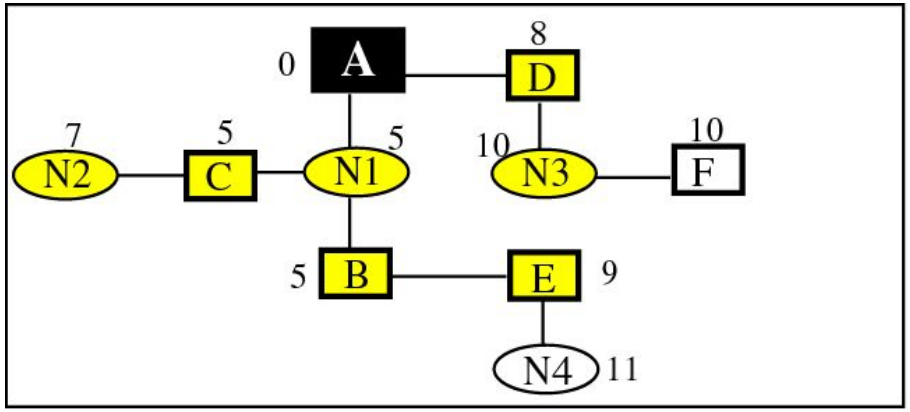
g. Make D permanent, add its neighbors



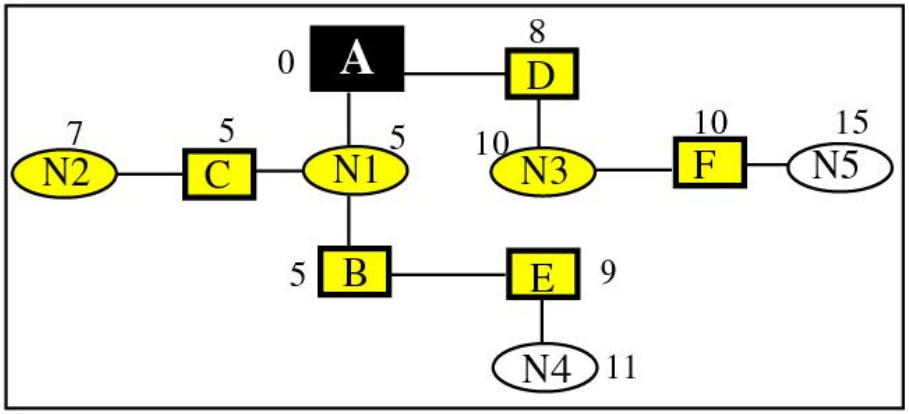
h. Make E permanent, add its neighbors

Link State Routing contd..

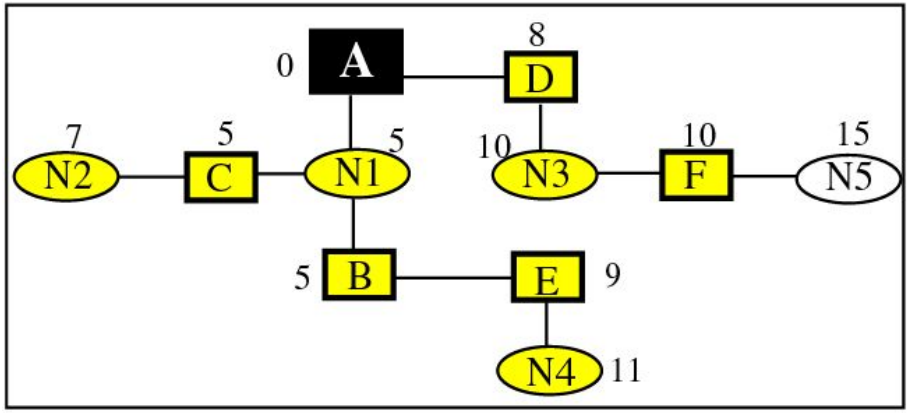
Shortest path calculation



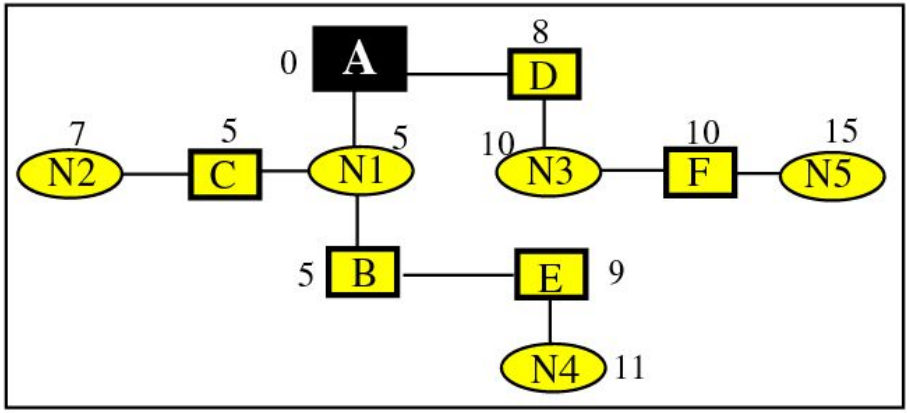
i. Make N3 permanent, add its neighbors



j. Make F permanent, add its neighbors



k. Make N4 permanent



l. Make N5 permanent

Link State Routing contd..

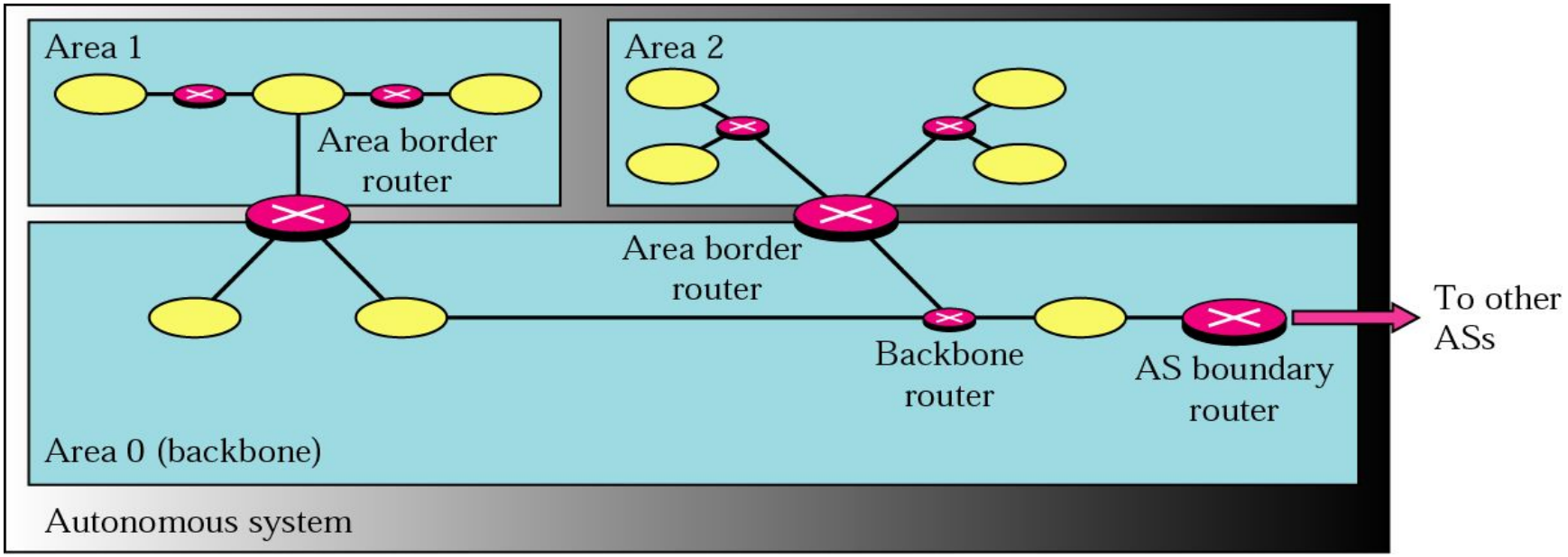
Link state routing table for router A

Network	Cost	Next Router	Other Information
N1	5		
N2	7	C	
N3	10	D	
N4	11	B	
N5	15	D	

OSPF

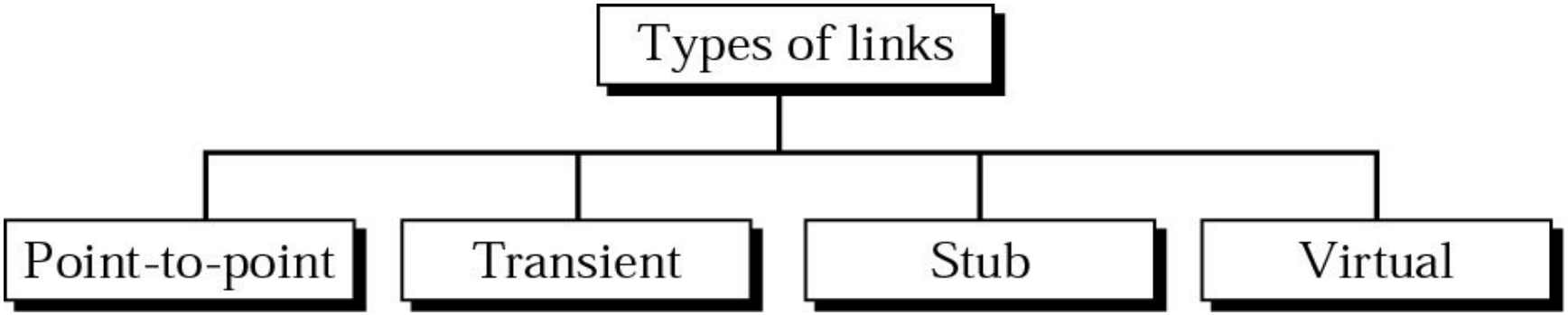
- AS is divided into **Areas** – Area is a collection of n/w, hosts, routers.
- At the border – special **area border routers** are there to summarize the information of that area.
- **AS Boundary Router** summarize the information of that AS.

OSPF contd..


Area 1
Area border
router
Area 2
Area border
router
Backbone
router
AS boundary
router
To other
ASs
Area 0 (backbone)
Autonomous system

OSPF contd..

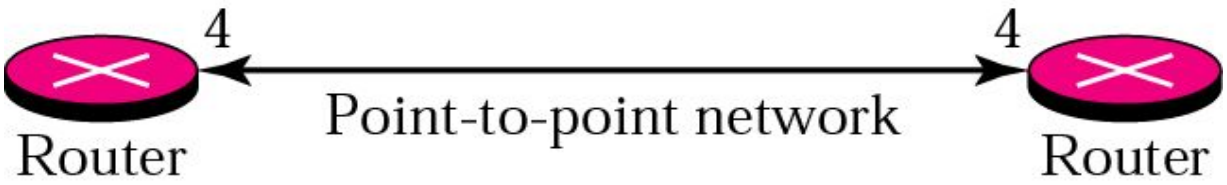
Type of Links



Type of Links contd..

Point to Point

- Connects two routers without any host in between.
- No need to assign a network address to this type of links.



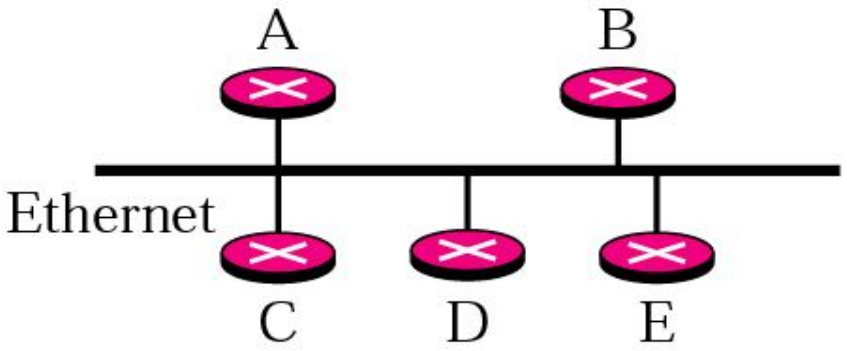
Type of Links contd..

Transient link

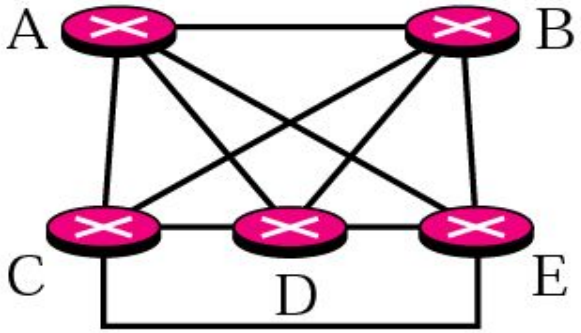
- It is a network with several routers attached to it.
- Each router has many neighbor- e.g. in the fig. (a)- A is neighbor of B,C,D & E. Similarly B,C,D & E are.
- So can be represented as fig.(b) or fig. (c)
- In fig (b) – advertisement needed more.
- In fig (c) – advertisement needed less.
- Designated routers are also true routers.

Type of Links contd..

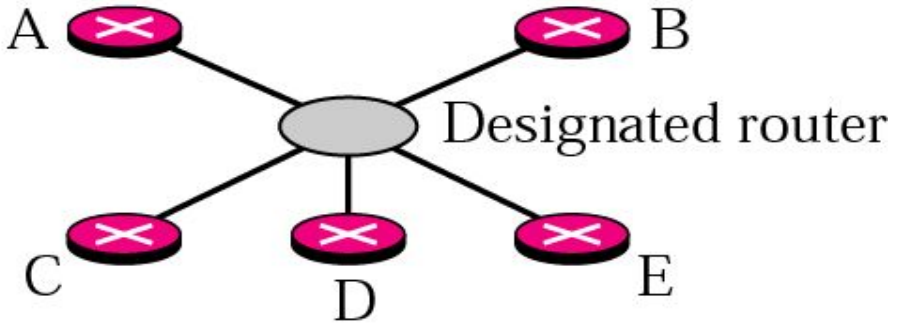
Transient link & Stub Link



a. Transient network



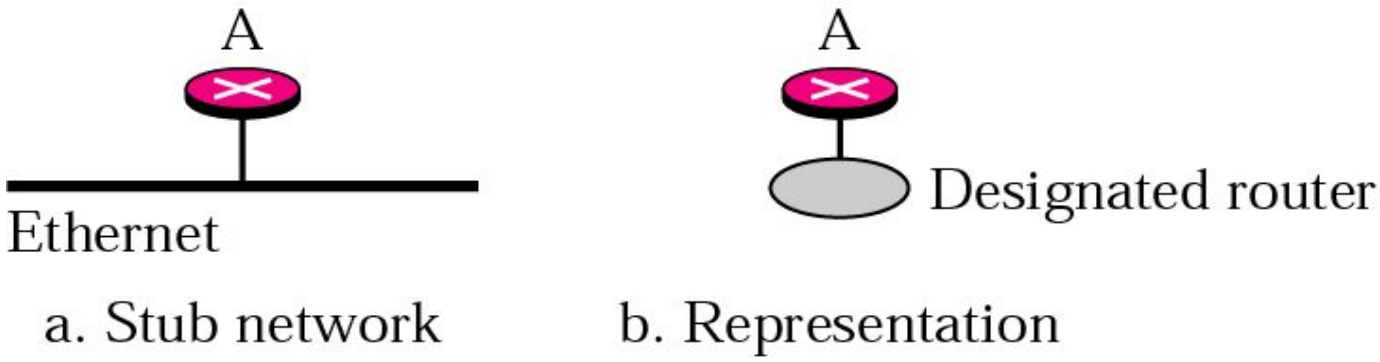
b. Unrealistic representation



c. Realistic representation

Type of Links contd..
Stub Link

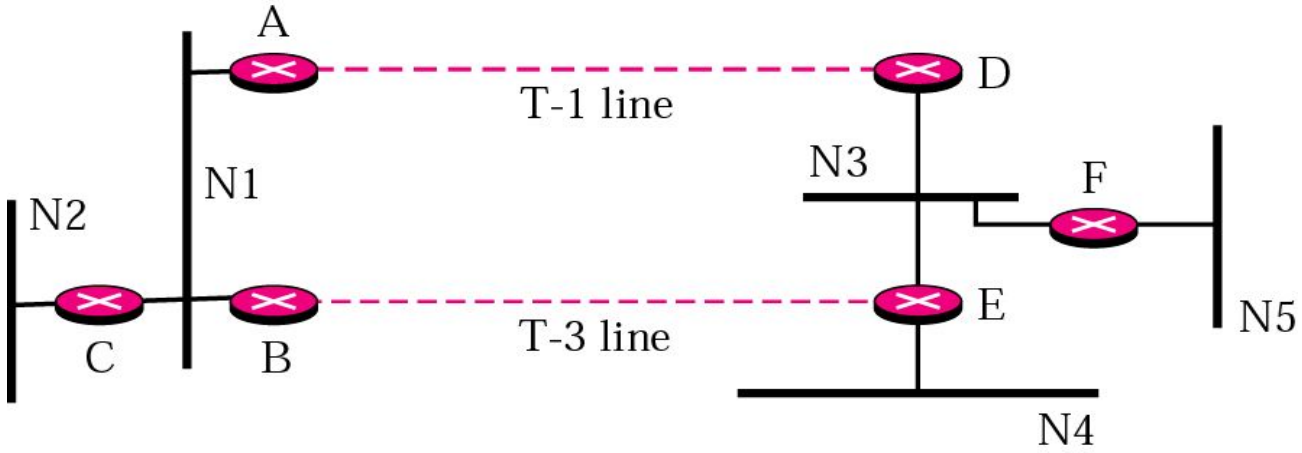
- Stub link is a network that is connected to only one router.
- This is a special case of transient links.

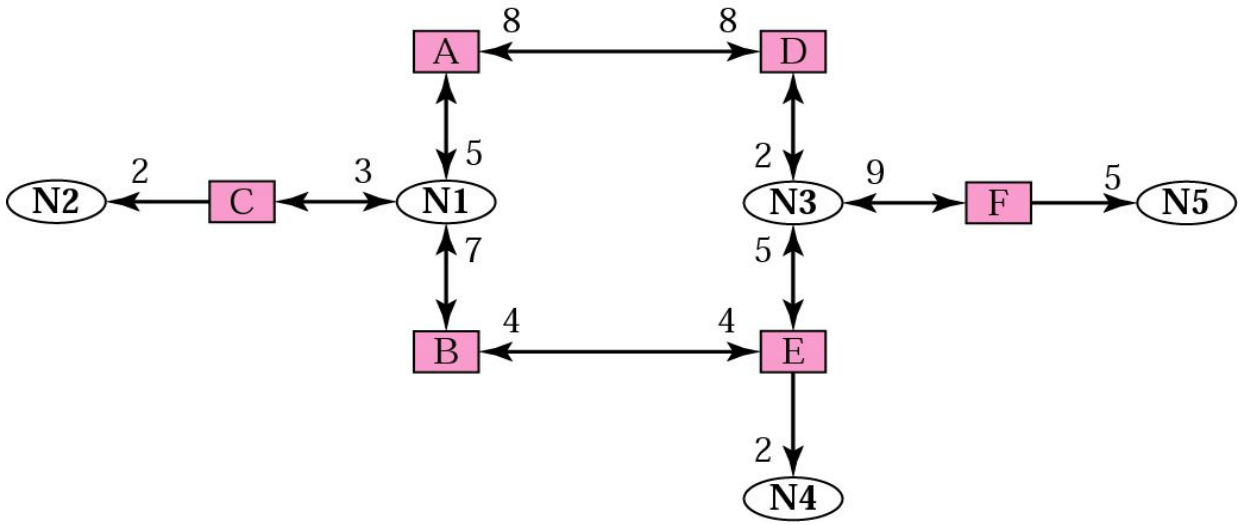


a. Stub network b. Representation

OSPF contd..

AS and its graphical representation

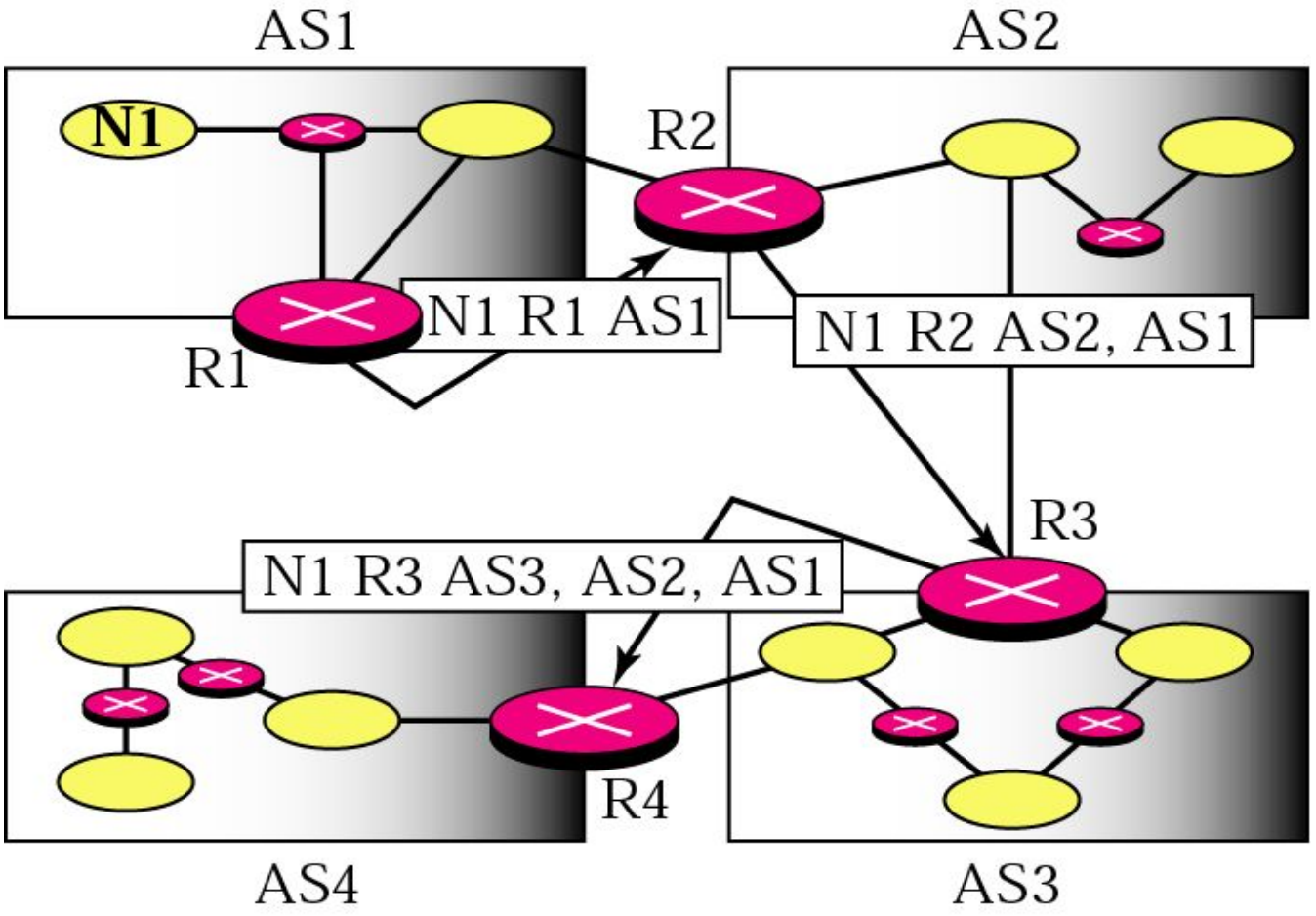




Path Vector Routing

- Inter Domain Routing.
- Each AS is assume as one node – Speaker Node
- According to speaker nodes, topology is made and then works like Distance vector i.e. Dijkastra's Algo is applied.
- E.g. R1, R2, R3 & R4 are speaker nodes.

Path Vector Routing contd..



Path Vector Routing contd..

Routing Table

Network	Next Router	Path
N01	R01	AS14, AS23, AS67
N02	R05	AS22, AS67, AS05, AS89
N03	R06	AS67, AS89, AS09, AS34
N04	R12	AS62, AS02, AS09

Type of BGP Messages