MODULE 3

NETWORK LAYER:

Internet works — IPv4 and IPv6, Transition from IPv4 to IPv6,

Sub-netting, Routing – Distance Vector Routing – Link State Routing – Routers.

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.

Address Space

- An address space is the total number of addresses used by the protocol.
- If the protocol uses N bits to define an address, the address space is 2^(N)

Note

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

Notations

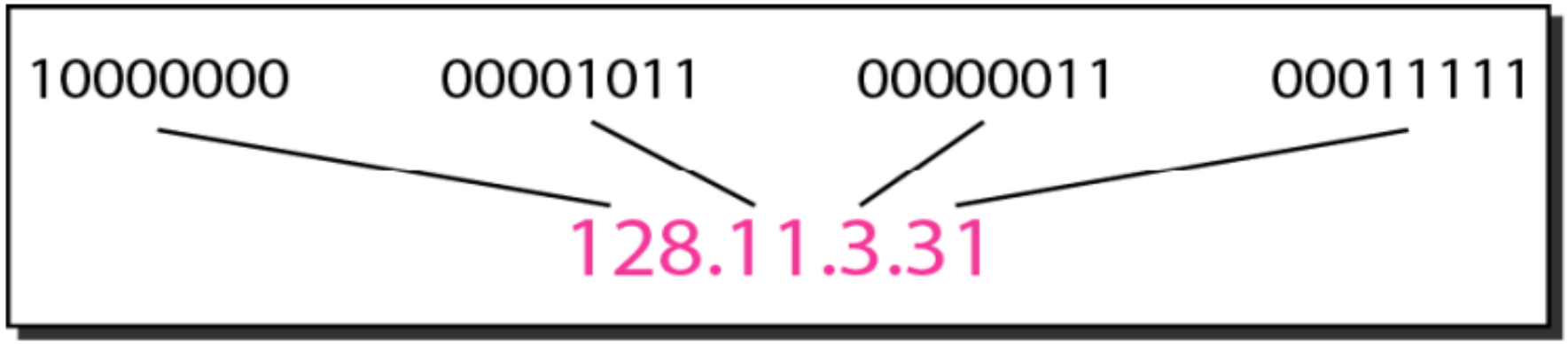
1. Binary notation

- The IPv4 address is displayed as 32 bits.
- What is an octet ?
- Hence, IPv4 address is a 32-bit address or a 4-byte address.
- Ex: 01110101 10010101 00011101 00000010

2. Dotted – Decimal Notation

- It’s compact and easier to read.
- Address is written in decimal form with a decimal point separating the bytes
- Ex. 117.149.29.2

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



Example 19.1

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

a. 10000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

ANSWER:

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

ANSWER:

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.

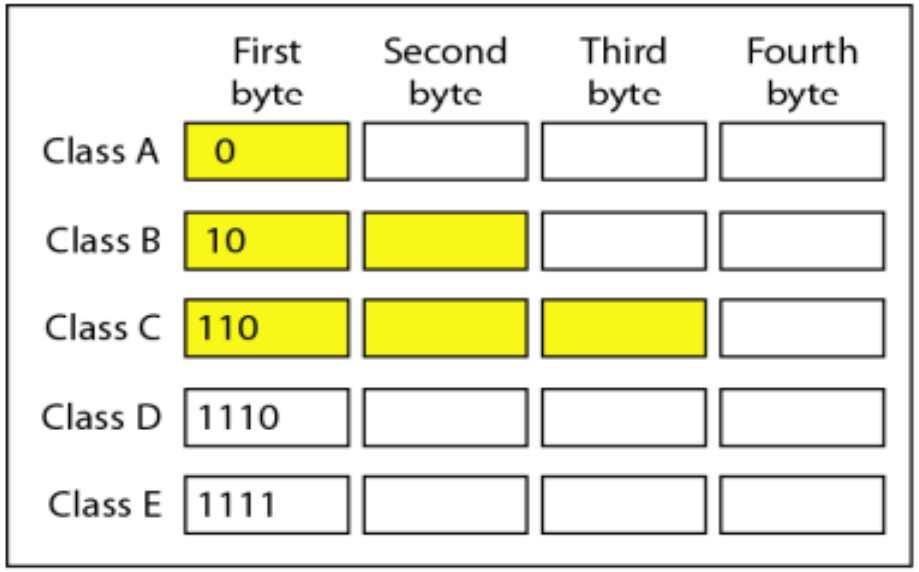
c. Each number needs to be less than or equal to 255.

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

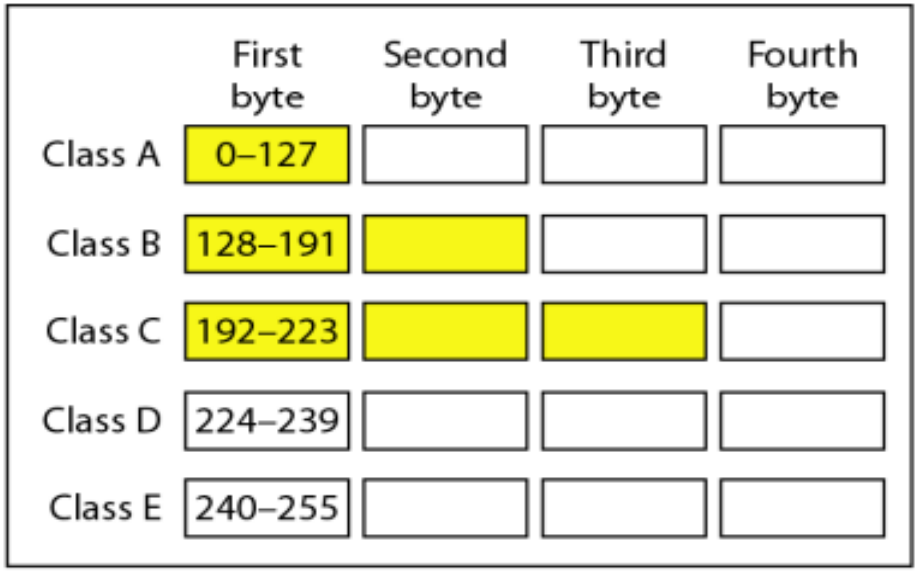
Classful Addressing

- At first, IPv4 addressing used the concept of classes
- Here, the address space is divided into 5 classes: A,B,C,D and E
- Each class occupies some part of the address space.

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



a. Binary notation



b. Dotted-decimal notation

Example 19.4

Find the class of each address.

a. 00000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

c. 14.23.120.8

d. 252.5.15.111

Answer:

a. The first bit is 0. This is a class A address.

b. The first 2 bits are 1; the third bit is 0. This is a class C address.

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

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

Classes and Blocks

A disadvantage of Classful addressing is that each class is divided into a fixed number of blocks with each block having a fixed size.

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

Note

In classful addressing, a large part of the available addresses were wasted.

Netid and Hostid

- In classful addressing, an IP address in class A,B or C is divided into netid and hostid.
- These parts are of varying lengths, depending on the class of the address.
- MASK:Although the length of the netid and hostid is predetermined in classful addressing, we can also use a MASK (Default Mask), a 32-bit number made of contiguous 0s

- CIDR : Classless Interdomain Routing Notation or Slash Notation

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

Note

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

SUBNETTING

If an organization was granted a large block in class A or B, it could divide the addresses into several contiguous groups and assign each group to smaller network – called subnets.

SUPERNETTING

In Supernetting, an organization can combine several class C blocks to create a larger range of addresses.

Several networks are combined to create a super network or a supernet.

Note

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

Classless Addressing

- To overcome address depletion and give more organizations access to the Internet classless addressing was designed and implemented.
- Here, there are no classes , but the addresses are still granted in blocks.
- The size of the block varies based on the nature and size of the entity.

Restrictions

- There are 3 restrictions:

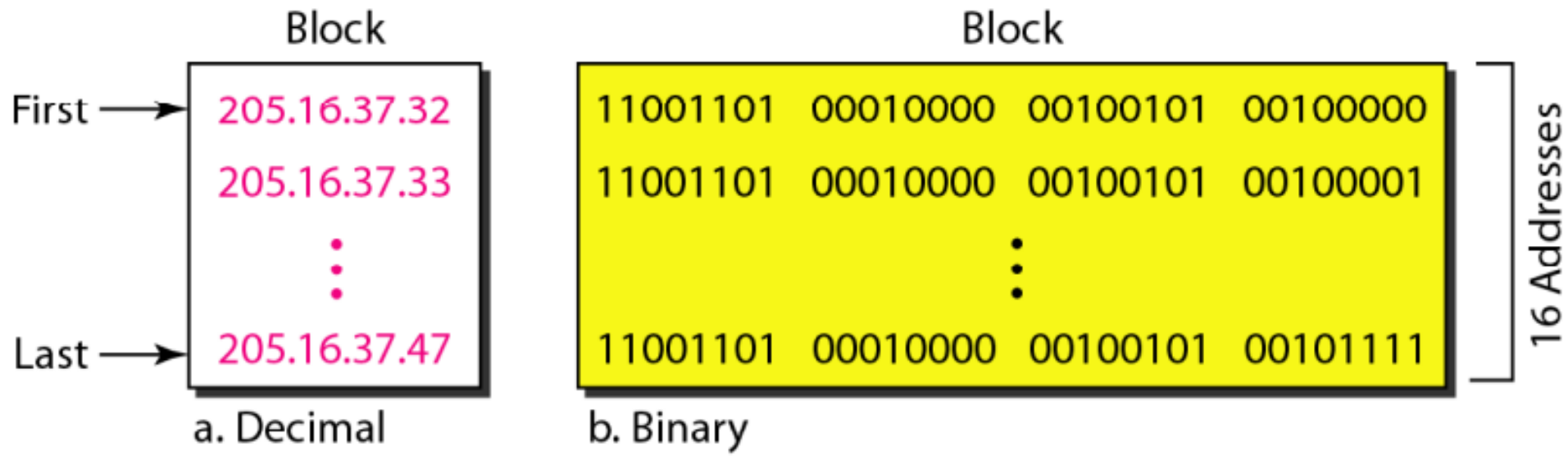
1. The address in the block must be contiguous, one after the other
2. The number of addresses in a block must be power of 2
3. The first address must be evenly divisible by the number of addresses.

Example 19.5

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 ($16 = 2^4$), and the first address is divisible by 16. The first address, when converted to a decimal number, is 3,440,387,360, which when divided by 16 results in 215,024,210.

Figure 19.3 *A block of 16 addresses granted to a small organization*



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.

3 values to know

- First Address
- Last Address
- Number of addresses in a block

Note

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

Example 19.6

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

Solution

The binary representation of the given address is

11001101 00010000 00100101 00100111

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

11001101 00010000 00100101 0010000

or

205.16.37.32.

This is actually the block shown in Figure 19.3.

Note

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

Example 19.7

Find the last address for the block in Example 19.6.

Solution

The binary representation of the given address is

11001101 00010000 00100101 00100111

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.

Note

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

Example 19.8

Find the number of addresses in Example 19.6.

Solution

The value of n is 28, which means that number of addresses is 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. In Example 19.5 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.

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

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:	**000000000 00000000 00000000 00001111**
Number of addresses:	15 + 1 = 16

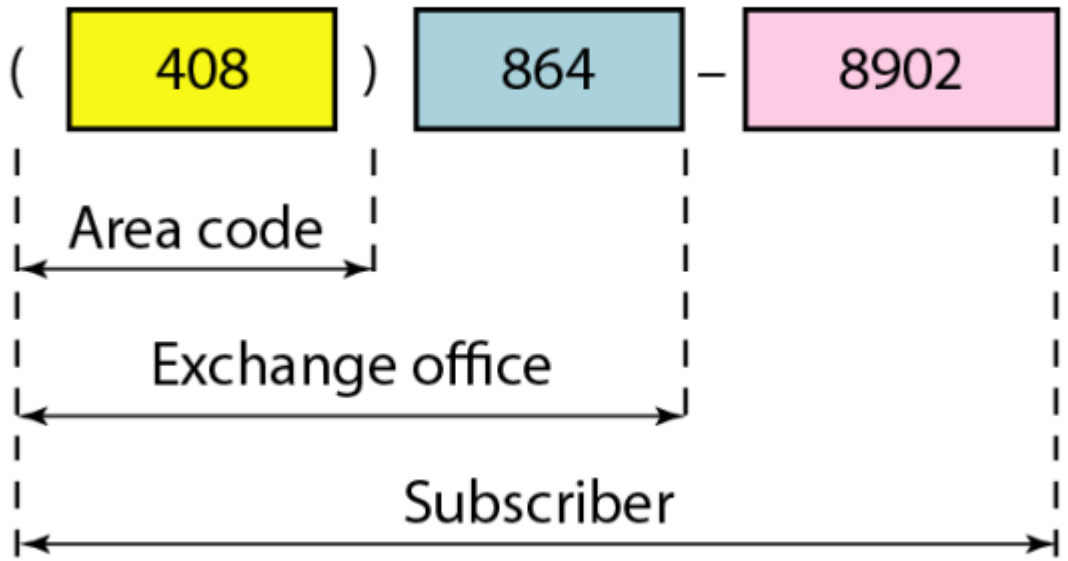
Network Addresses

- First Address is special and defines the organization network.
- The first address is used by routers to direct the message sent to the organization from outside.

Note

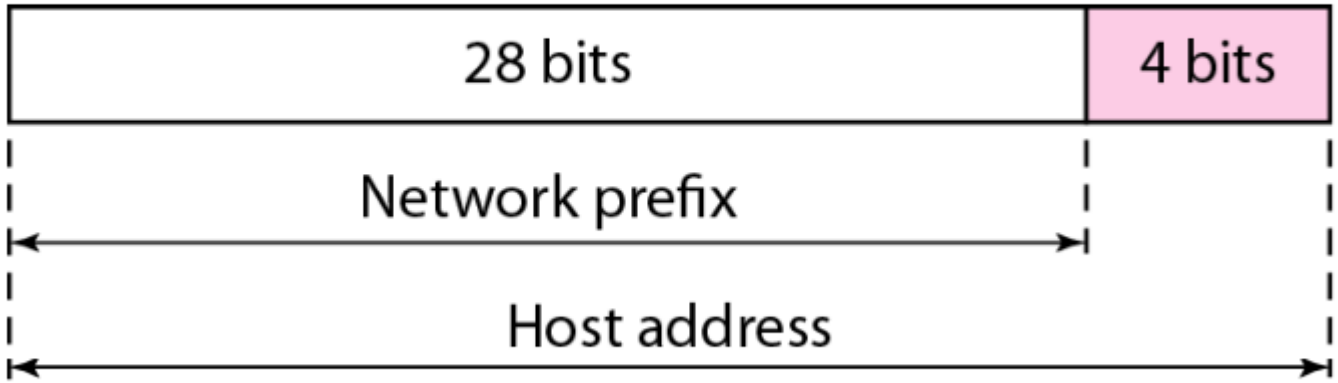
The first address in a block is normally not assigned to any device; it is used as the network address that represents the organization to the rest of the world.

Hierarchy



Two- Level Hierarchy: No subnetting

- An IP address can define only 2 levels of hierarchy when not subnetted



Note

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.

Three –Levels of Hierarchy : Subnetting

Figure 19.7 *Configuration and addresses in a subnetted network*

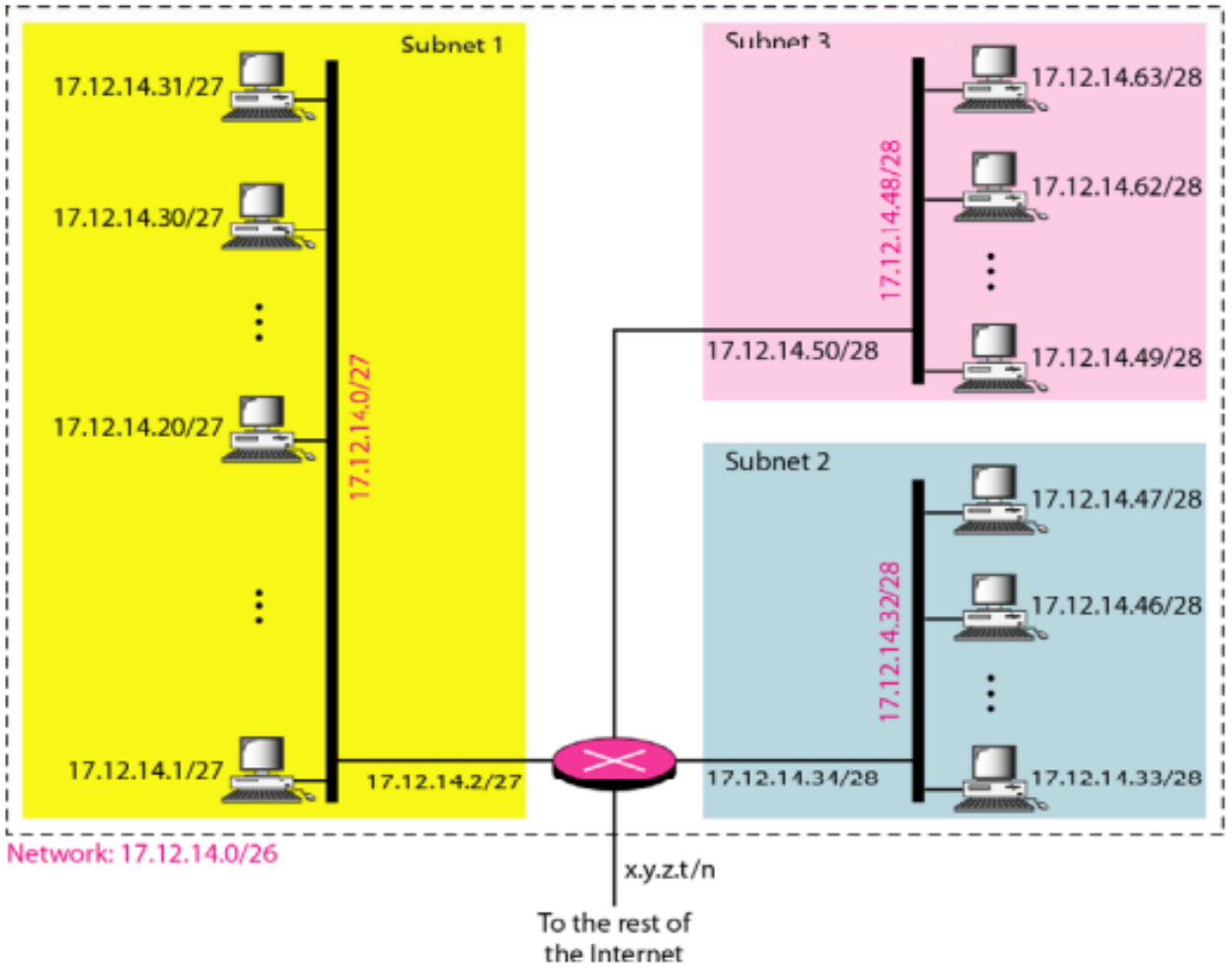
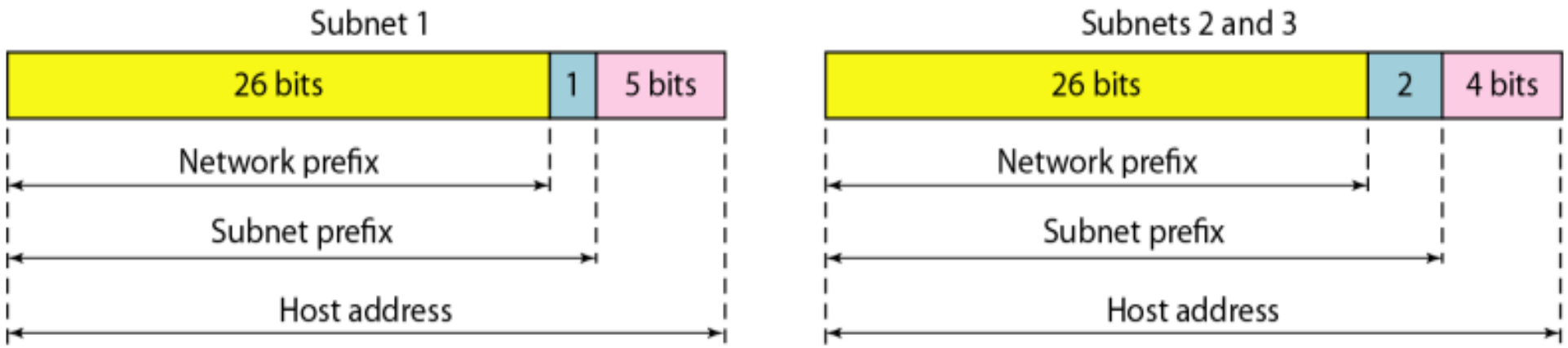


Figure 19.8 *Three-level hierarchy in an IPv4 address*



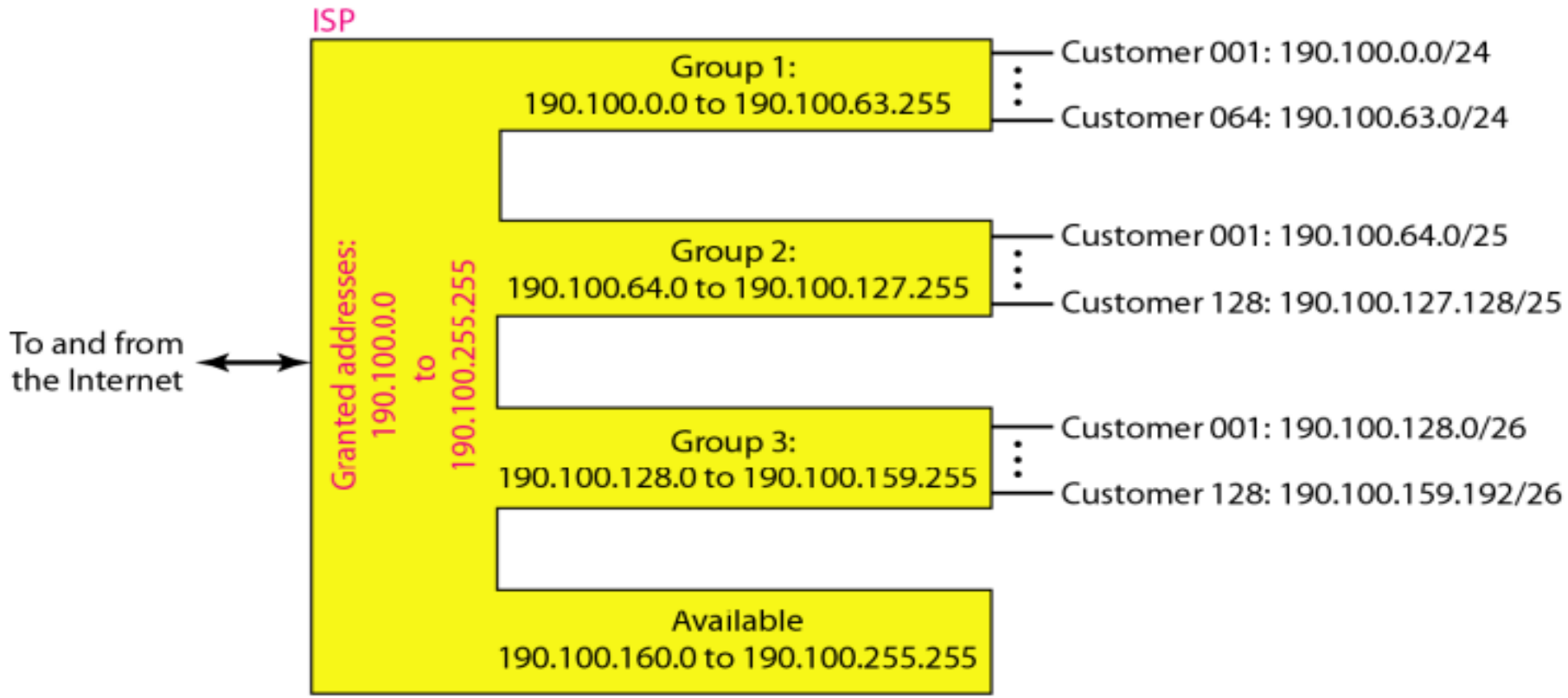
Example 19.10

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.

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



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

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 × 256 = 16,384

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		

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

Note

Switching at the network layer in the Internet uses the datagram approach to packet switching.

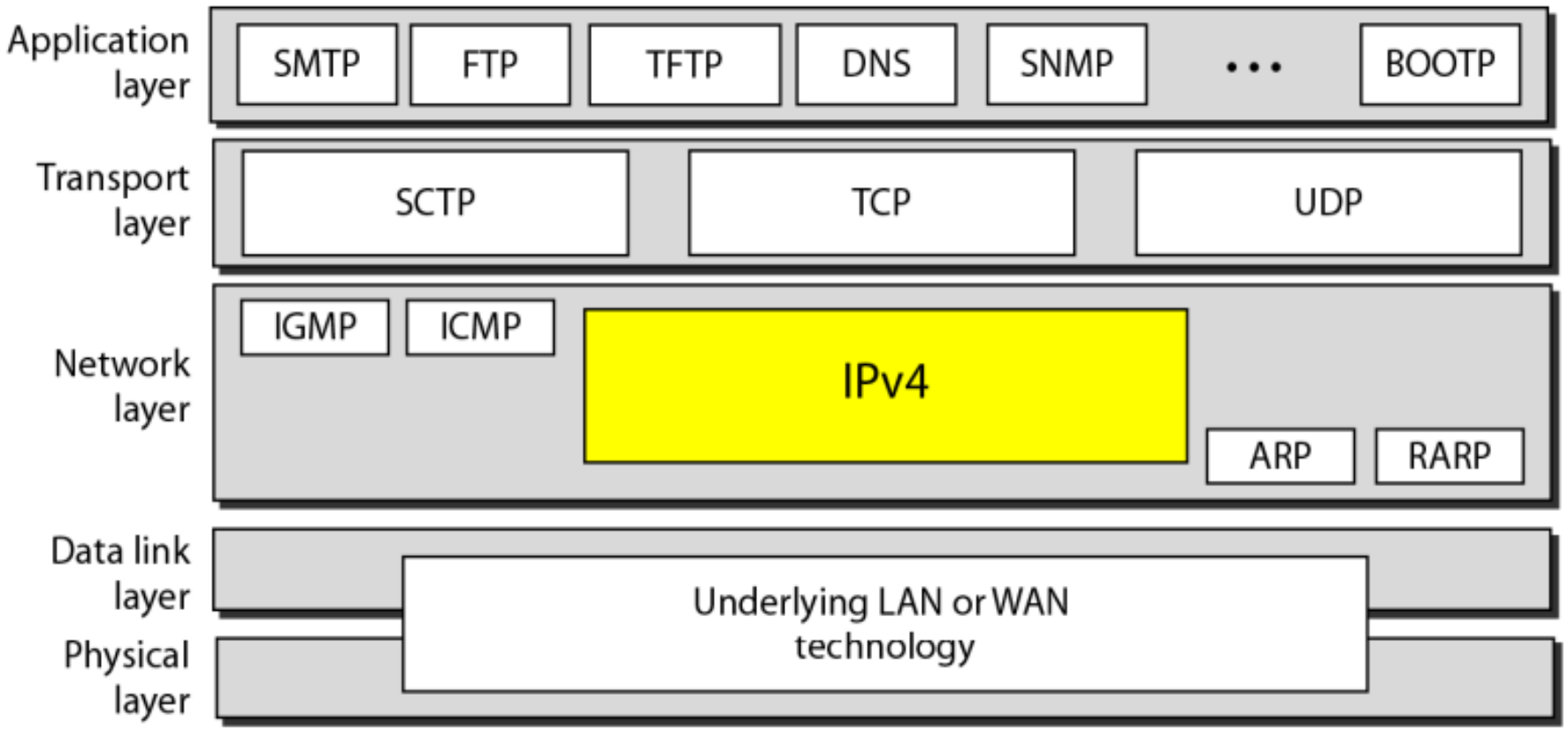
Note

Communication at the network layer in the Internet is connectionless.

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*



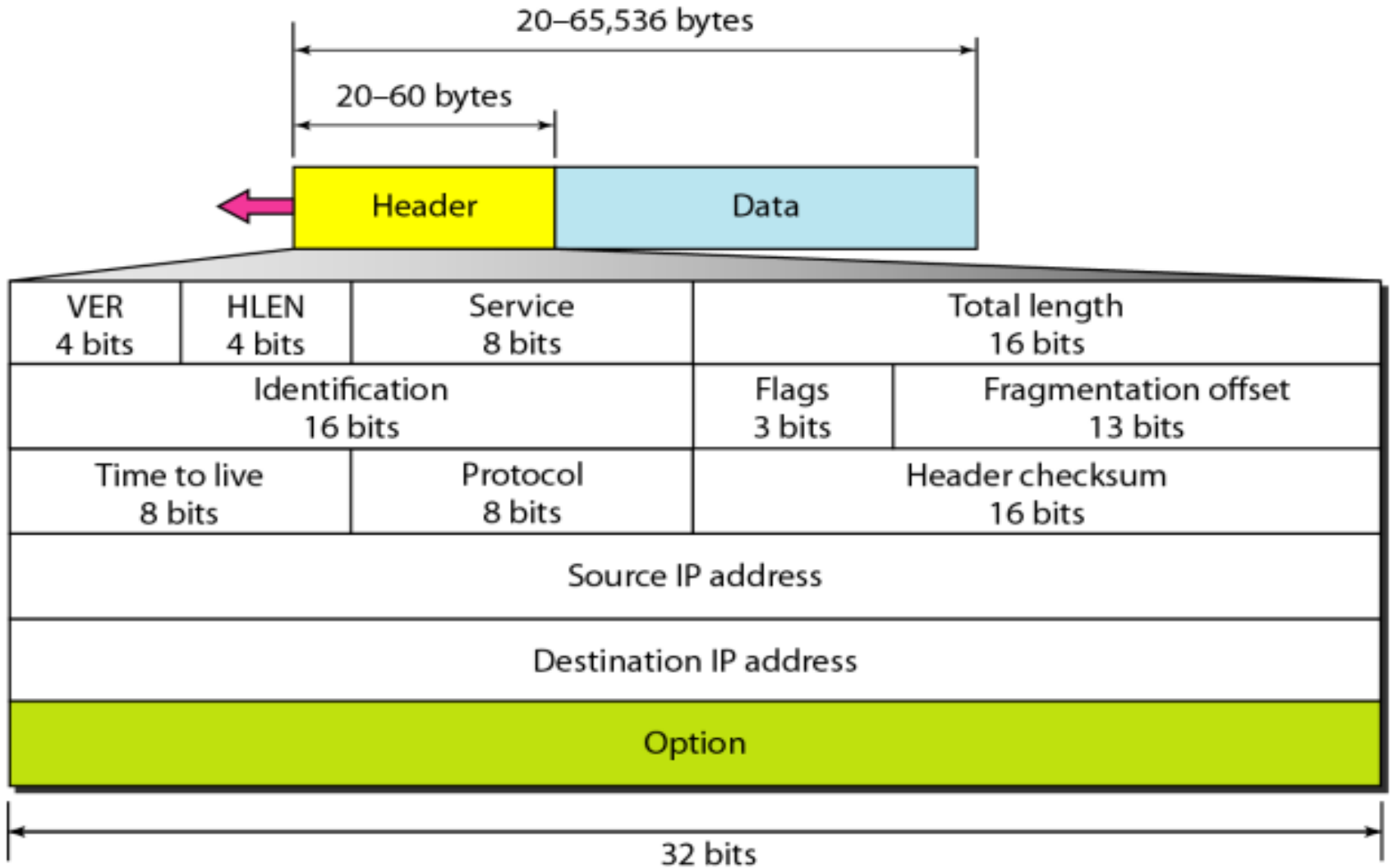
IPv4

- It is an unreliable and connectionless datagram protocol – a best effort delivery service.
- It provides no error or flow control.
- If reliability is important, IPv4 must be paired with a reliable protocol such as TCP.
- IPv4 is a connectionless protocol for a packet switching network that uses datagram approach.
- Here each datagram is handled independently and each datagram can follow a different route to the destination, thus can arrive out of order.
- Some datagrams can be lost or corrupted during transmission

Datagram

- Packets in the IPv4 layer are called datagrams.

Figure 20.5 *IPv4 datagram format*



Details

- A datagram is of variable length.
- Consists of two parts: Header + Data
- Header’s length is 20 to 60 bytes.
- Header contains information essential for routing and delivering Data.

Description of each field

1. VERSION (VER) Header Length (HLEN)

- **4 bit in length**
- Defines the version of IP (either IPv6 or IPv4)

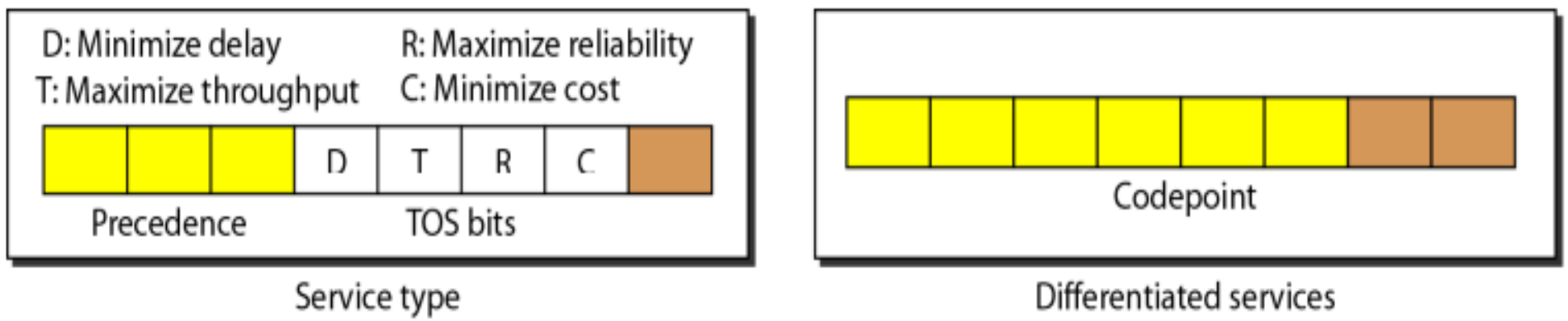
2. Header Length (HLEN)

- **4 bit in length**
- Defines the length of the header.
- Its value falls between 20 to 60 bytes

3. Service

- It was previously called as service type, now called differentiated services.

Figure 20.6 *Service type or differentiated services*



1. Service Type

- **First 3 bits are Precedence bits.**
- Next 4 bits are called Type of Service (TOS) bits,
- The last bit is not used.

A. Precedence:

- Value ranges from 000 to 111.
- Defines priority of the datagram in issues.
- Used in situations of Network Congestion - Router discards datagrams of low precedence in case of congestion.

B. TOS bits

- **4 bit in length**
- Out of 4 only a single bit can be 1 at a time, thus we have 5 different types of services.
- Bit patterns and their interpretations are shown below.

Table 20.1 ***Types of service***

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

- Application programs can request a specific type of service.

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

Note

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

2. Differentiated Services

- IETF has changed the name and interpretation.

- Here, the first 6 bits makeup a codepoint subfield and the last 2 bits are not used.

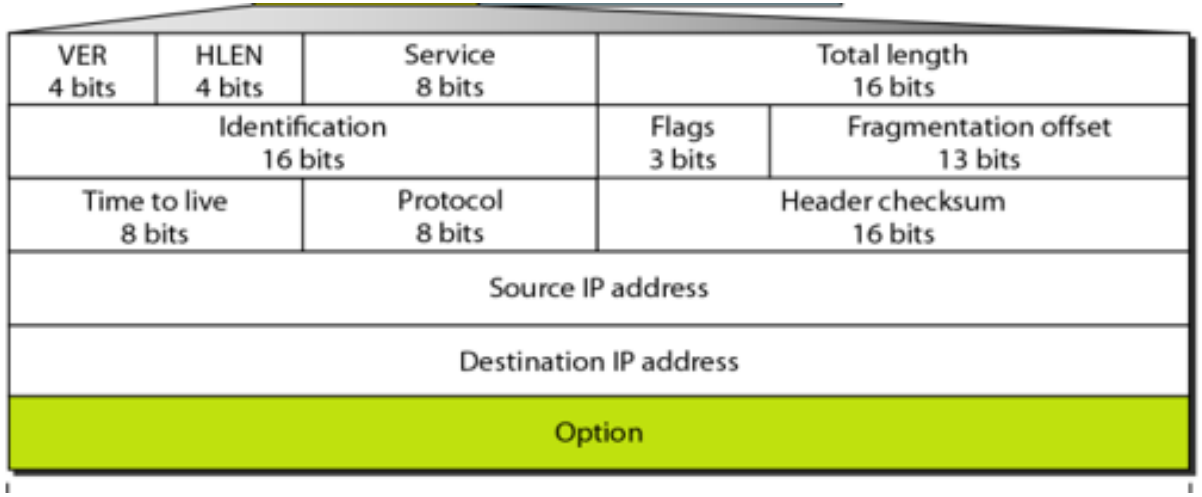
Category	Codepoint	Assigning Authority
1	XXXXX0	Internet
2	XXXX10	Local
3	XXXX01	Temporary / Experimental

4. Total Length

- It is a 16-bit field.
- This field defines the total length of the Datagram (header + Data)
- Length of data =total length –header length
- Total length of IPv4 datagram is limited to 65535 bytes.

Note

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



- Identification, Flags, Fragmentation offset -They are used in fragmentation
- Time to Live : It holds the timestamp, which gets decremented by each visited router. The datagram is discarded when the value becomes zero. It intentionally limits the journey of the packet.
- Protocol: This 8 bit field defines the higher level protocol (TCP, UDP, ICMP, IGMP)that uses the services of IPv4 layer. The field specifies the final destination protocol to which IPv4 datagram is delivered.

Figure 20.8 *Protocol field and encapsulated data*

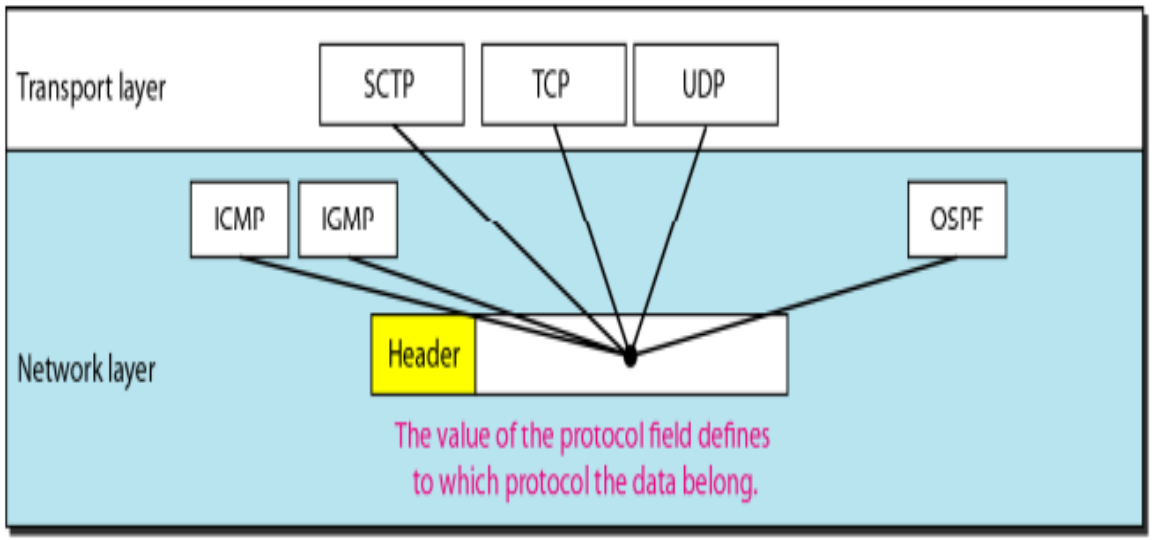


Table 20.4 *Protocol values*

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

- Checksum : It is used for error detection.
- Source Address: 32- bit field. It remains unchanged during the time IPv4 datagram travels from the source host to the destination host.
- Destination Address: 32- bit field. It remains unchanged during the time IPv4 datagram travels from the source host to the destination host.

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.

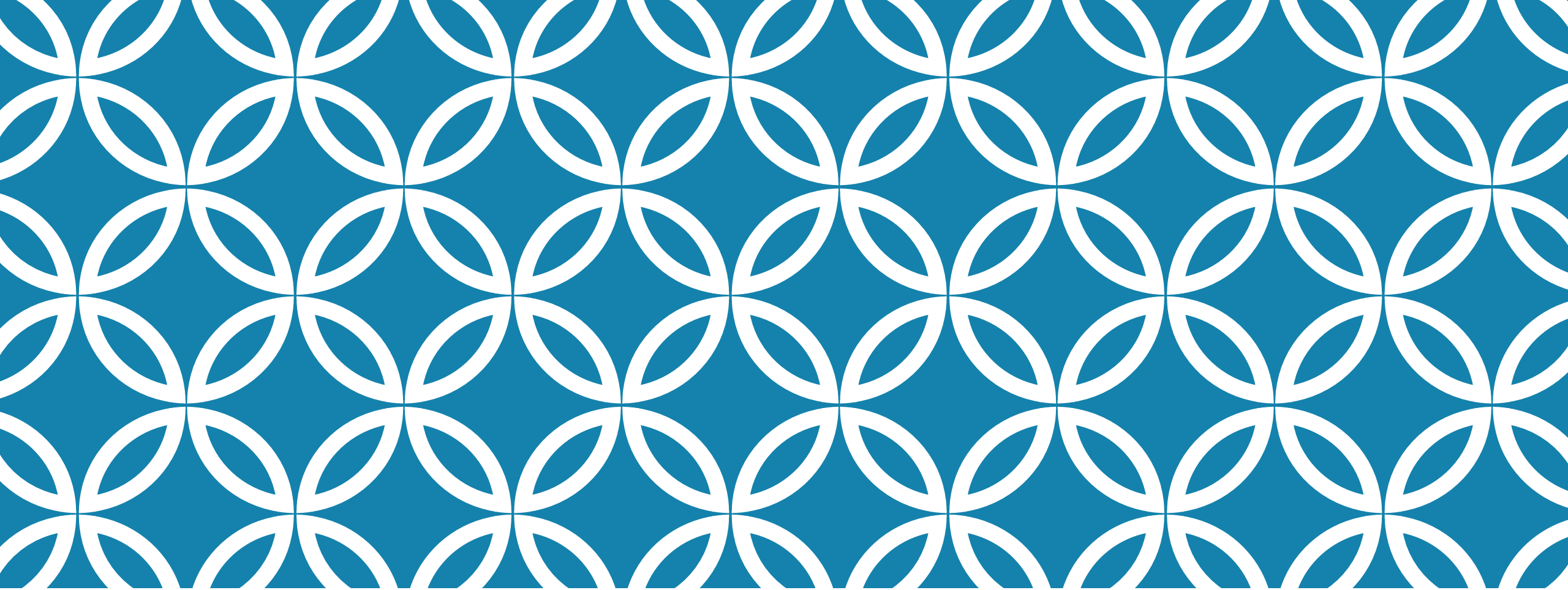
0x45000028000100000102 . . .

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.

FRAGMENTATION



FRAGMENTATION & DISTANCE VECTOR ROUTING

FRAGMENTATION

- A datagram can travel through different networks.
- Each router decapsulates the IPv4 datagram from the frame it receives, processes it and then encapsulates into another frame
- If a router connects a LAN to WAN , it receives a frame in LAN format and sends the fame in WAN format

MAXIMUM TRANSFER UNIT (MTU)

- Maximum size of the data field is mentioned in one of the fields of a frame.

- When a datagram is encapsulated in a frame , the total size of the datagram must be less than this maximum size, which is defined by the restrictions imposed by HW/SW used in the network.

Figure 20.9 *Maximum transfer unit (MTU)*

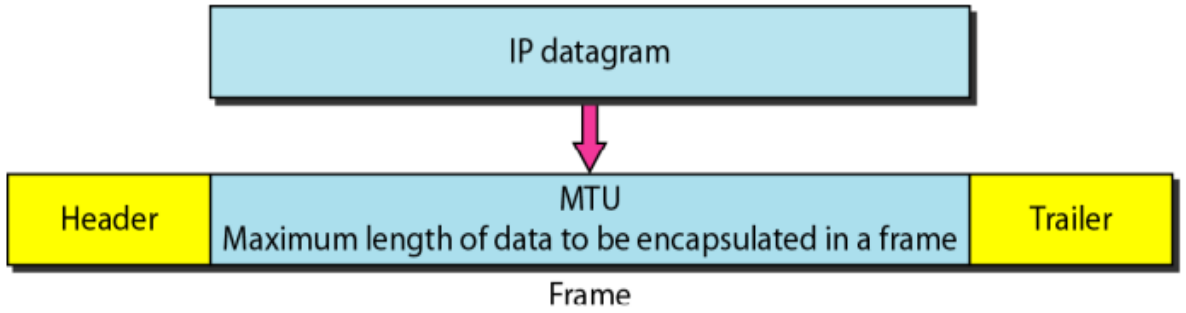


Table 20.5 ***MTUs for some networks***

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

FRAGMENTATION

- To make IPv4 protocol independent of the physical network, the designers decided to make the maximum length of the IPv4 datagram equal to 65,535 bytes.
- This makes transmission more efficient.
- However for other physical networks, we must divide the datagram to make it possible to pass through these networks. This is called as fragmentation.
- When a datagram is fragmented, each fragment has its own header with most of the fields repeated.
- A fragmented datagram may itself be fragmented if it encounters network with even smaller MTU.
- The reassembly of the datagram is done at the final destination.

FIELDS RELATED TO FRAGMENTATION

- IDENTIFICATION:
  - 16- bit field
  - It identifies a datagram originating from the source host.
  - The combination of Identification and source IPv4 address must uniquely define the datagram as it leaves the source host.
  - It holds a counter number to make the address unique.
  - When datagram is fragmented, it copies the identification field to all fragments
  - This even helps the destination to reassemble all fragments with same Identification field.

Figure 20.10 *Flags used in fragmentation*

- FLAGS:
  - ✓ 3 bit field
  - ✓ First bit is reserved

  - ✓ Second bit is called Do Not fragment bit.
    - ✓ If 1 – the machine must not fragment the datagram
    - ✓ If it cannot pass the datagram through the available physical link, then it discard the packet and sends an ICMP(Internet Control Message Protocol) error message to the source host.
    - ✓ If 0 – the datagram can be fragmented if necessary
  - ✓ Third bit is called the more fragment bit
    - ✓ If 1 – the datagram is not the last fragment and there are more fragments after this one.
    - ✓ If 0 – this is the last or only fragment.

❑ FRAGMENTATION OFFSET:

- ✓ 13 Bit field
- ✓ It shows the relative position of this fragment with respect to the whole datagram
- ✓ It is the offset of the data in the original datagram measured in units of 8 bytes.
- ✓ Length of offset field is 13 bits and can represent a max of 8191 bytes

Figure 20.11 *Fragmentation example*

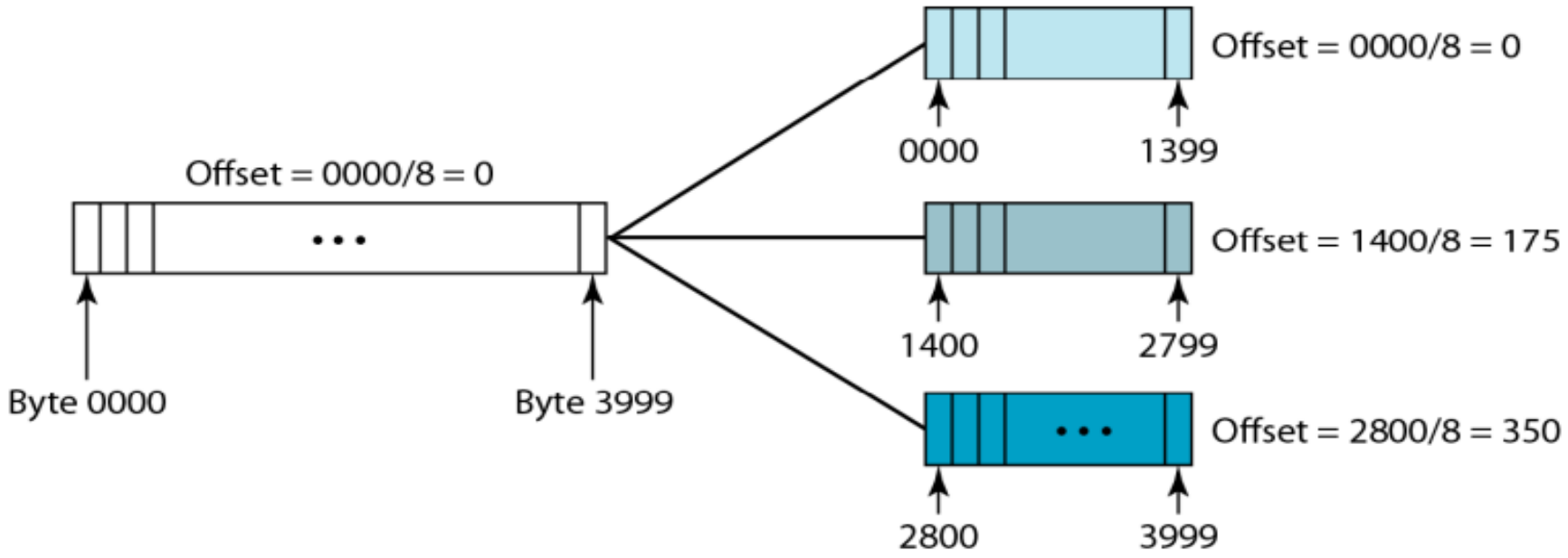
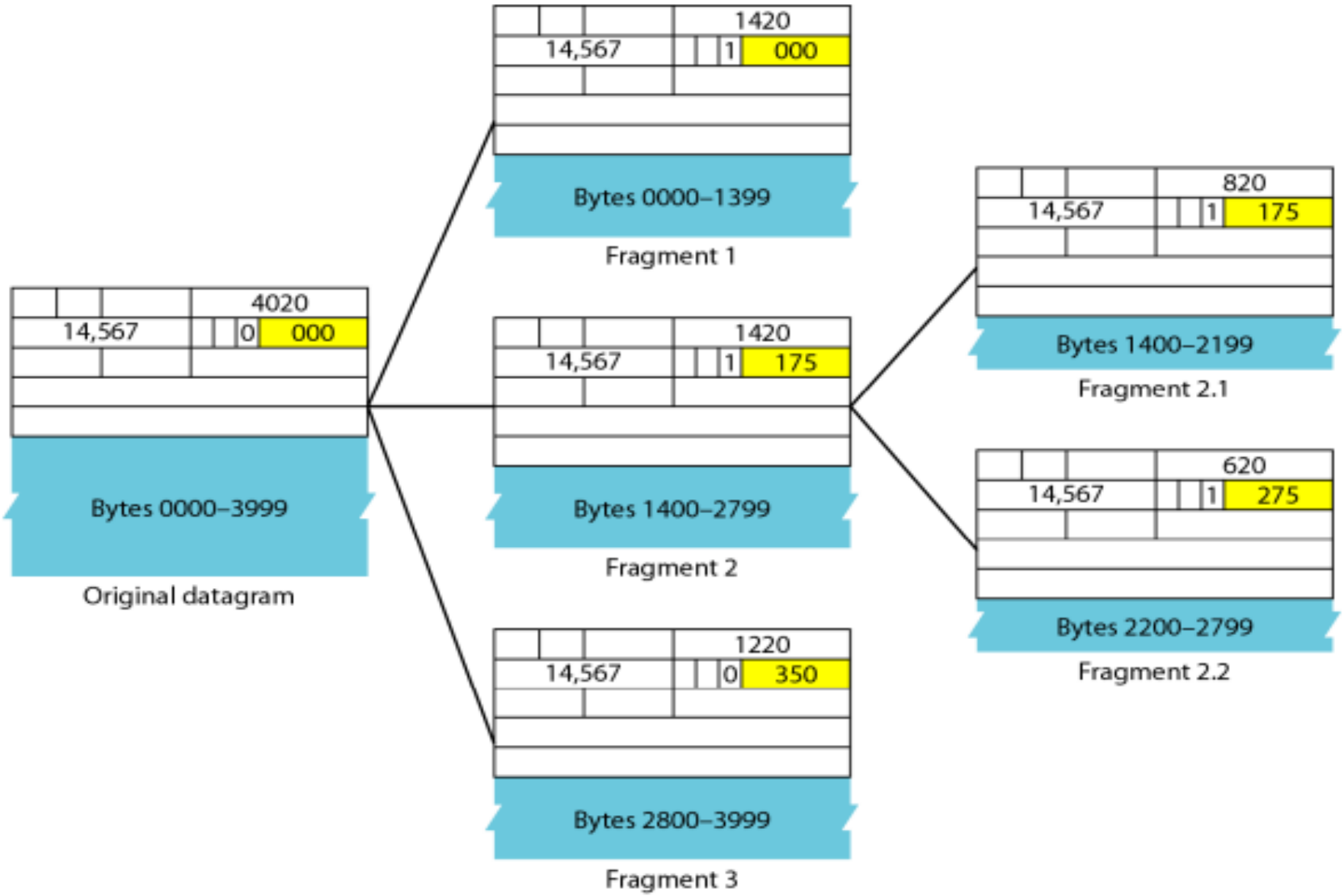


Figure 20.12 *Detailed fragmentation example*



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?

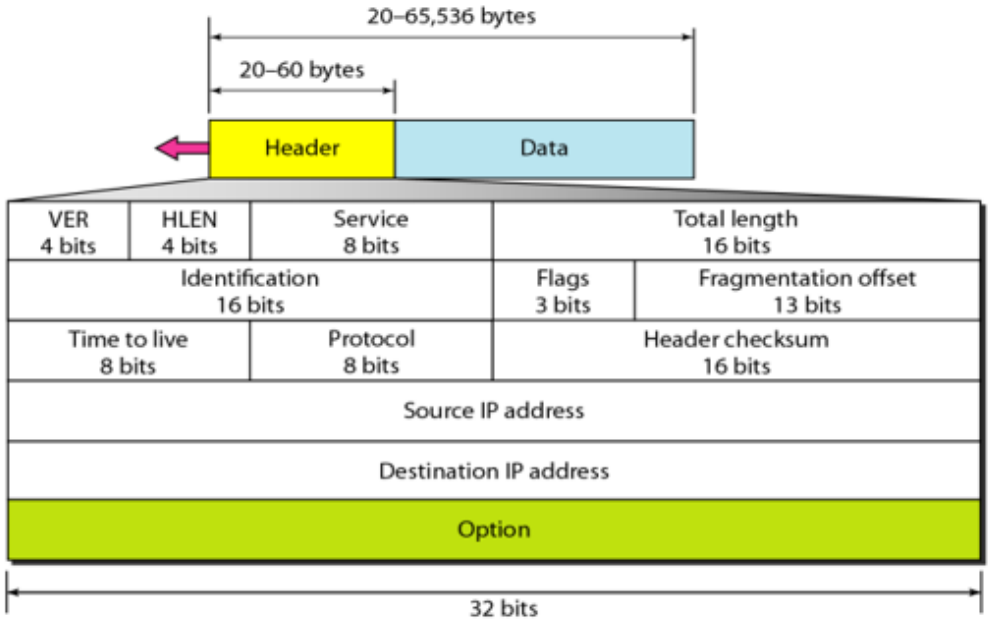
Solution

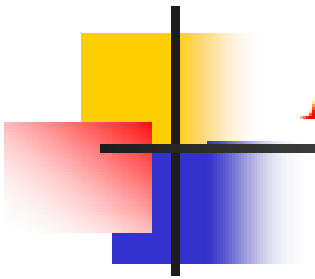
The first byte number is 100 × 8 = 800. The total length is 100 bytes, and the header length is 20 bytes (5 × 4), which means that there are 80 bytes in this datagram. If the first byte number is 800, the last byte number must be 879.

CHECKSUM

- First the value of the checksum field is set to 0
- Then the entire header is divided into 16 bit sections and added together.
- The result is complemented and inserted into checksum field

Figure 20.5 *IPv4 datagram format*

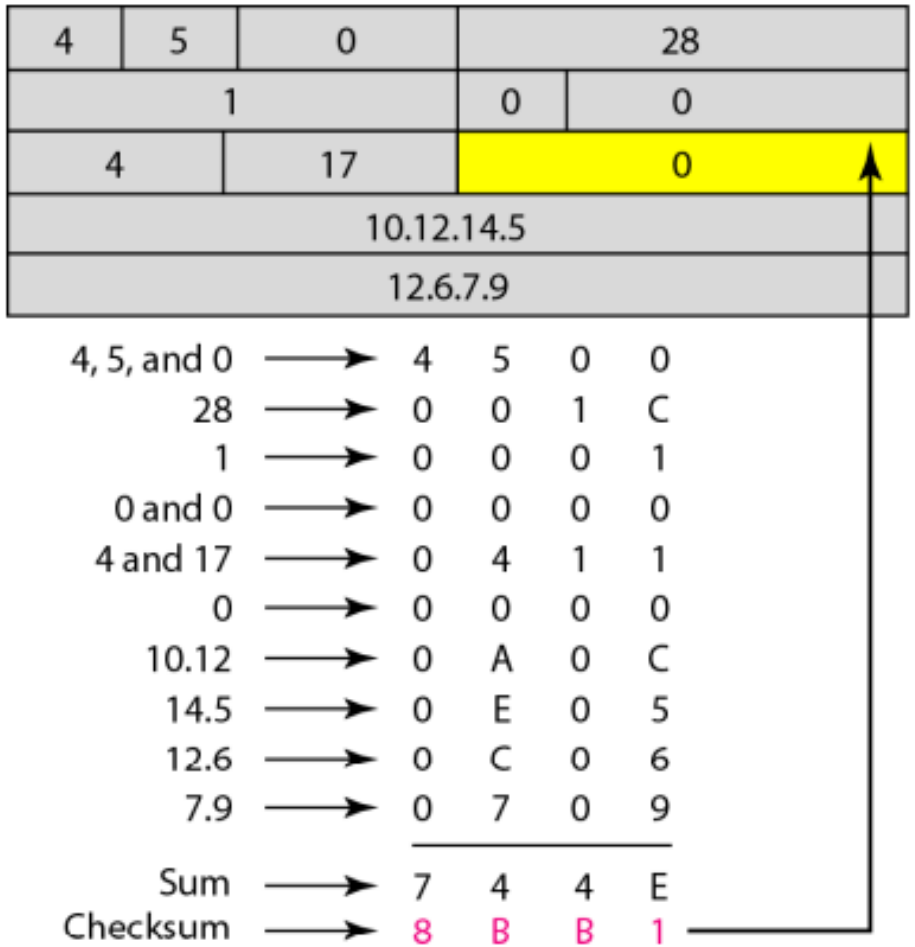




Example 20.10

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.

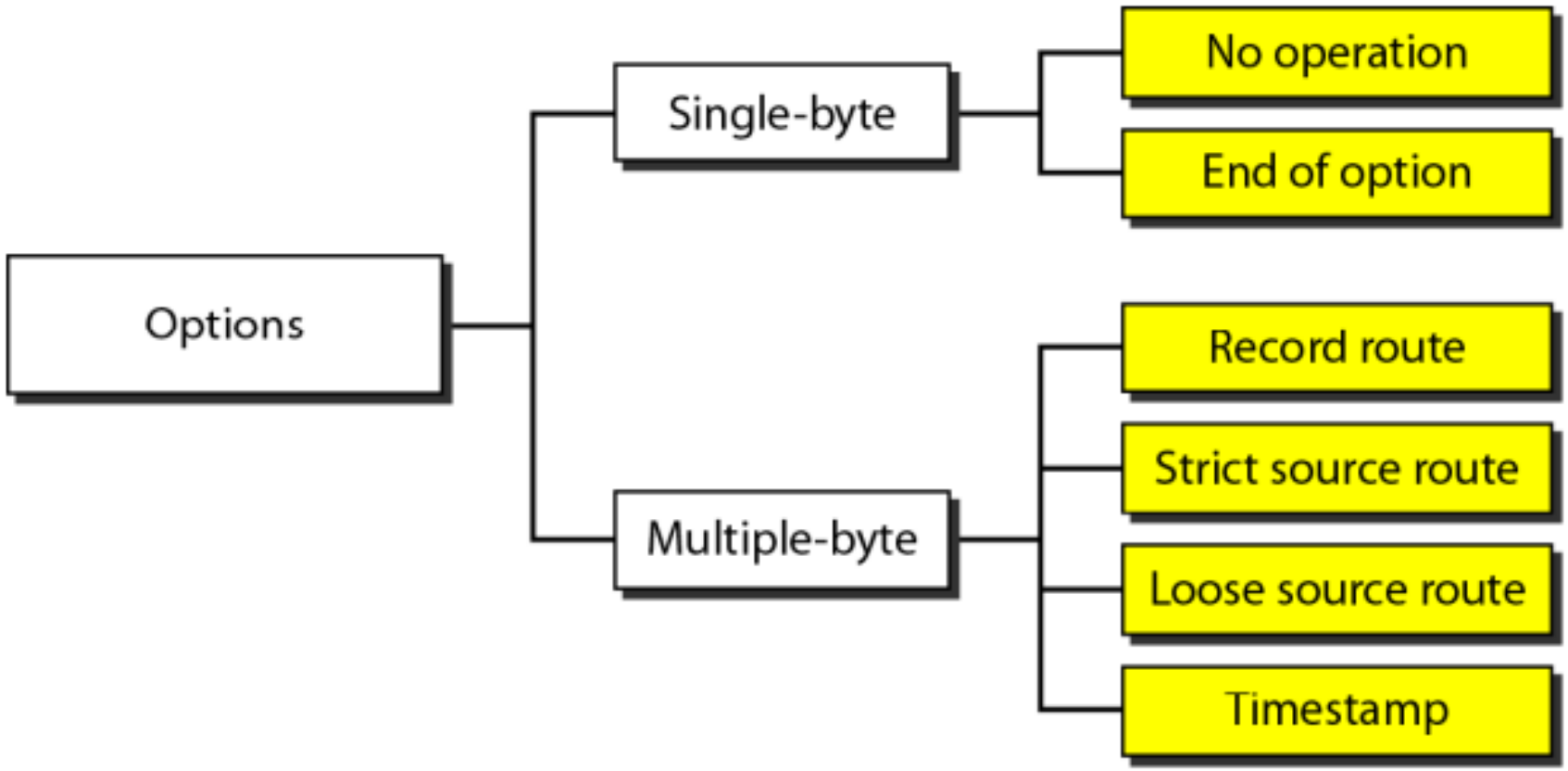
Figure 20.13 *Example of checksum calculation in IPv4*



OPTIONS

- The header of IPv4 datagram is made of 2 parts: a fixed part and a variable part
- Fixed part : 20 bytes long
- Variable part : max of 40 bytes long
- Options are used for network testing and debugging

Figure 20.14 *Taxonomy of options in IPv4*



DETAILS:

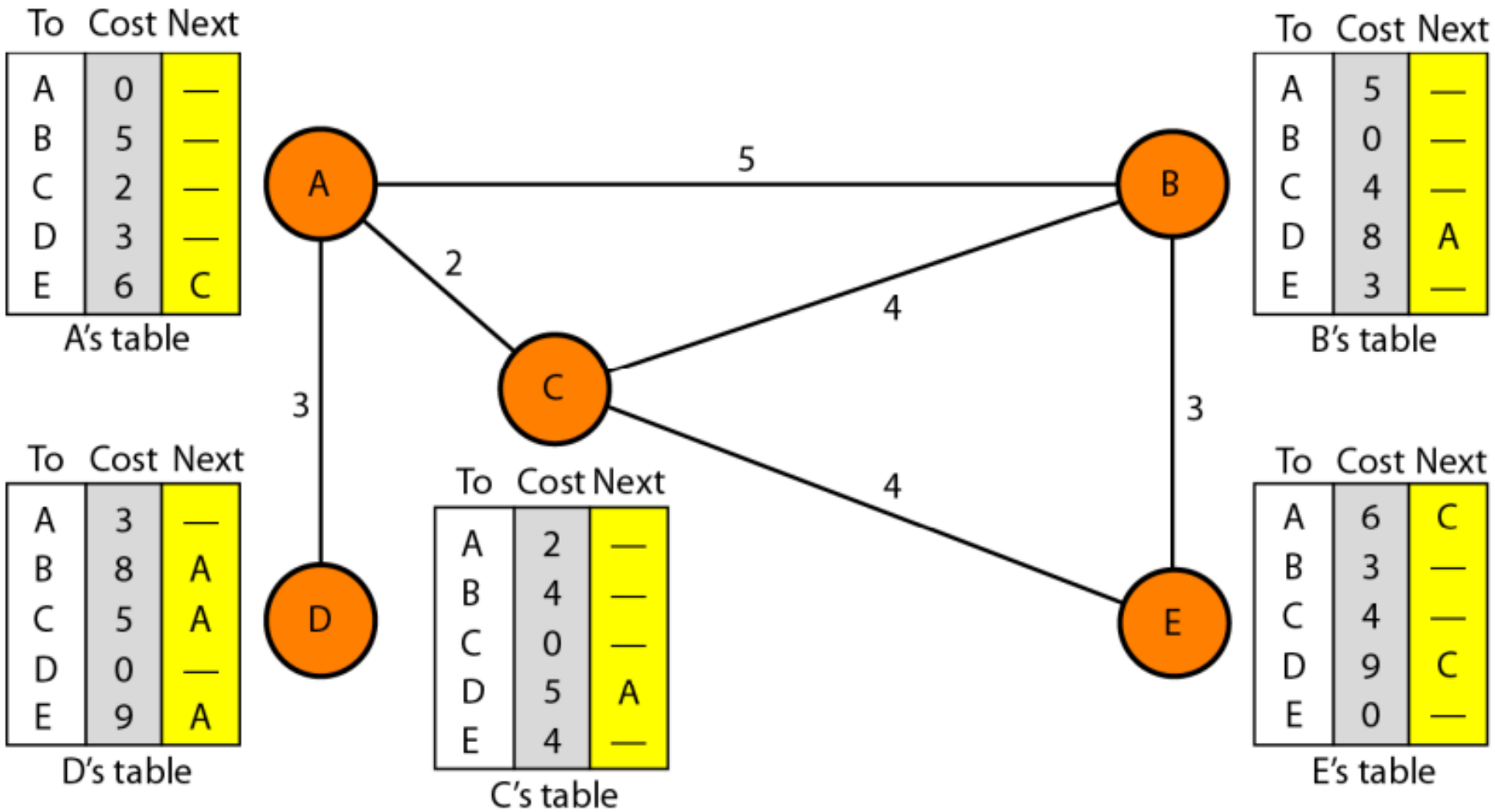
- No operation : 1 byte option used as a filler between options
- End of option: 1 byte option used for padding at the end of option field
- Record Route : To record the internet routers the internet outers that can handle the datagram. It can list up to 9 routers addresses. It is used for debugging and management purposes.
- Strict Source Route: It is used by the source to predetermine the route for the datagrams as it travels through the internet.
- Loose Source Route: It is less rigid. Each router in the list must be visited, but the datagram can visit other routers as well.
- Timestamp: It is used to record the time of datagram processing by a router. It is expresses in milliseconds. It helps to estimate the time it takes for a datagram to go from one router to another.

ROUTING PROTOCOLS

DISTANCE VECTOR ROUTING

- The least cost route between any two nodes is the route with minimum distance
- Here, each route maintains a vector (table) of minimum distances to every node.
- The table at each node guides the packets to the desired node by showing the next stop in the route.

Figure 22.14 *Distance vector routing tables*

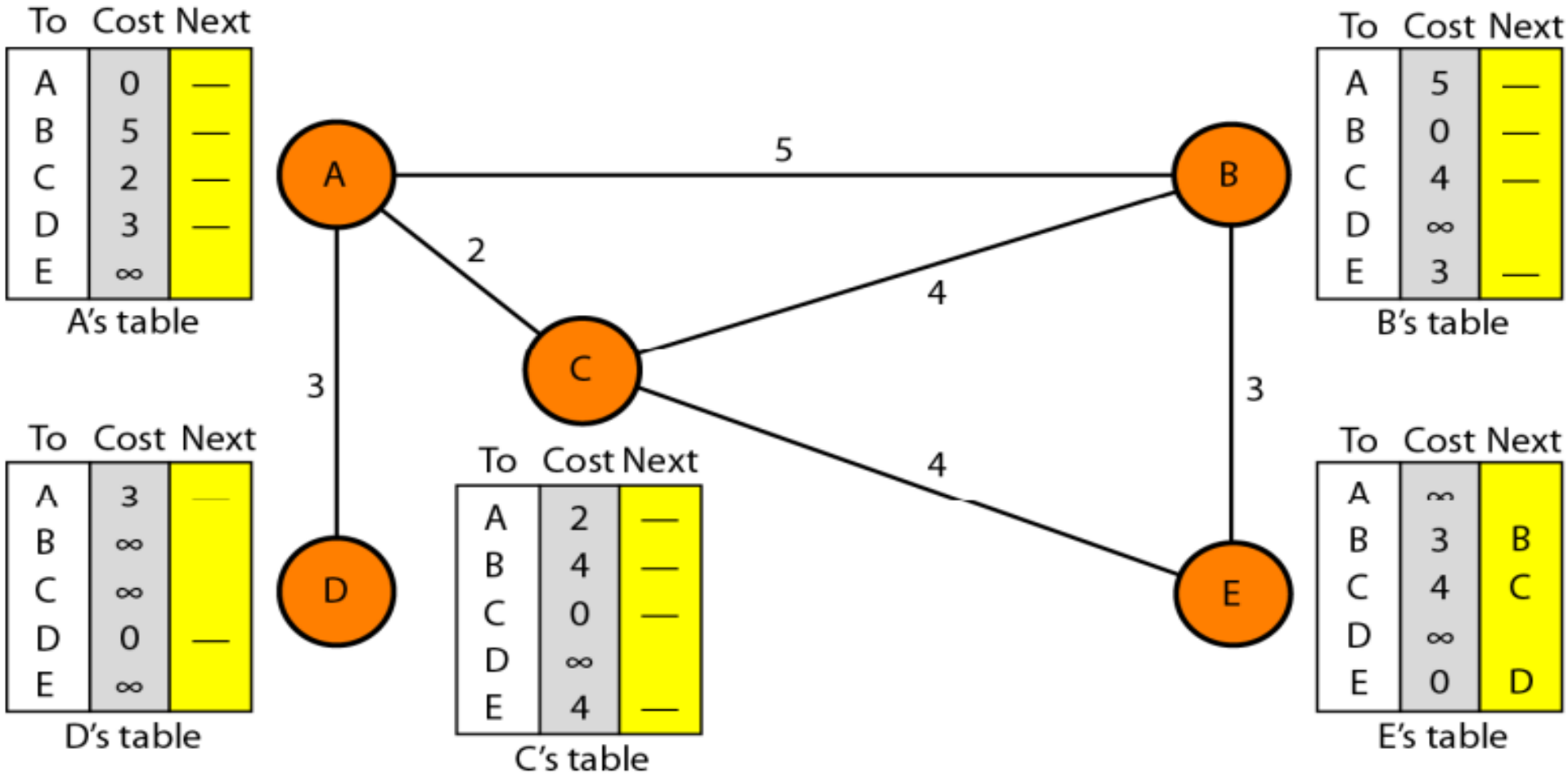


DISTANCE VECTOR ROUTING

Steps:

1. Initialization
2. Sharing ---- But how much ? When to share ? – Periodic or Triggered
3. Updating : 3 steps

Figure 22.15 *Initialization of tables in distance vector routing*



Note

In distance vector routing, each node shares its routing table with its immediate neighbors periodically and when there is a change.

UPDATING : 3 STEPS

1. the receiving node needs to add the cost between itself and the sending node to each value in the second column.

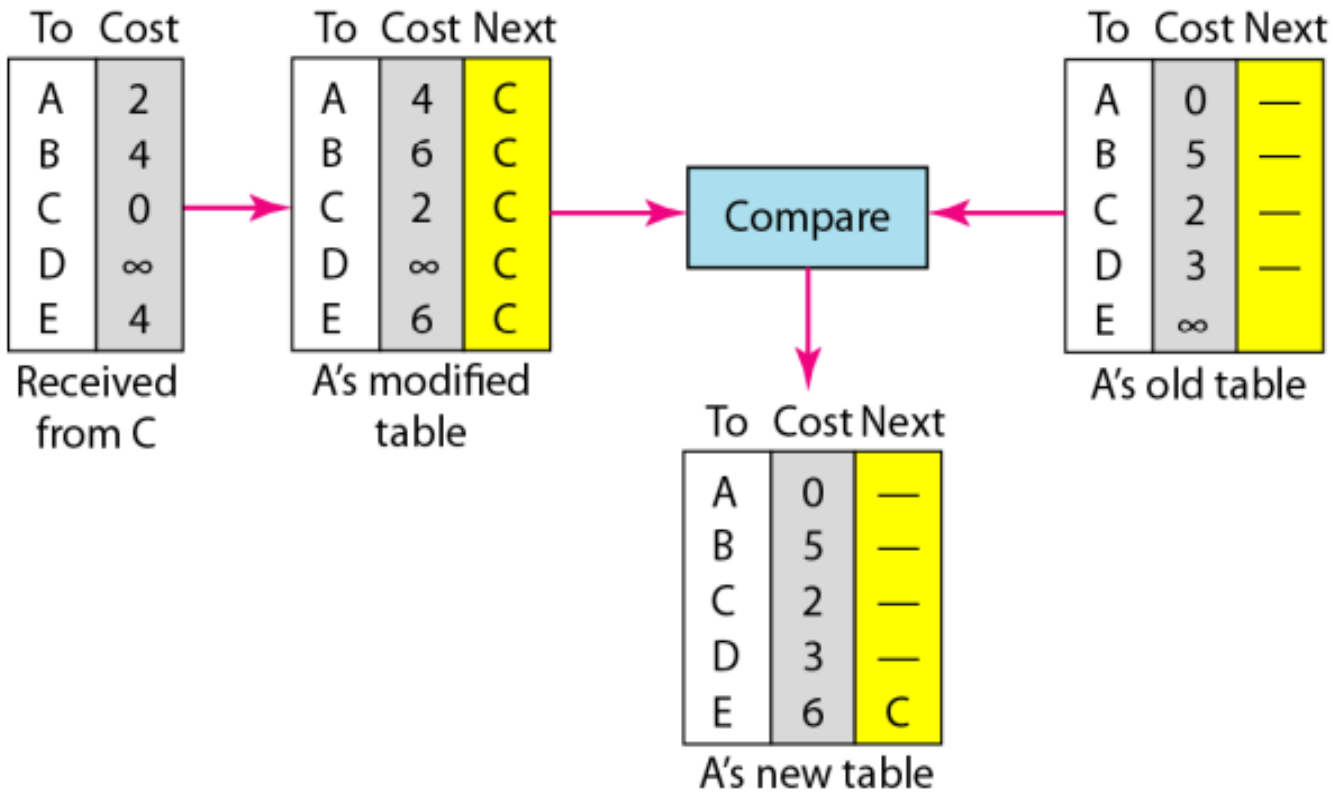
2. the receiving node needs to add the name of the sending node to each row as the third column if receiving node uses any info from that row.

3. the receiving node needs to compare each row of its old table with the modified version

A. If the next node entry is different, the receiving node chooses the row with smaller cost. If there is a tie, then old one is kept.

B. If the next-node entry is same, the receiving node chooses the new row

Figure 22.16 *Updating in distance vector routing*



Module 3

IPV6, Transition from IPV4 to IPv6, Link State Protocol

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.

Figure 20.15 *IPv6 datagram header and payload*

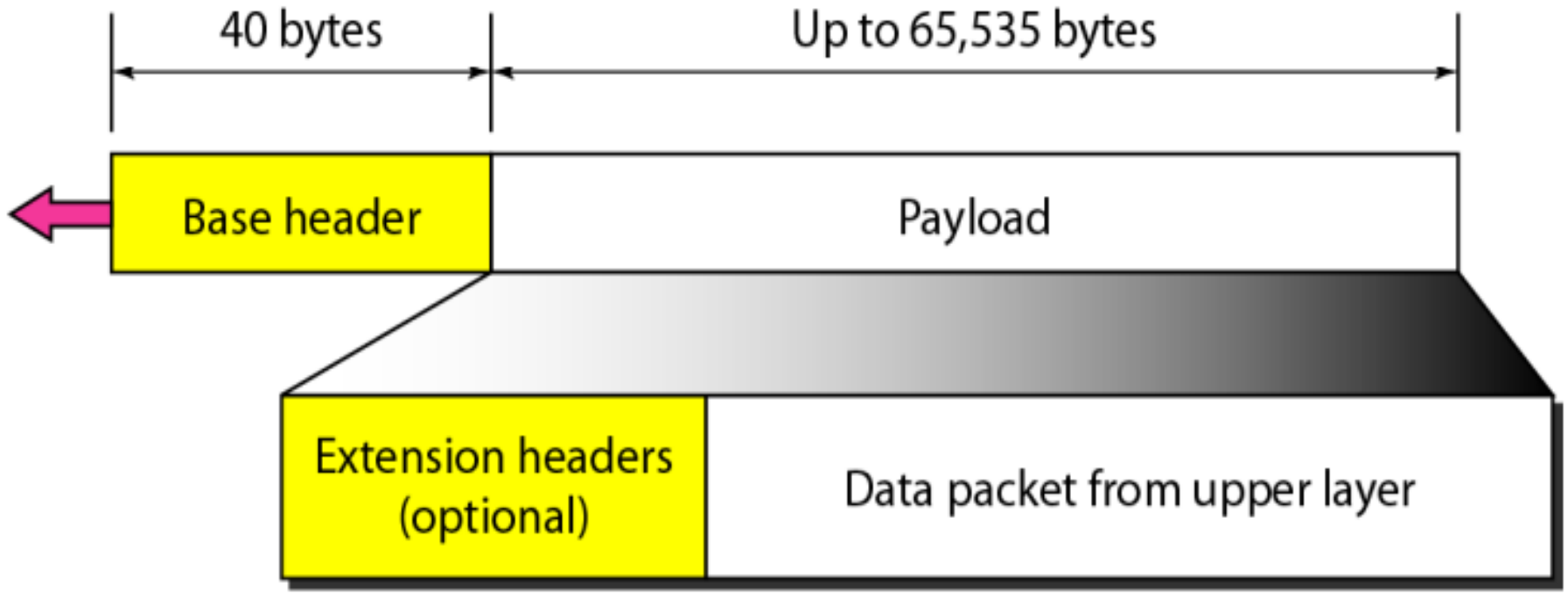


Figure 20.16 *Format of an IPv6 datagram*

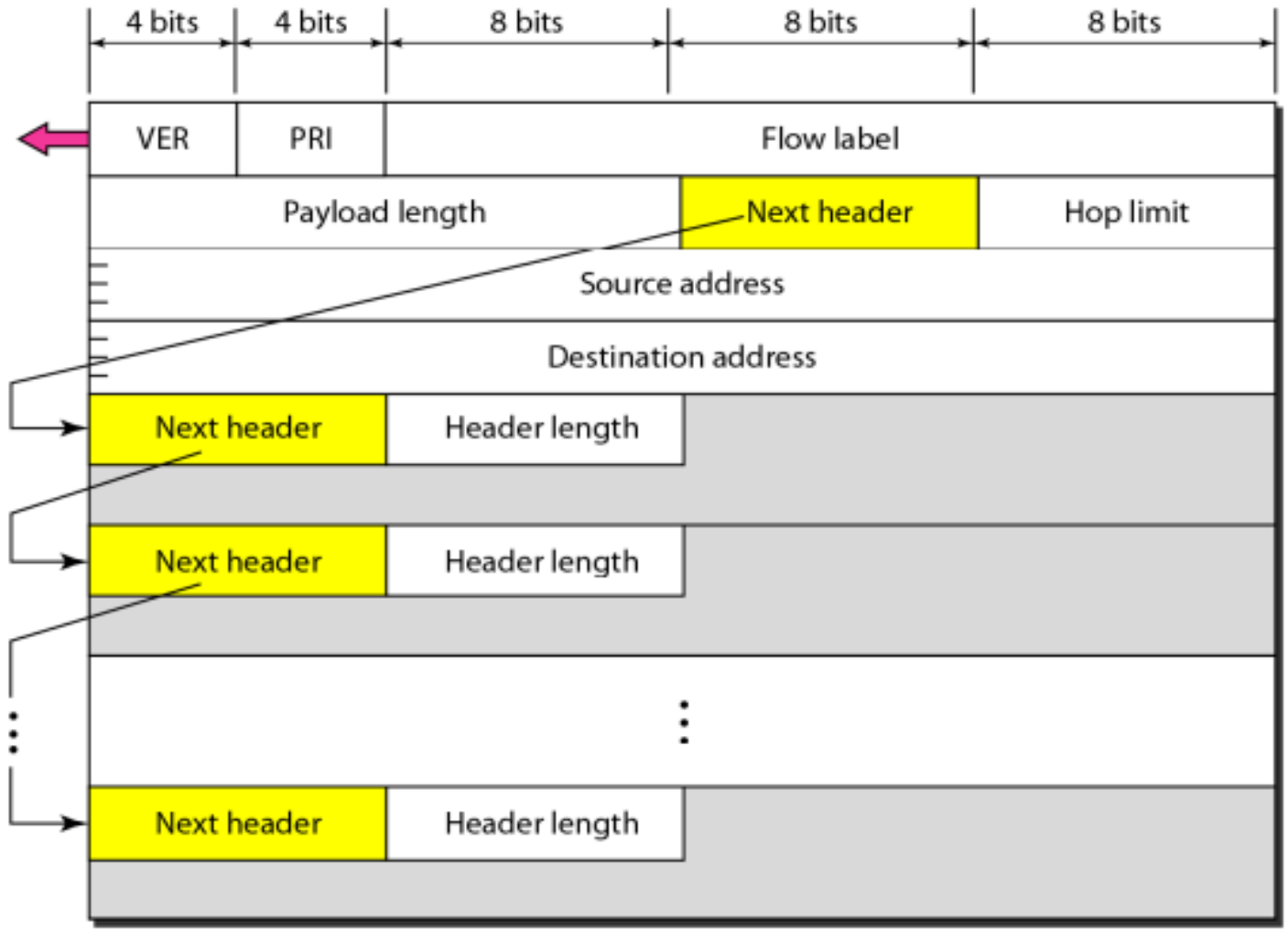


Figure 20.5 *IPv4 datagram format*

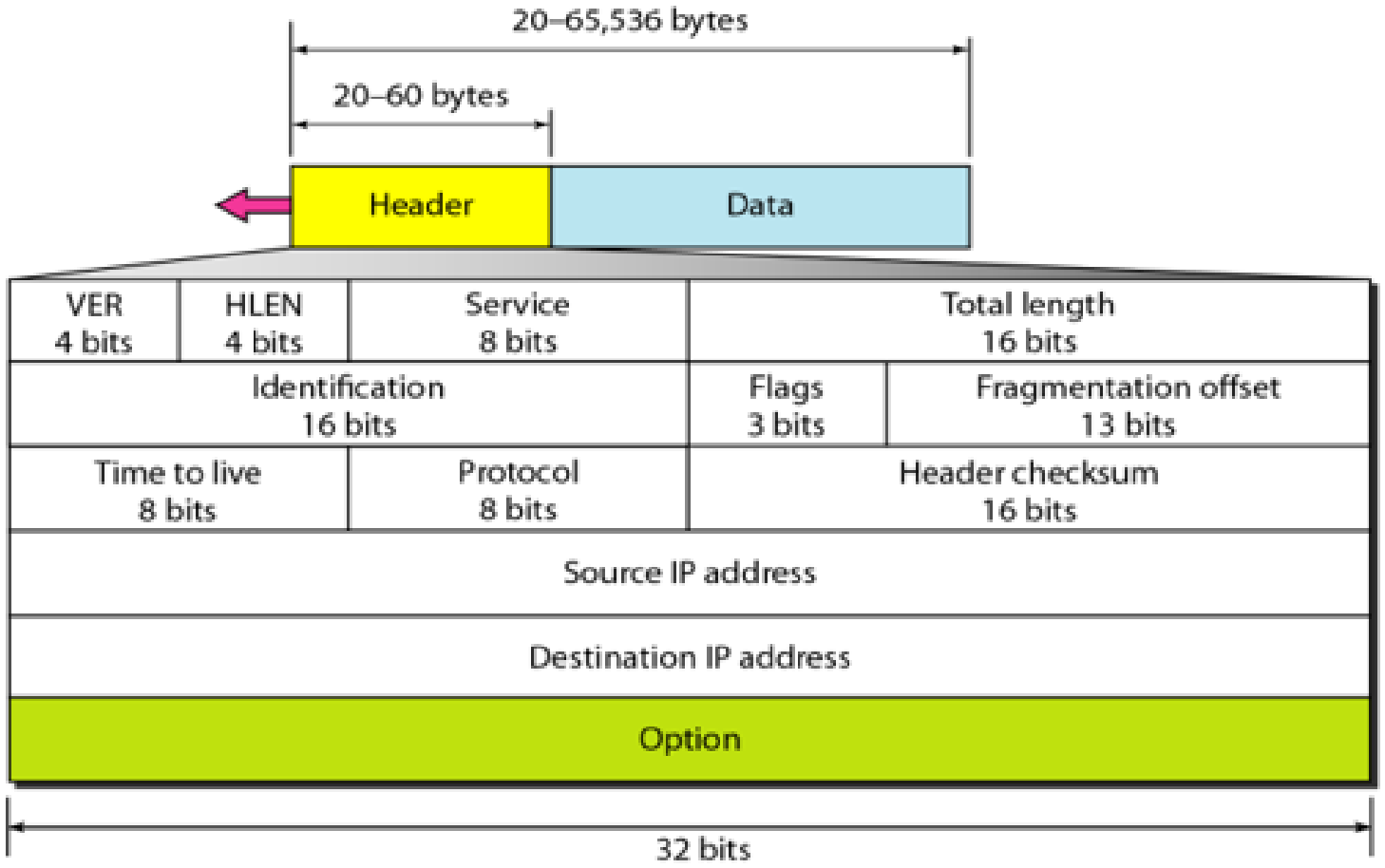


Table 20.6 ***Next header codes for IPv6***

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.

20-4 TRANSITION FROM IPv4 TO IPv6

Because of the huge number of systems on the Internet, the transition from IPv4 to IPv6 cannot happen suddenly. It takes a considerable amount of time before every system in the Internet can move from IPv4 to IPv6. The transition must be smooth to prevent any problems between IPv4 and IPv6 systems.

Figure 20.18 *Three transition strategies*

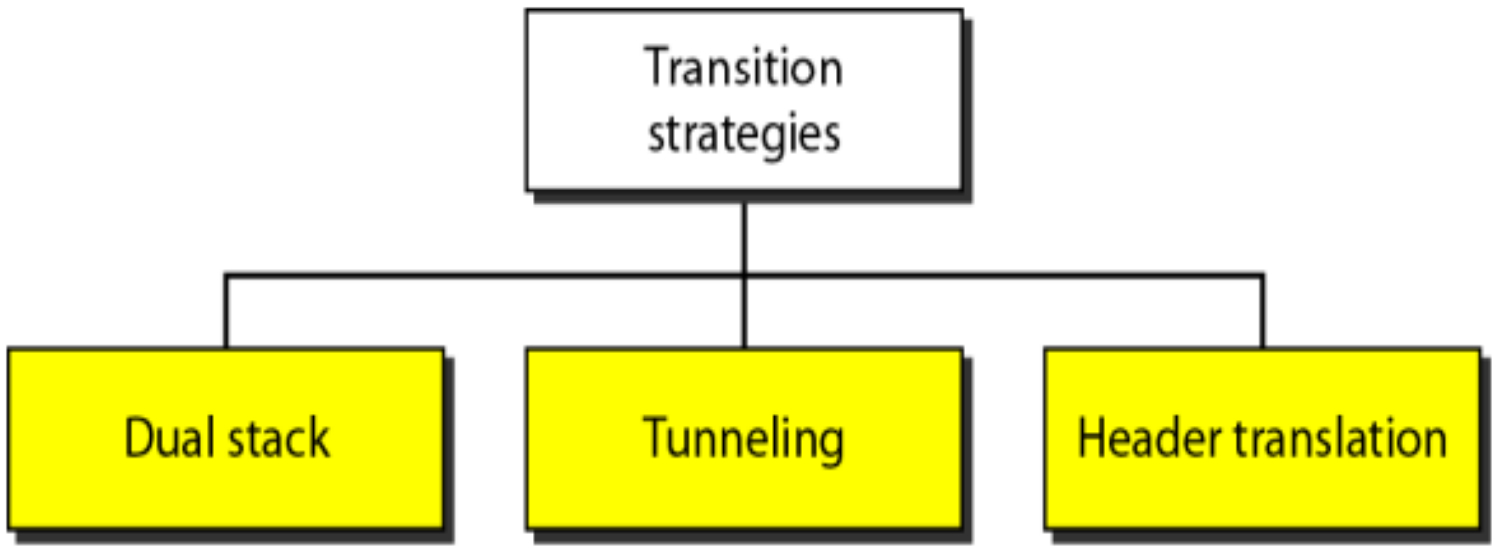


Figure 20.19 *Dual stack*

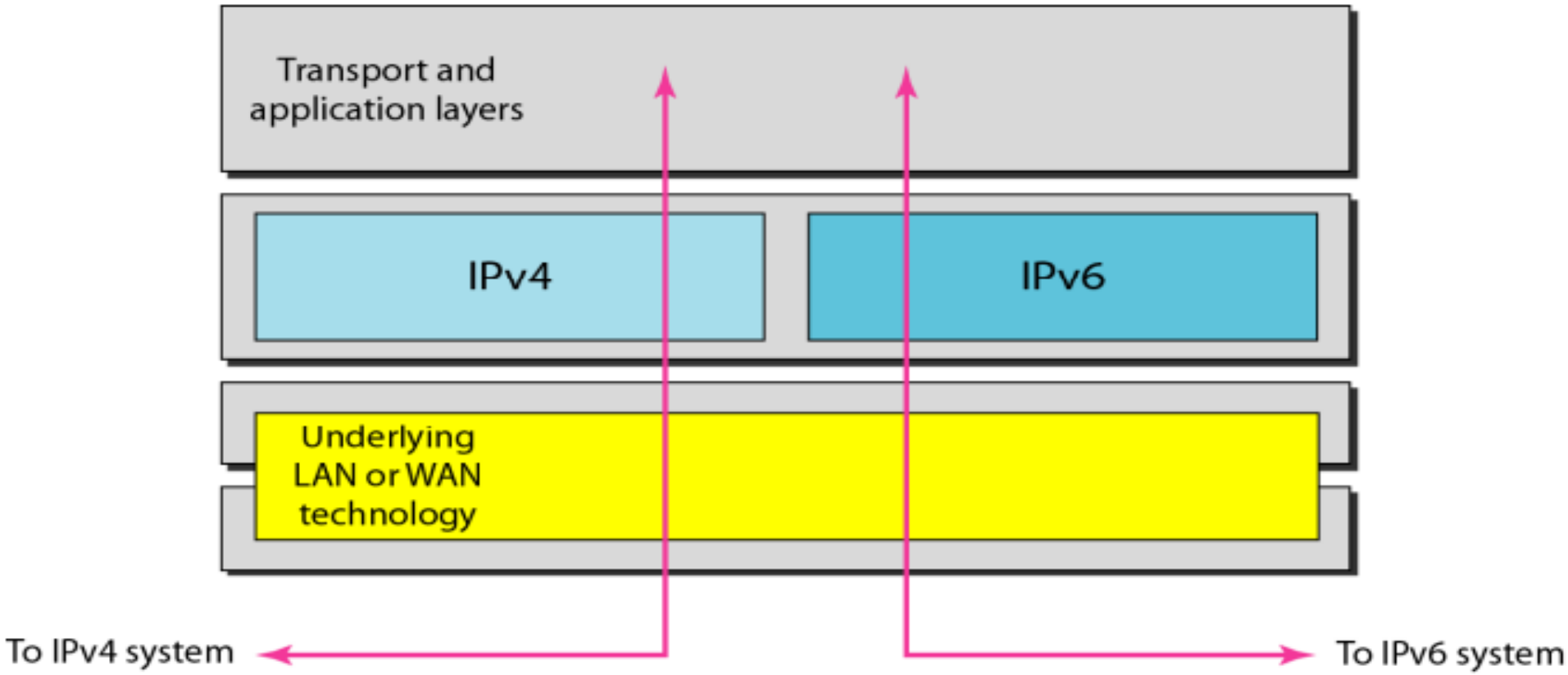


Figure 20.20 *Tunneling strategy*

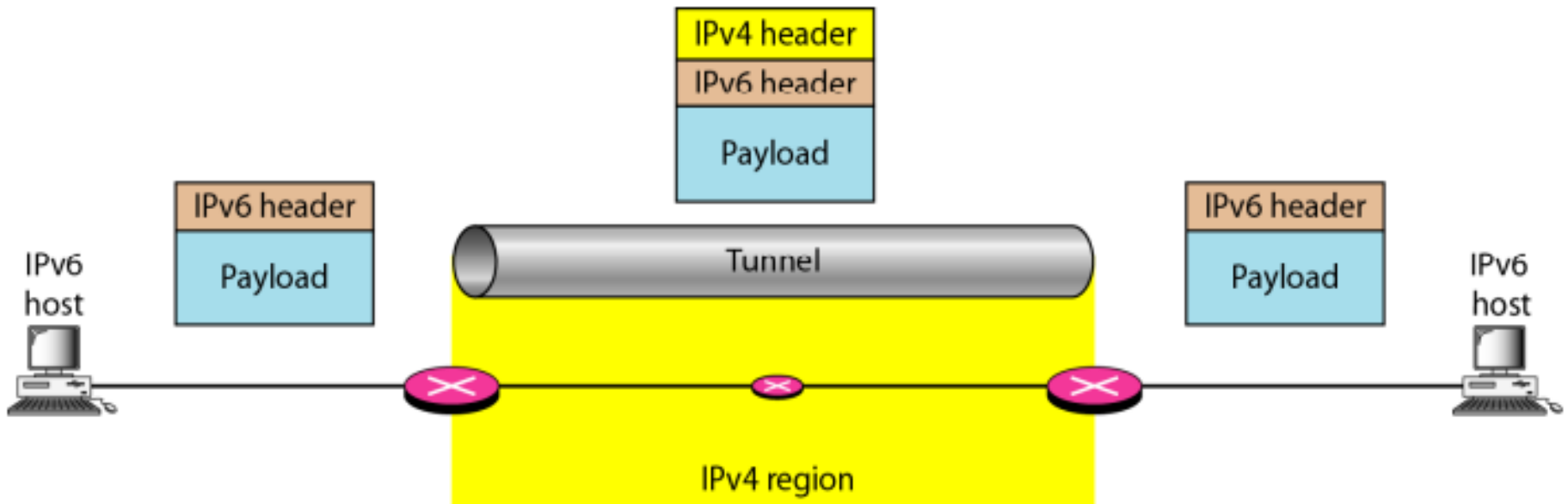


Figure 20.21 *Header translation strategy*

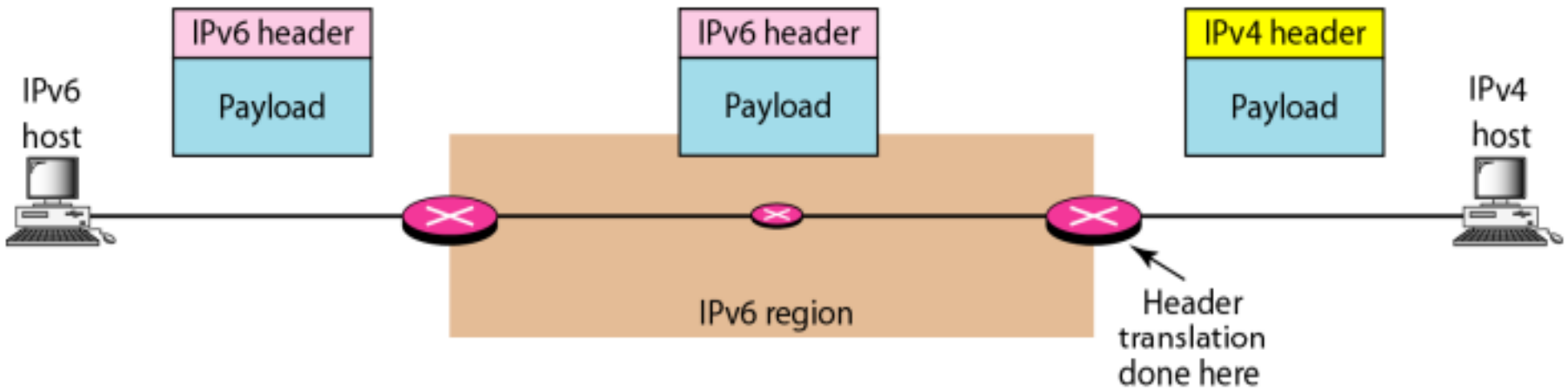
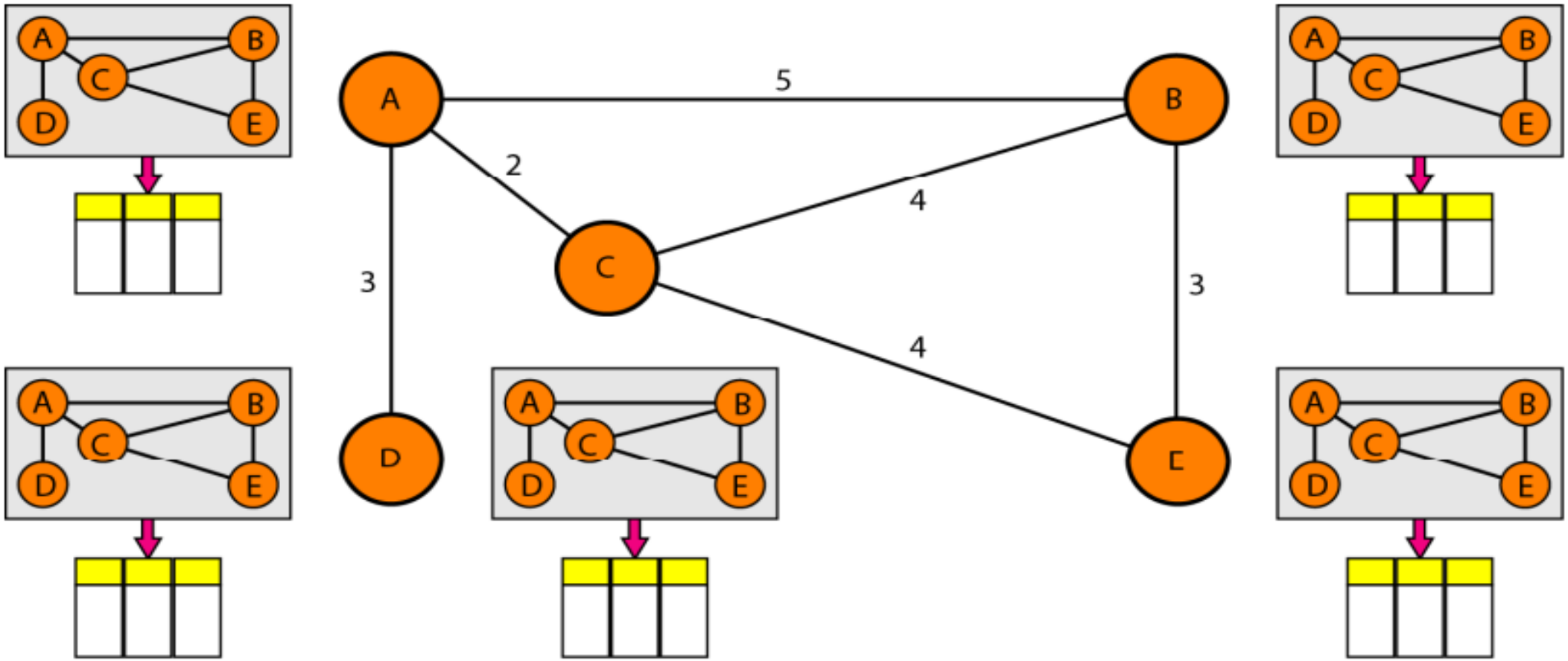


Table 20.11 *Header translation*

Header Translation Procedure
1. The IPv6 mapped address is changed to an IPv4 address by extracting the rightmost 32 bits.
2. The value of the IPv6 priority field is discarded.
3. The type of service field in IPv4 is set to zero.
4. The checksum for IPv4 is calculated and inserted in the corresponding field.
5. The IPv6 flow label is ignored.
6. Compatible extension headers are converted to options and inserted in the IPv4 header. Some may have to be dropped.
7. The length of IPv4 header is calculated and inserted into the corresponding field.
8. The total length of the IPv4 packet is calculated and inserted in the corresponding field.

LINK STATE PROTOCOL

Figure 22.20 *Concept of link state routing*



Link State Routing

Link state routing is a technique in which each router shares the knowledge of its neighborhood with every other router in the internetwork.

The three keys to understand the Link State Routing algorithm:

- **Knowledge about the neighborhood:** Instead of sending its routing table, a router sends the information about its neighborhood only. A router broadcast its identities and cost of the directly attached links to other routers.
- **Flooding:** Each router sends the information to every other router on the internetwork except its neighbors. This process is known as Flooding. Every router that receives the packet sends the copies to all its neighbors. Finally, each and every router receives a copy of the same information.
- **Information sharing:** A router sends the information to every other router only when the change occurs in the information.

Link State Routing has two phases:

Reliable Flooding

- **Initial state:** Each node knows the cost of its neighbors.
- **Final state:** Each node knows the entire graph.

Route Calculation

Each node uses Dijkstra's algorithm on the graph to calculate the optimal routes to all nodes.

- The Link state routing algorithm is also known as Dijkstra's algorithm which is used to find the shortest path from one node to every other node in the network.
- The Dijkstra's algorithm is an iterative, and it has the property that after k^{th} iteration of the algorithm, the least cost paths are well known for k destination nodes.

Figure 22.21 *Link state knowledge*

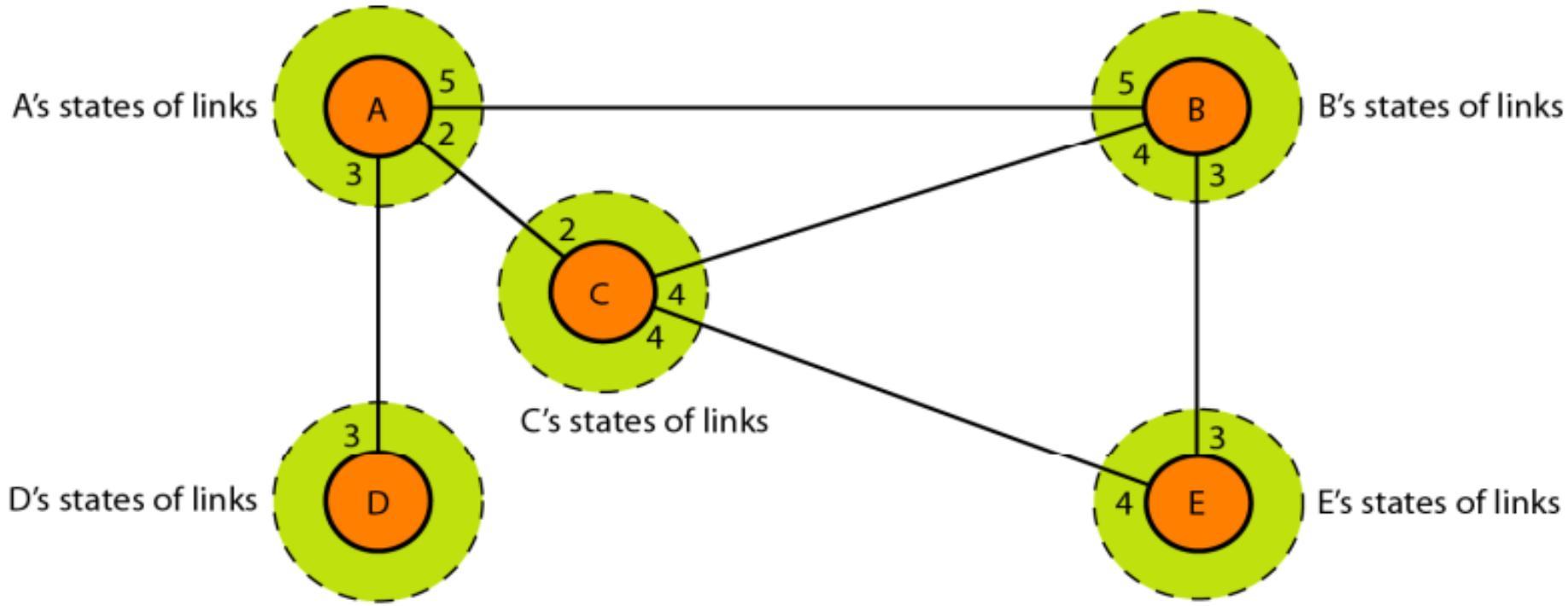


Figure 22.22 *Dijkstra algorithm*

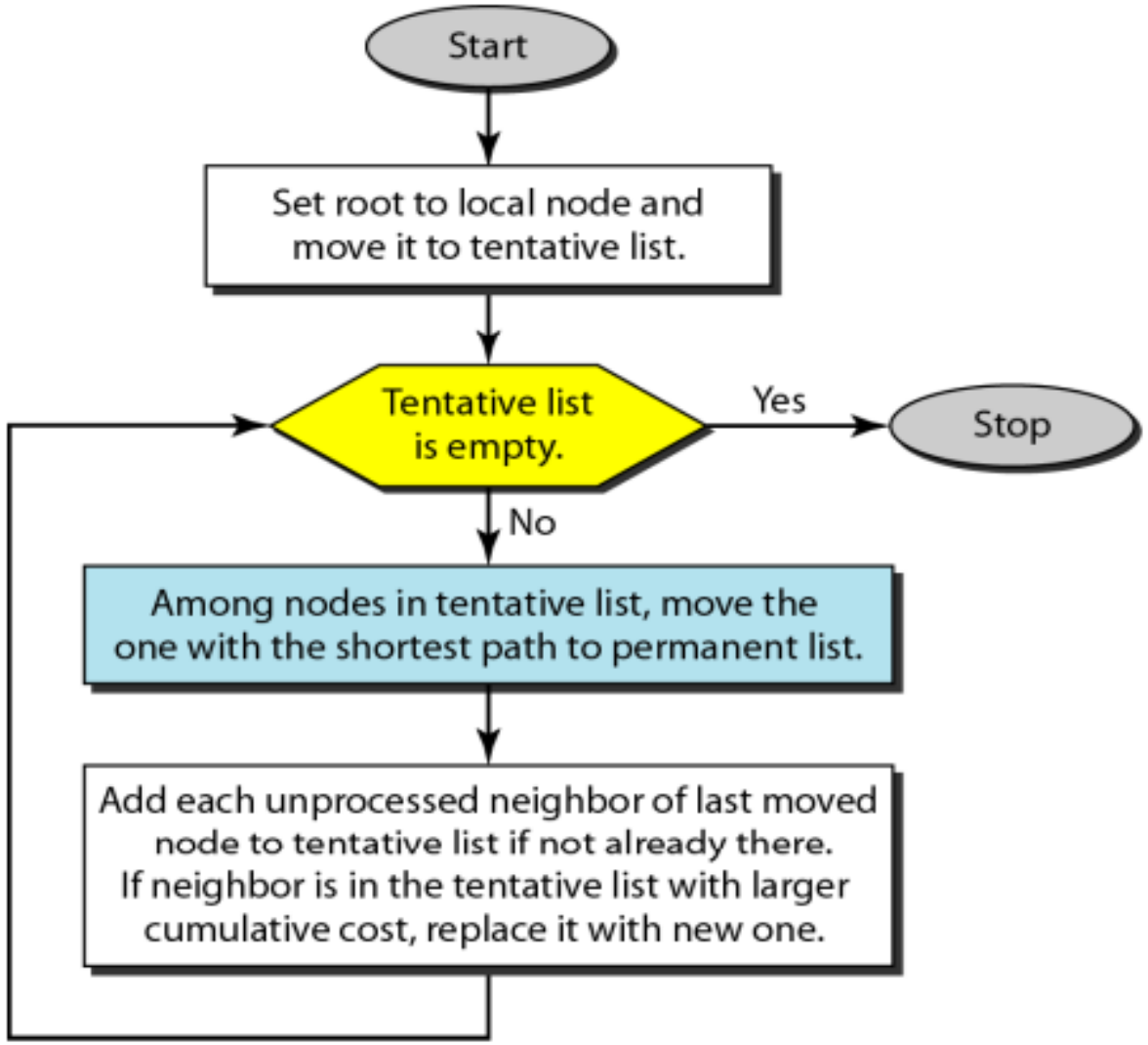


Figure 22.23 *Example of formation of shortest path tree*

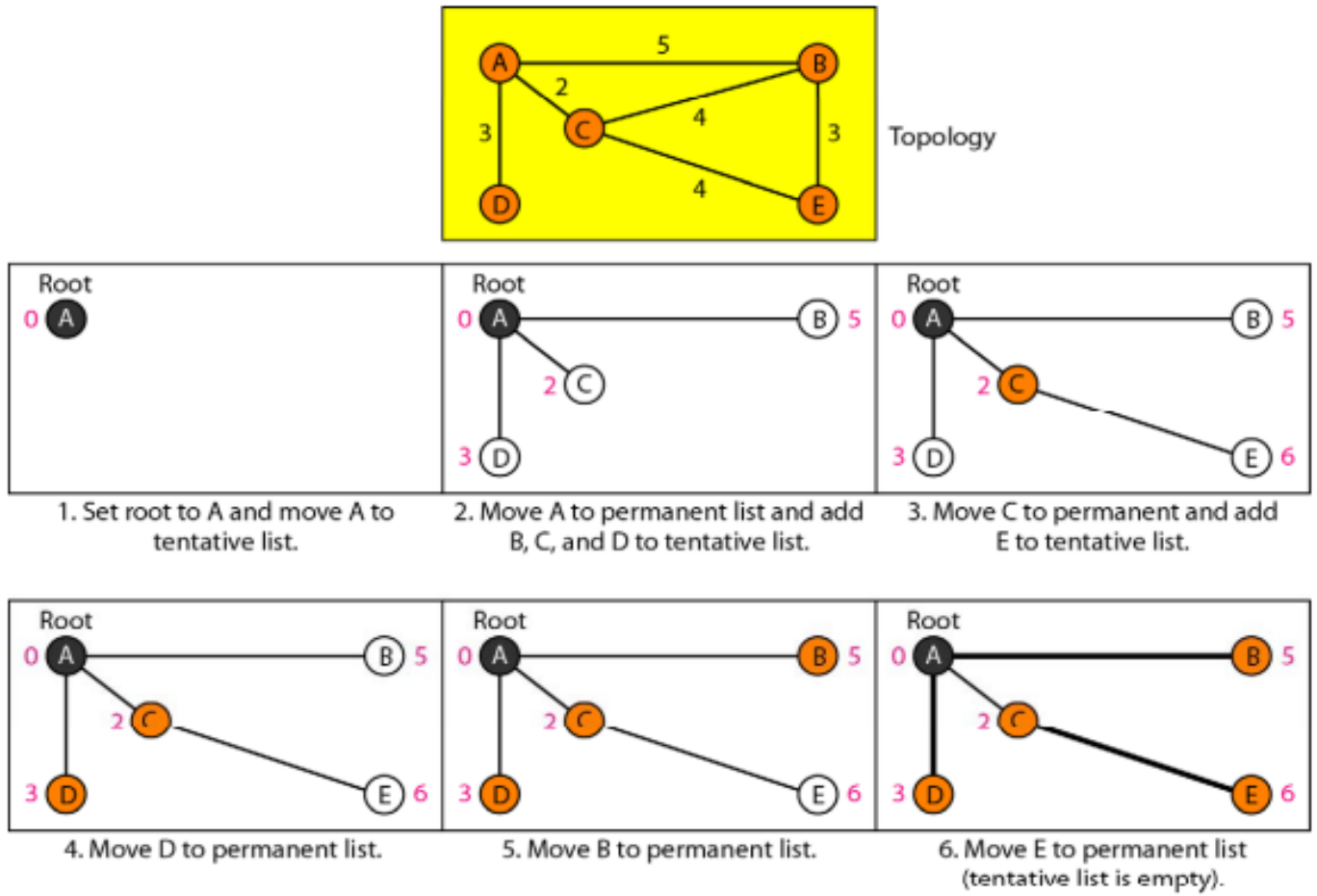
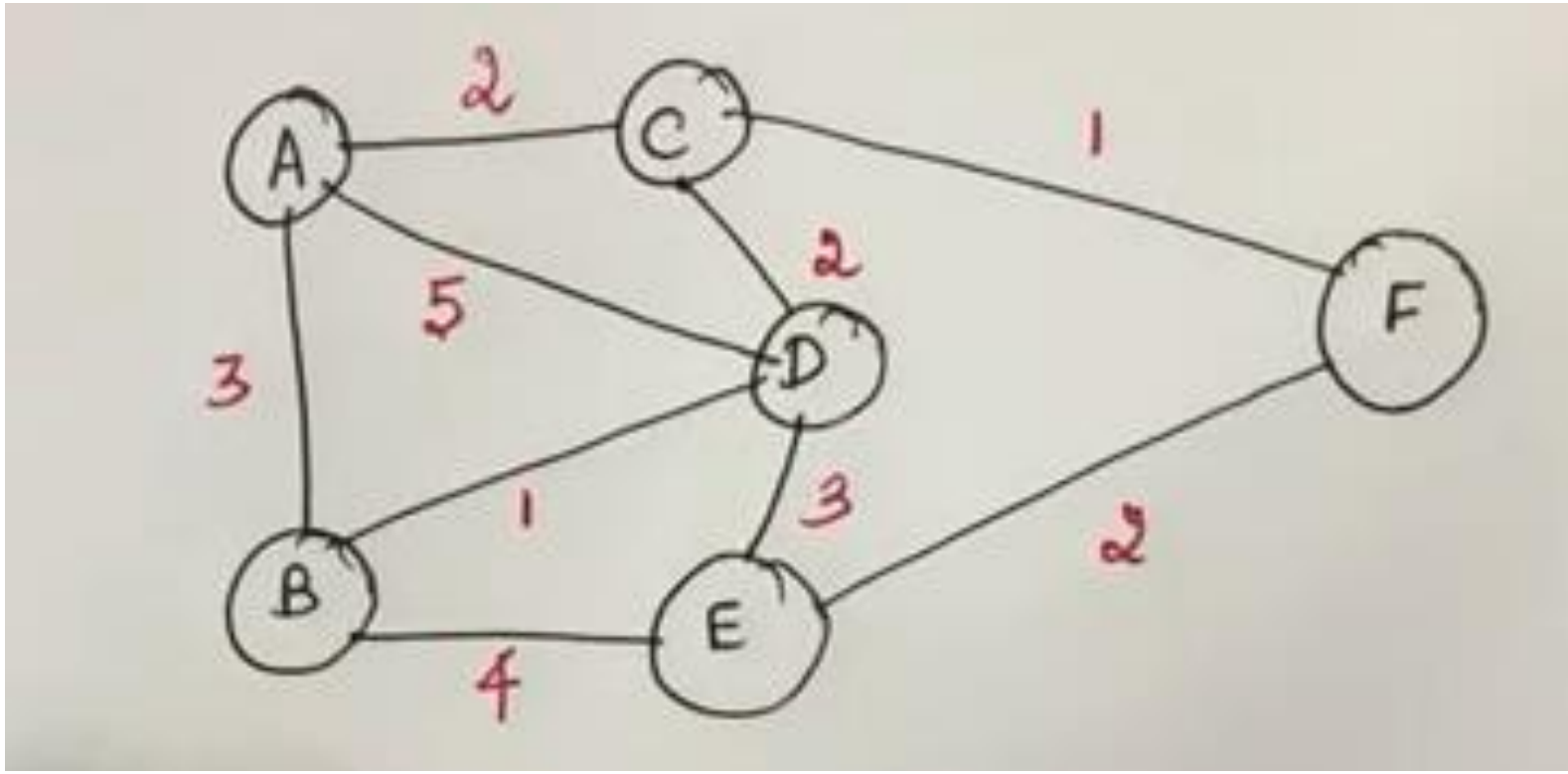
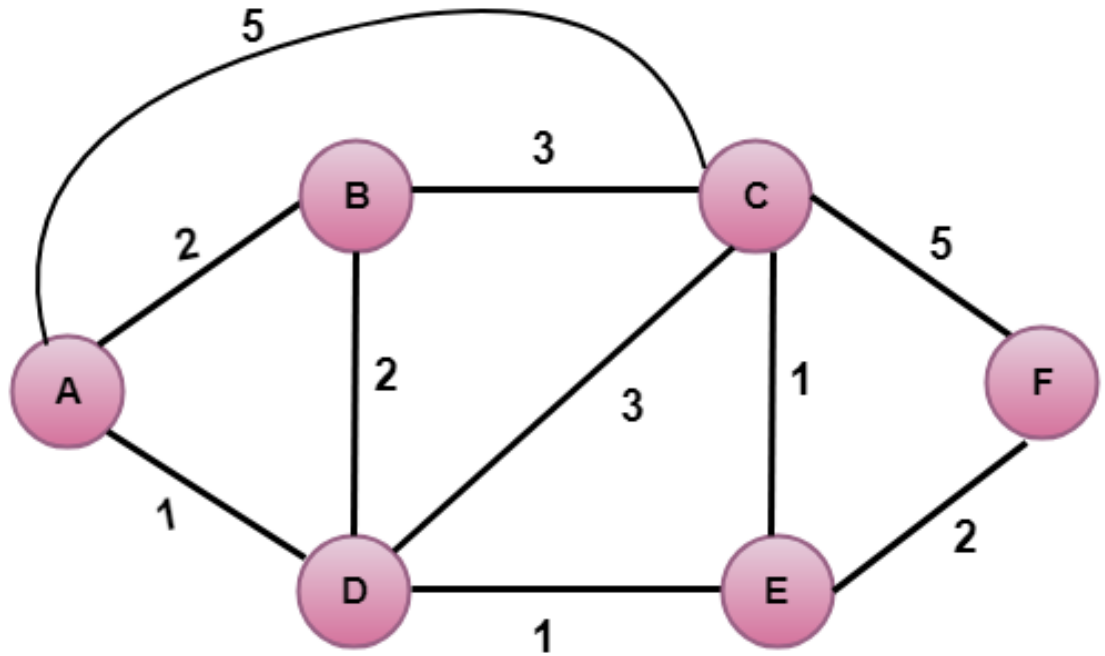
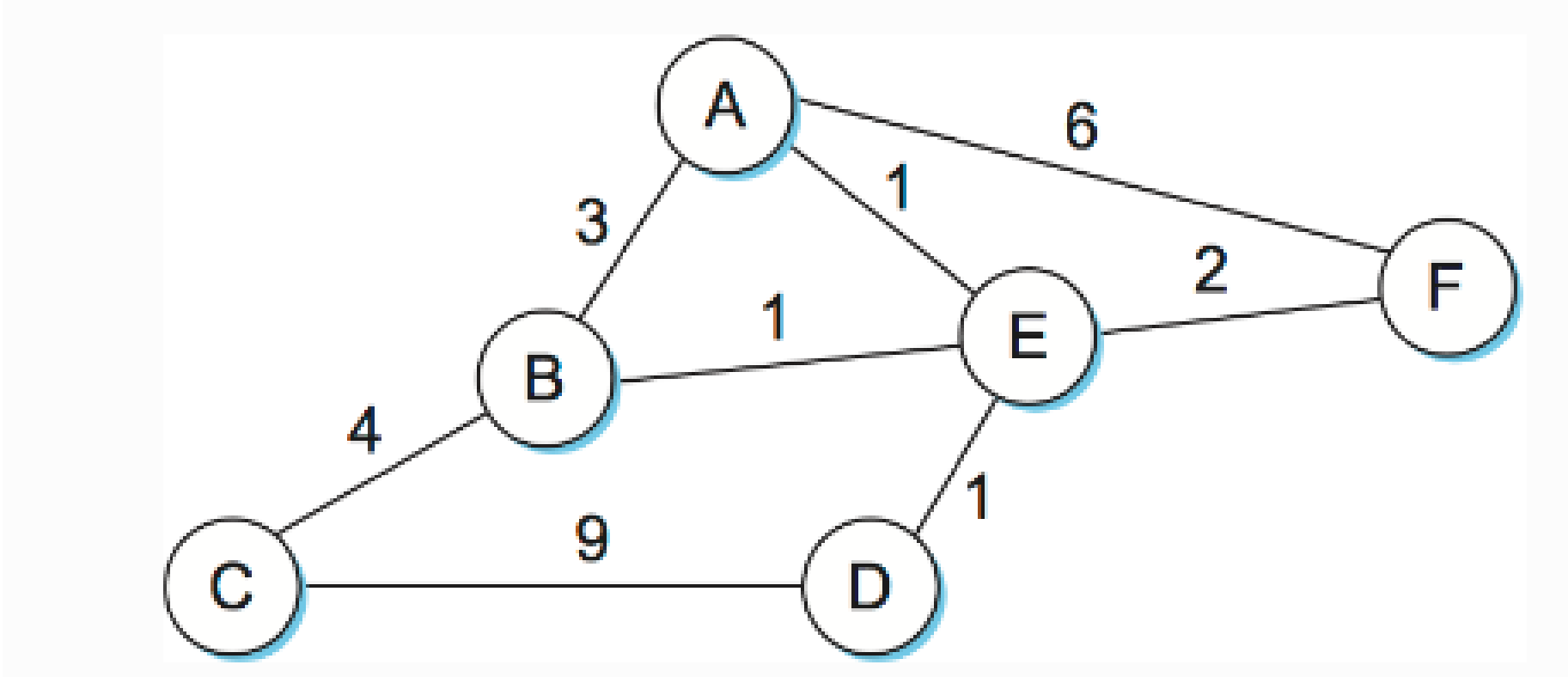


Table 22.2 *Routing table for node A*

Node	*Cost*	*Next Router*
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C


A
C
F
D
B
E
2
1
5
2
3
1
3
2
4


5
3
B
C
2
5
2
A
3
1
F
1
D
E
2
1


A
B
C
D
E
F
6
1
3
2
1
4
1
9