23CY404

COMPUTER NETWORKS

Introduction to Computer Network

Mr.R. Ashok,
Assistant Professor,
Computer Science and Engineering
(CYBER SECURITY)

Computer Networks

A collection of Autonomous computers interconnected by a single Technology.

or

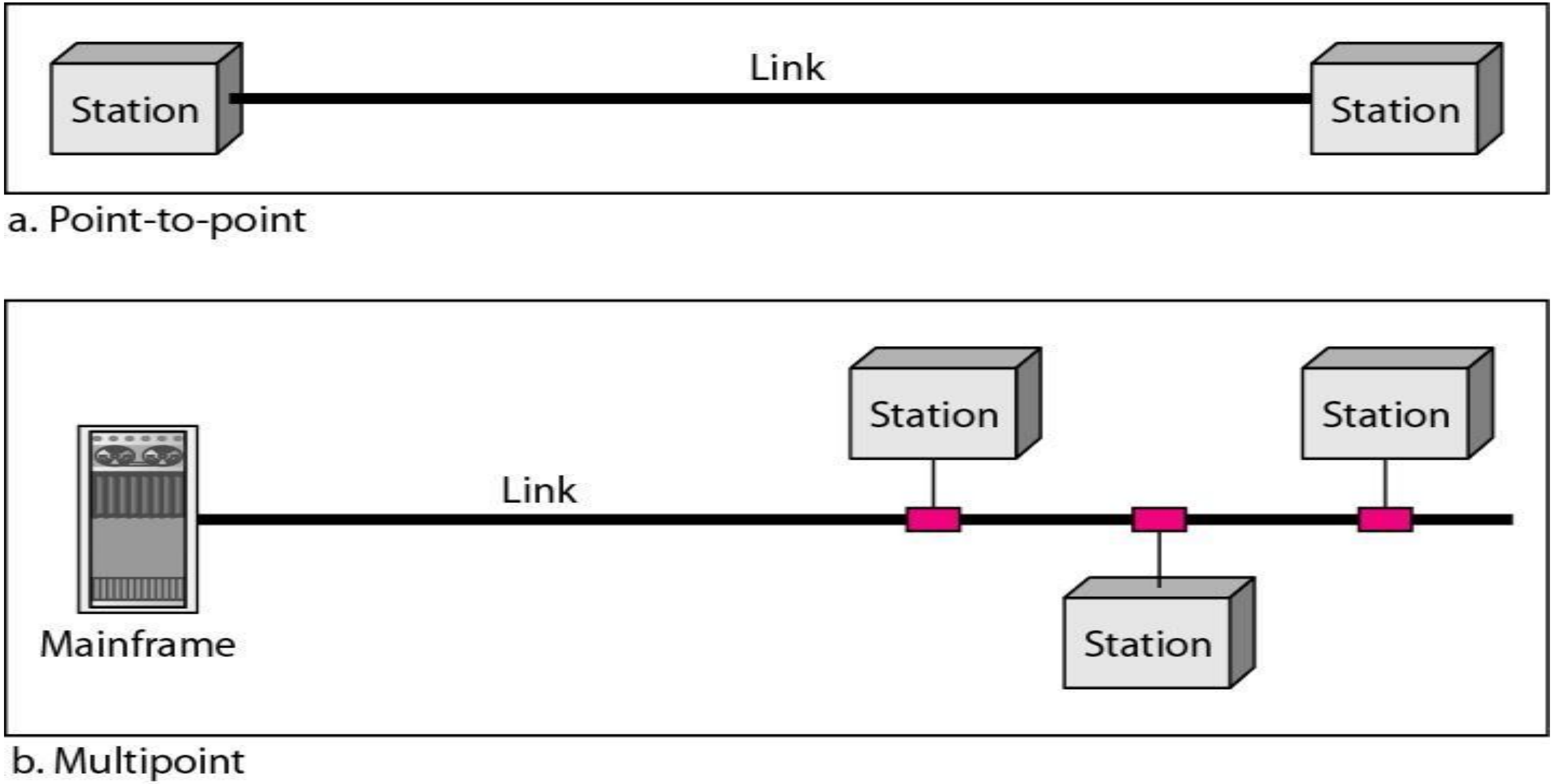
A set of nodes(***often referred to as computers***) connected by communication links.

NETWORK

A network is a set of devices (often referred to as nodes) connected by communication links.

A node can be a computer, printer, or any other device capable of sending and/or receiving data generated by other nodes on the network.

Types of connections: point-to-point and multipoint



Relational Database Model (1970s)

Edgar Codd's Relational Model:

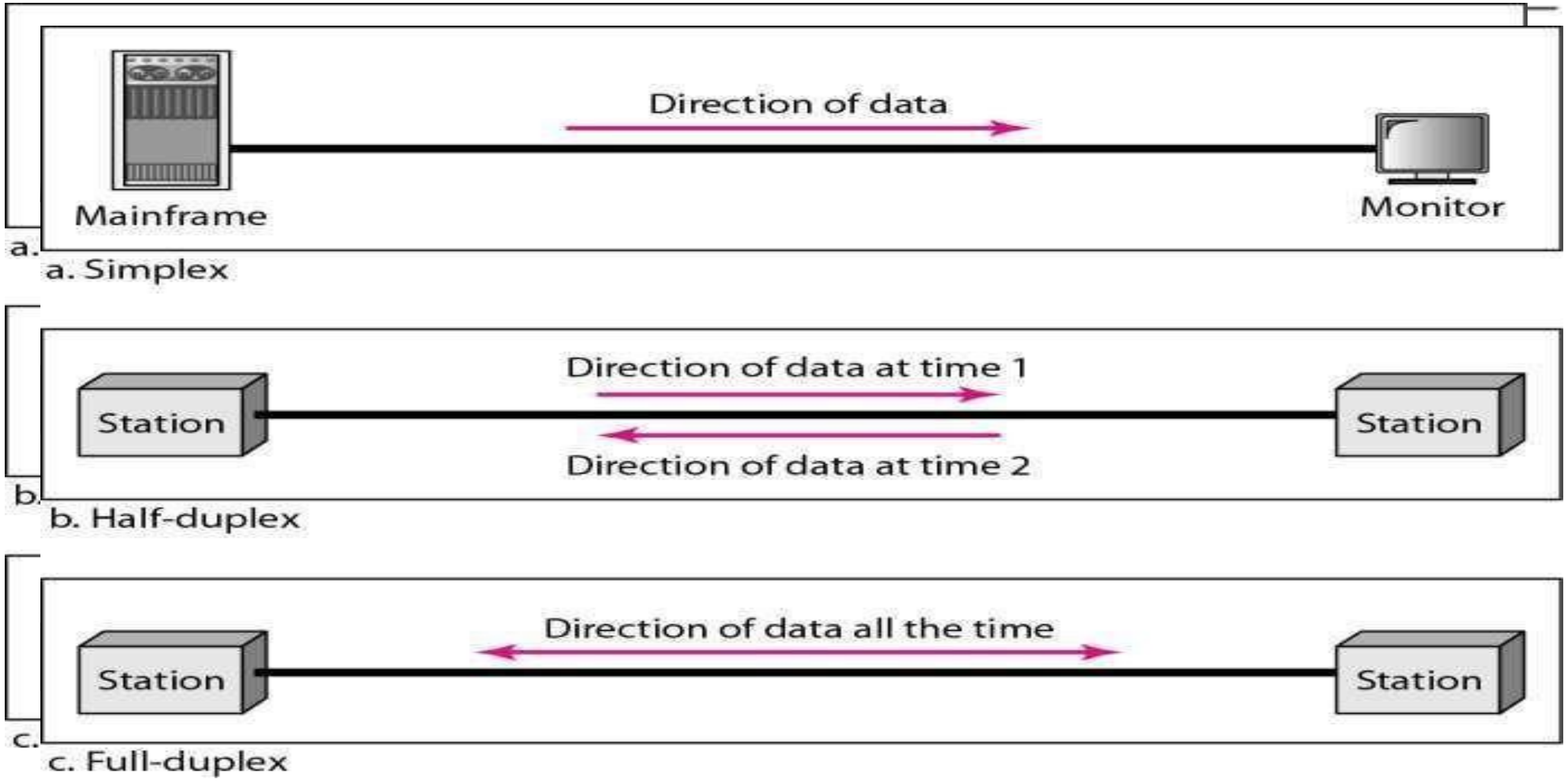
Proposed in late 1970s

Based on tables (rows = records, columns = fields)

Revolutionary for data efficiency and usability

Turing Award (1981) for the relational model

Data flow (simplex, half-duplex, and full-duplex)

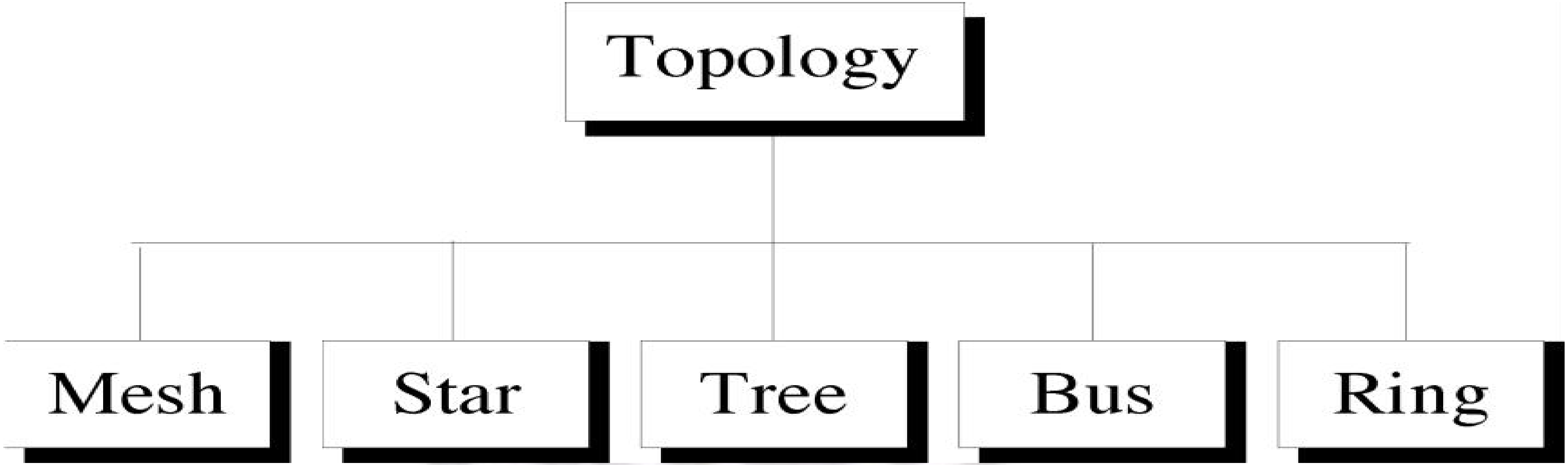


Network Topologies

What is a network topology?

In communication networks, a topology is a usually schematic description of the arrangement of a network, including its nodes and connecting lines.

Network topologies



Topology

Mesh

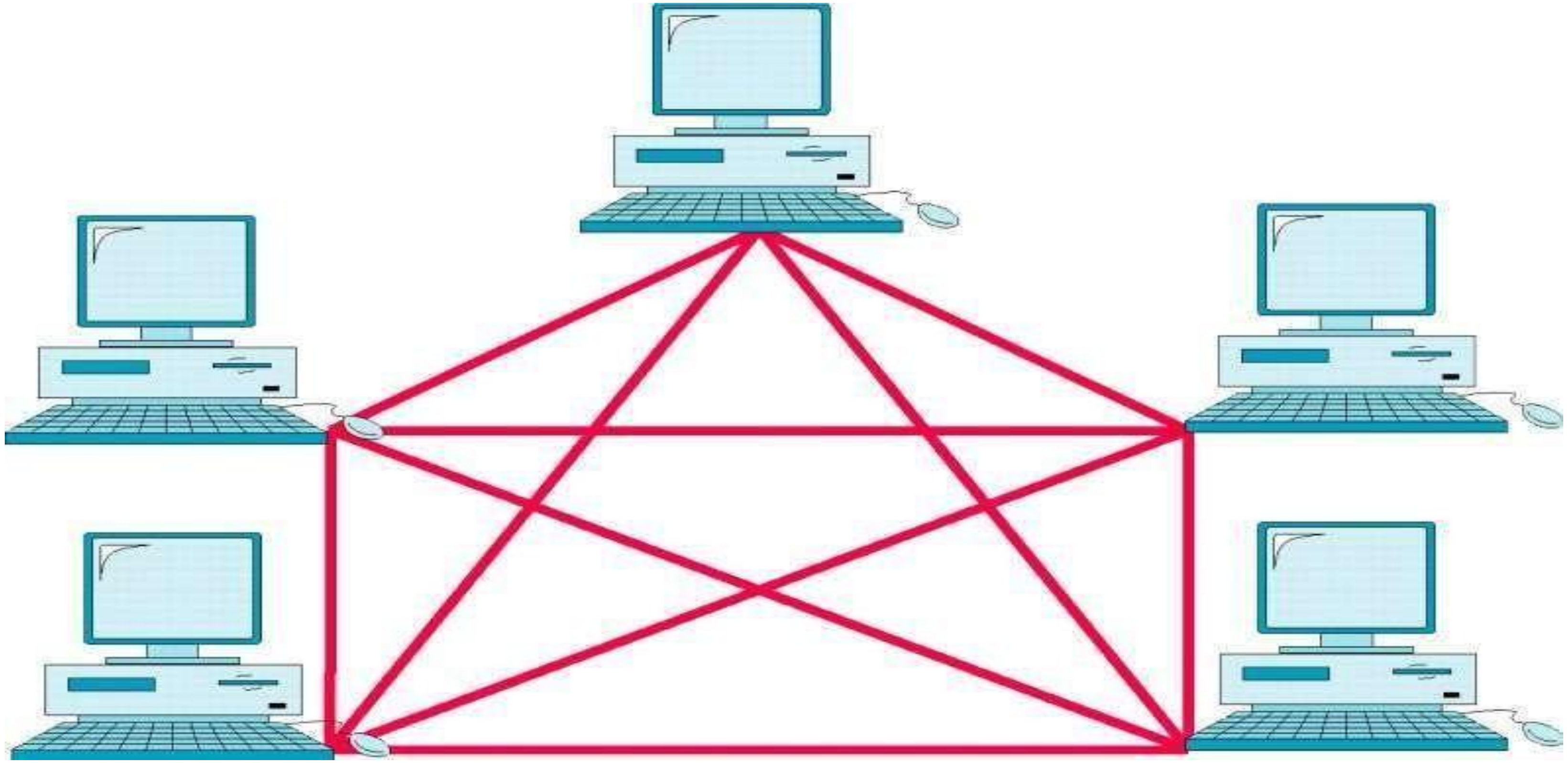
Star

Tree

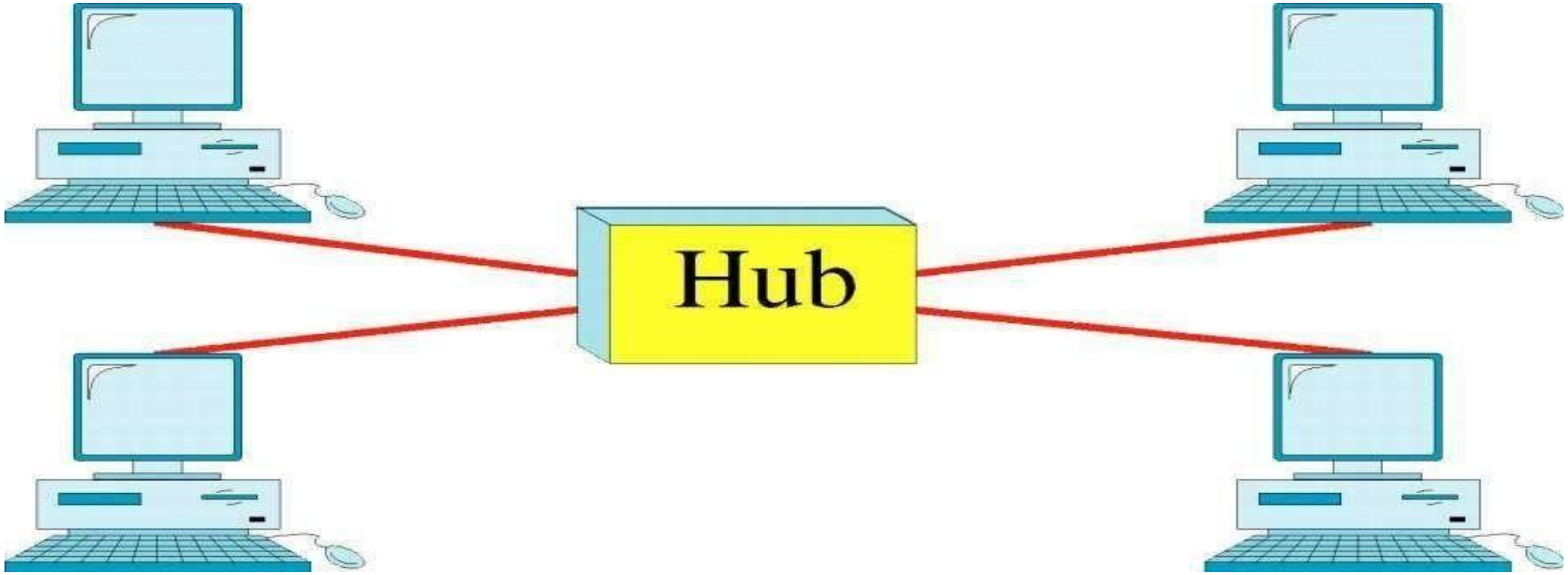
Bus

Ring

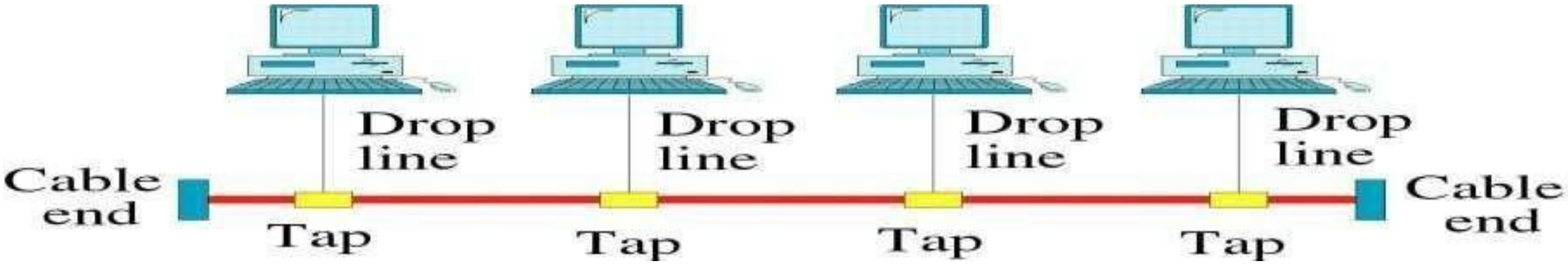
Mesh Topology



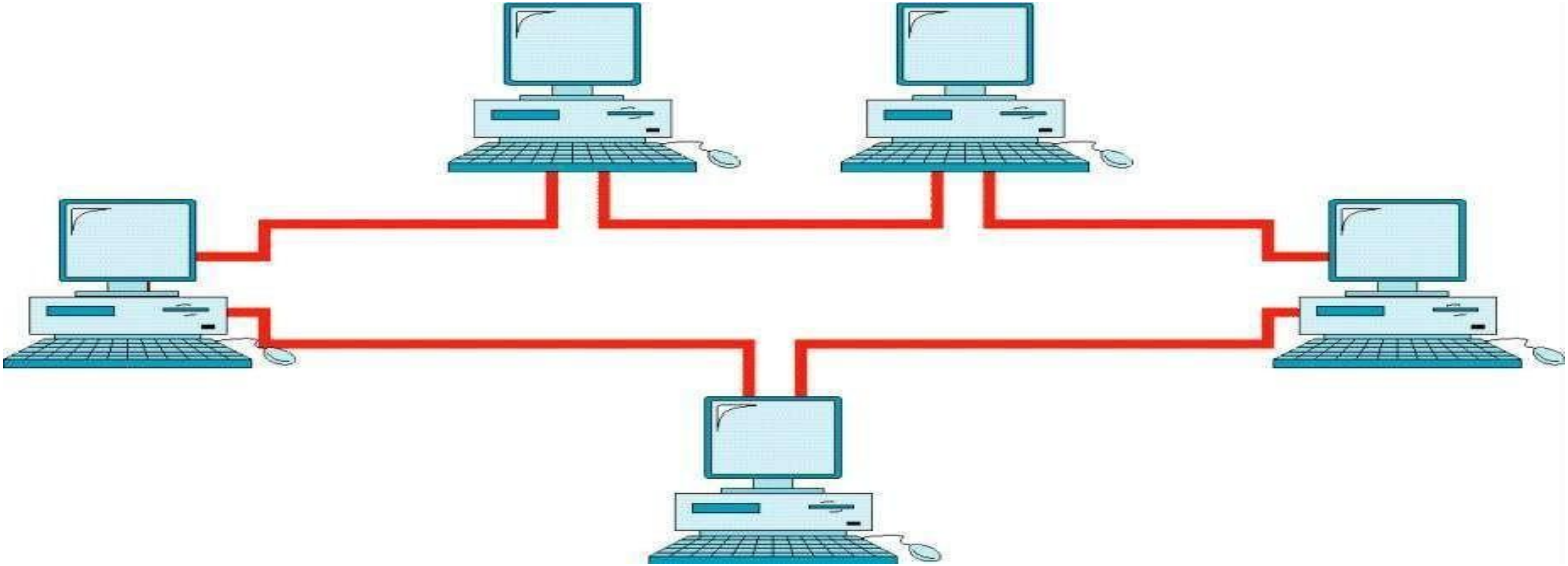
Star Topology



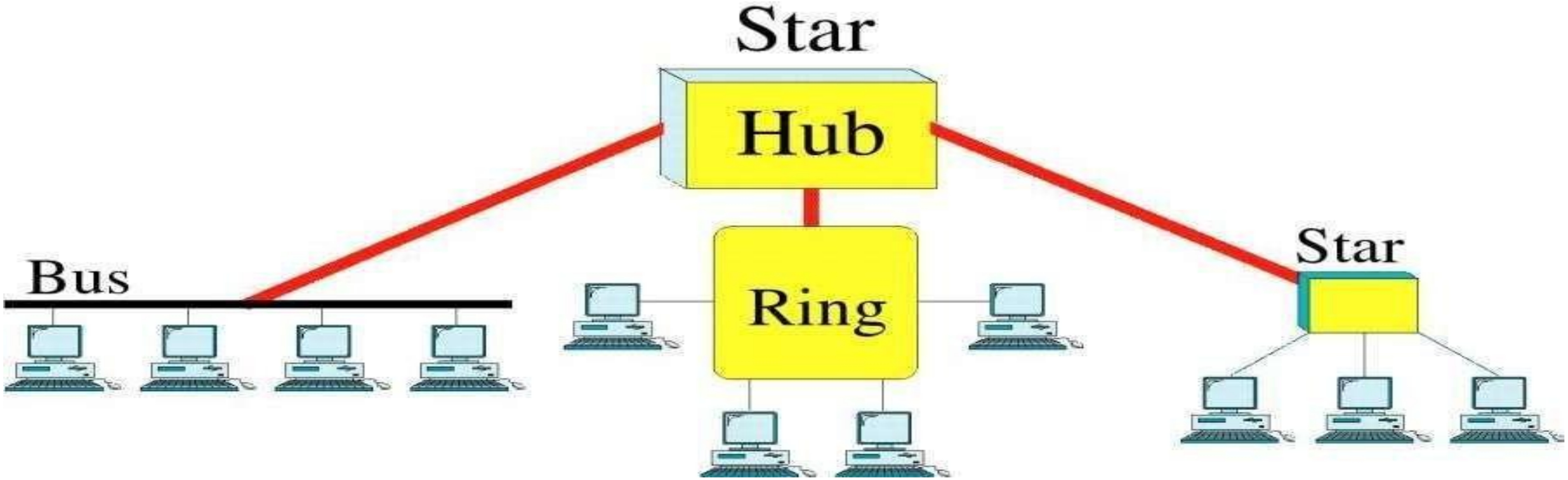
Bus Topology



Ring Topology

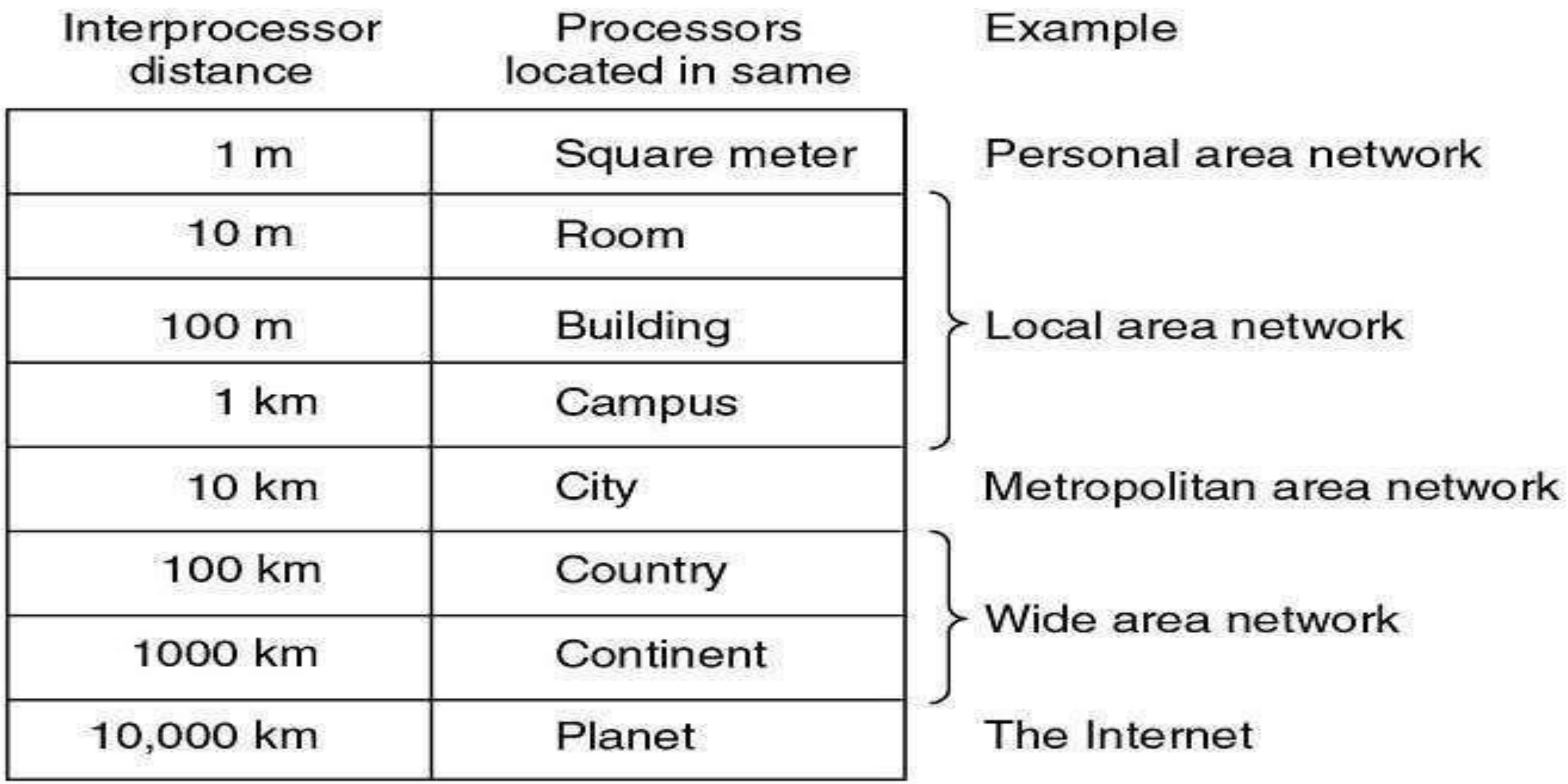


Hybrid Topology

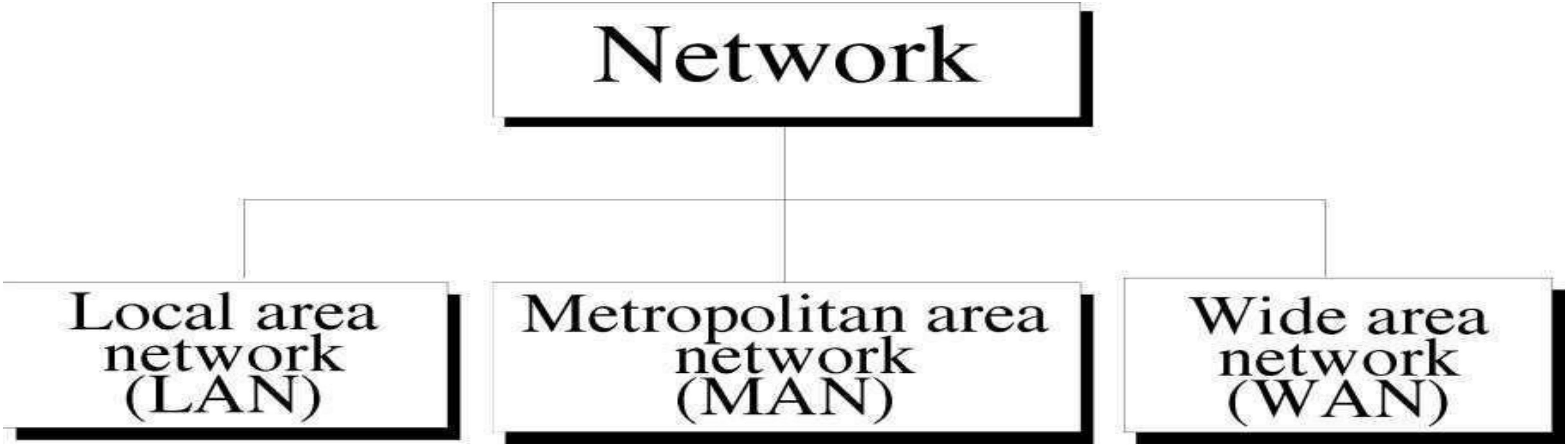


Categories of networks

1. Local Area Networks(LANs)
2. MetroPolitan Area Networks(MANs)
3. Wide Area Networks(WANs)



Interprocessor distance	Processors located in same	Example
1 m	Square meter	Personal area network
10 m	Room	Local area network
100 m	Building	
1 km	Campus	
10 km	City	Metropolitan area network
100 km	Country	Wide area network
1000 km	Continent	
10,000 km	Planet	The Internet



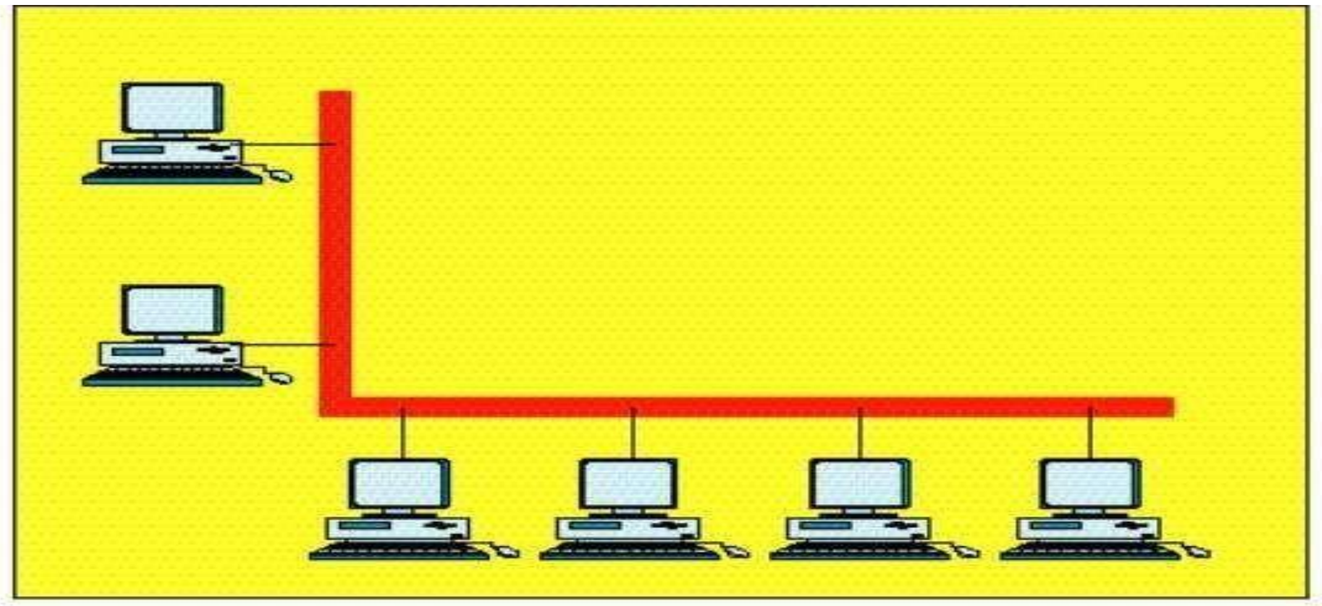
Network

Local area network (LAN)

Metropolitan area network (MAN)

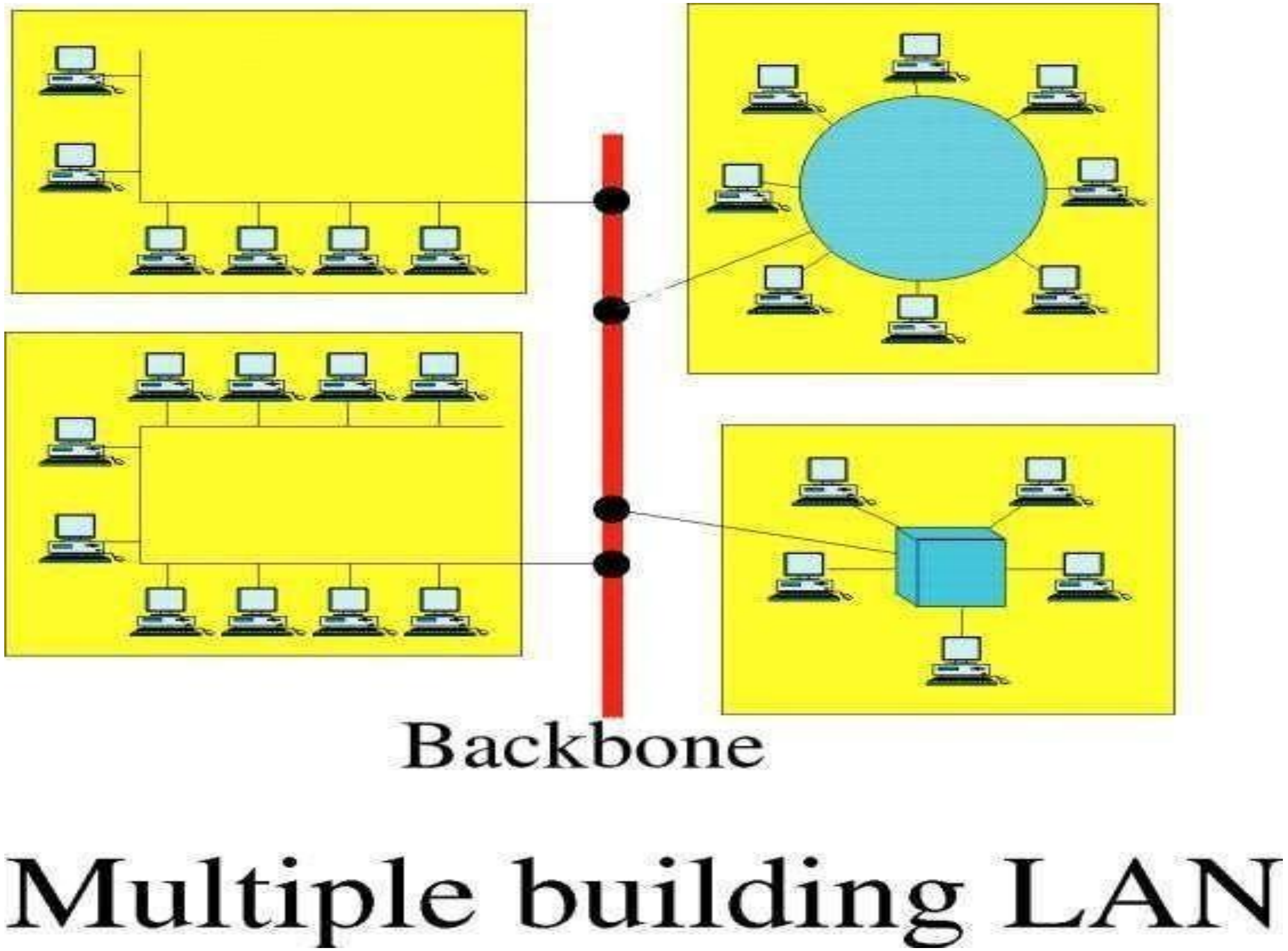
Wide area network (WAN)

Local Area Network

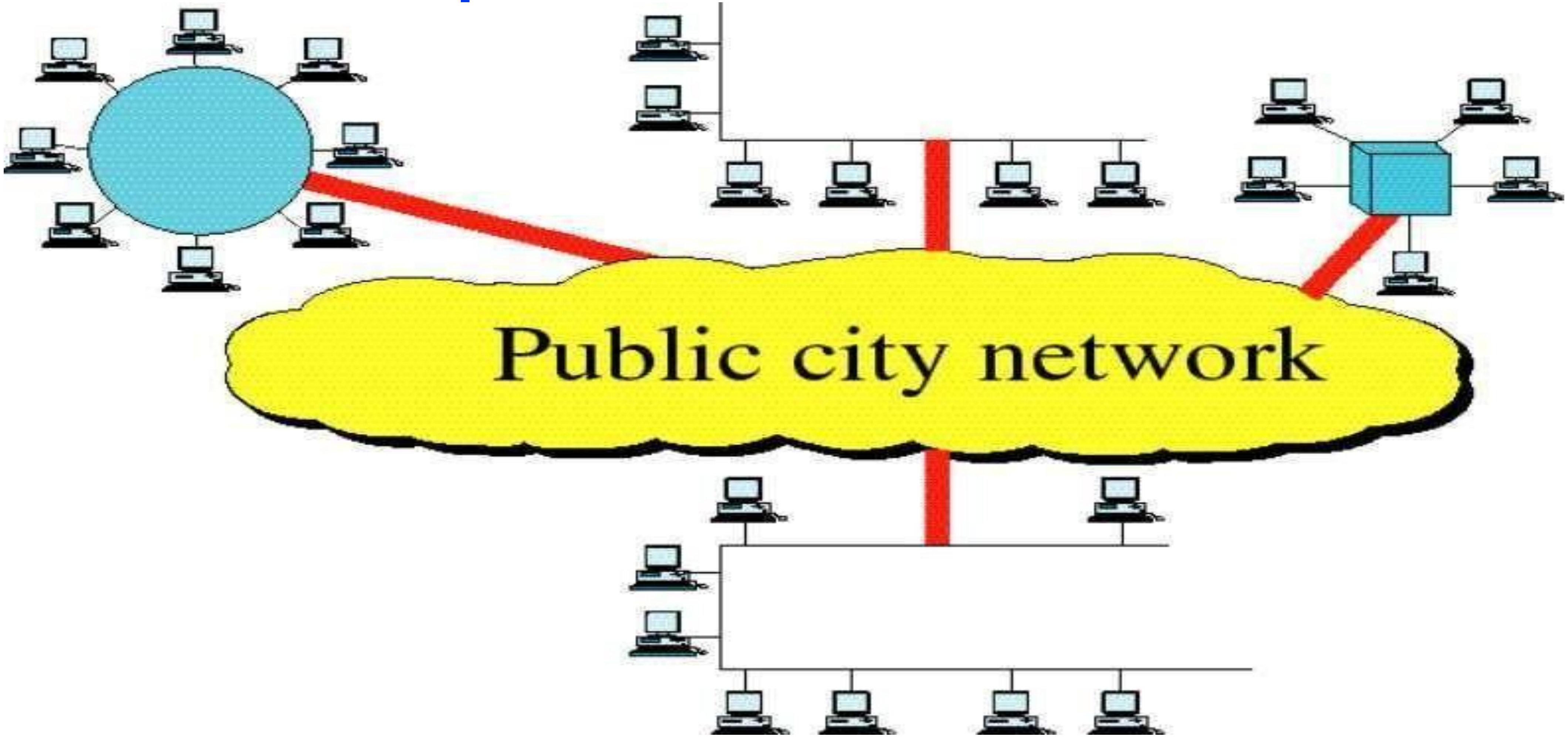


Single building LAN

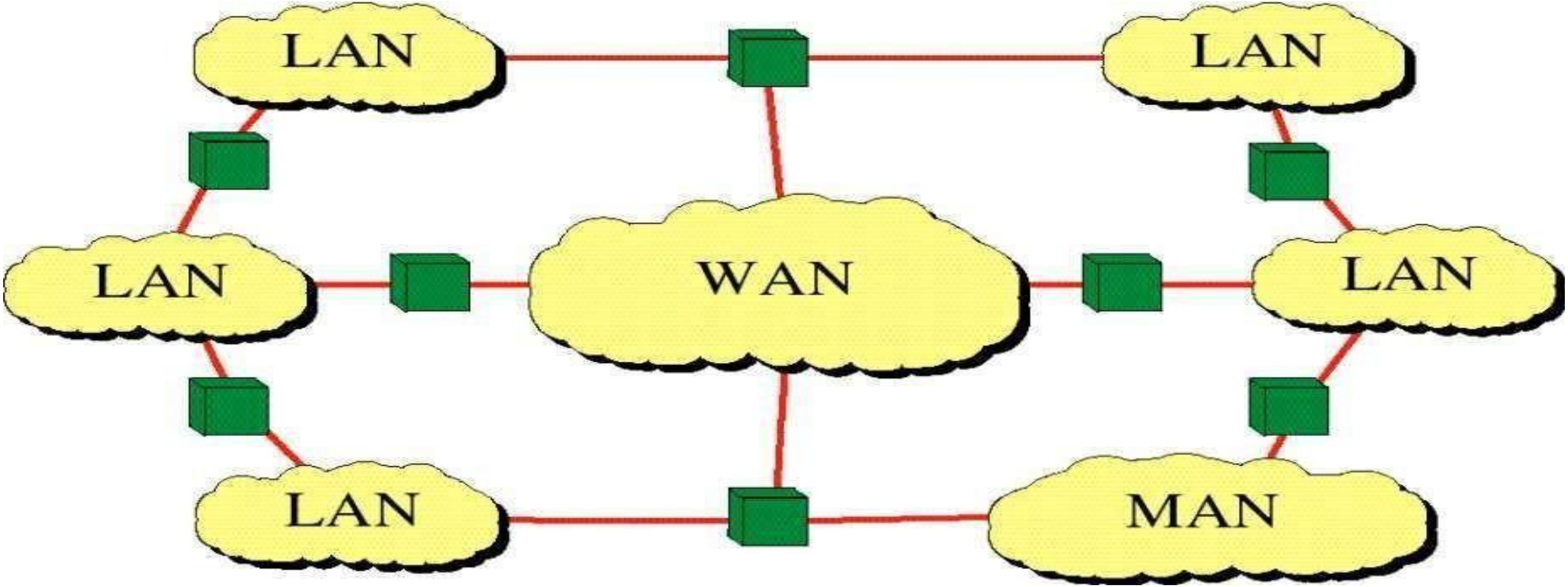
Local Area Network



Metropolitan Area Network

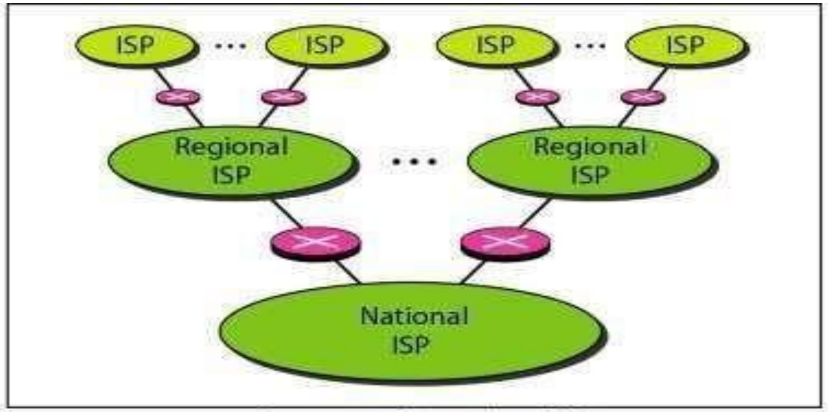


Internetwork (Internet)

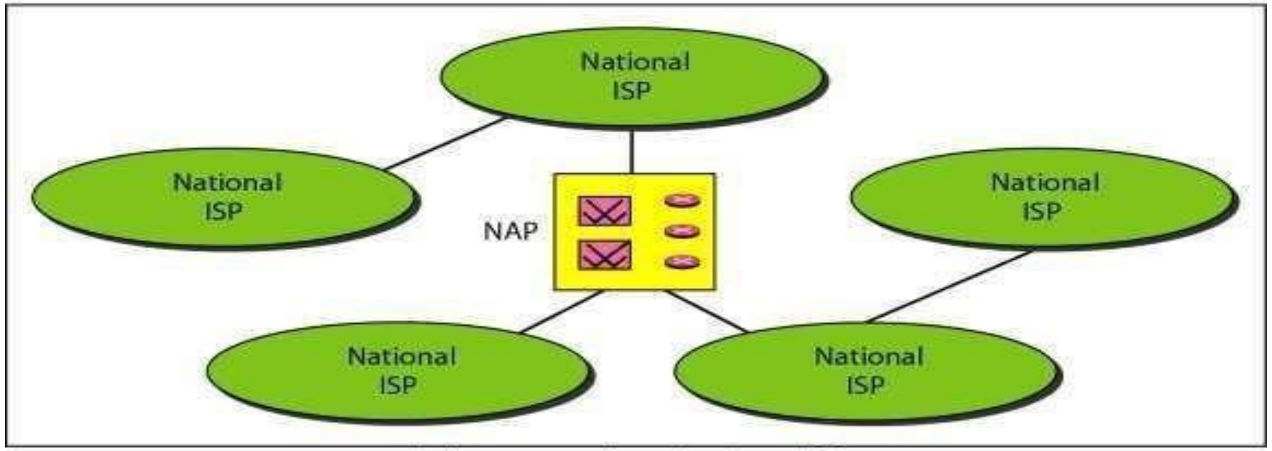


THE INTERNET

Hierarchical organization of the Internet



a. Structure of a national ISP



b. Interconnection of national ISPs

EX: National ISP’s

- SprintLink
- PSINet
- UUNet Technology
- AGIS and internet MCI

PROTOCOLS

- A protocol is a set of rules that govern data communications.
- The key elements of a protocol are

-> syntax

-> semantics

-> timing.

Standards Organizations

- standards creation committees

->International Organization for Standardization (ISO)

->International Telecommunication Union-Telecommunication Standards Sector (ITU-T)

->American National Standards Institute (ANSI)

->Institute of Electrical and Electronics Engineers (IEEE).

->Electronic Industries Association (EIA)

Standards Organizations

- standards creation committees

->International Organization for Standardization (ISO)

->International Telecommunication Union-Telecommunication Standards Sector (ITU-T)

->American National Standards Institute (ANSI)

->Institute of Electrical and Electronics Engineers (IEEE).

->Electronic Industries Association (EIA)

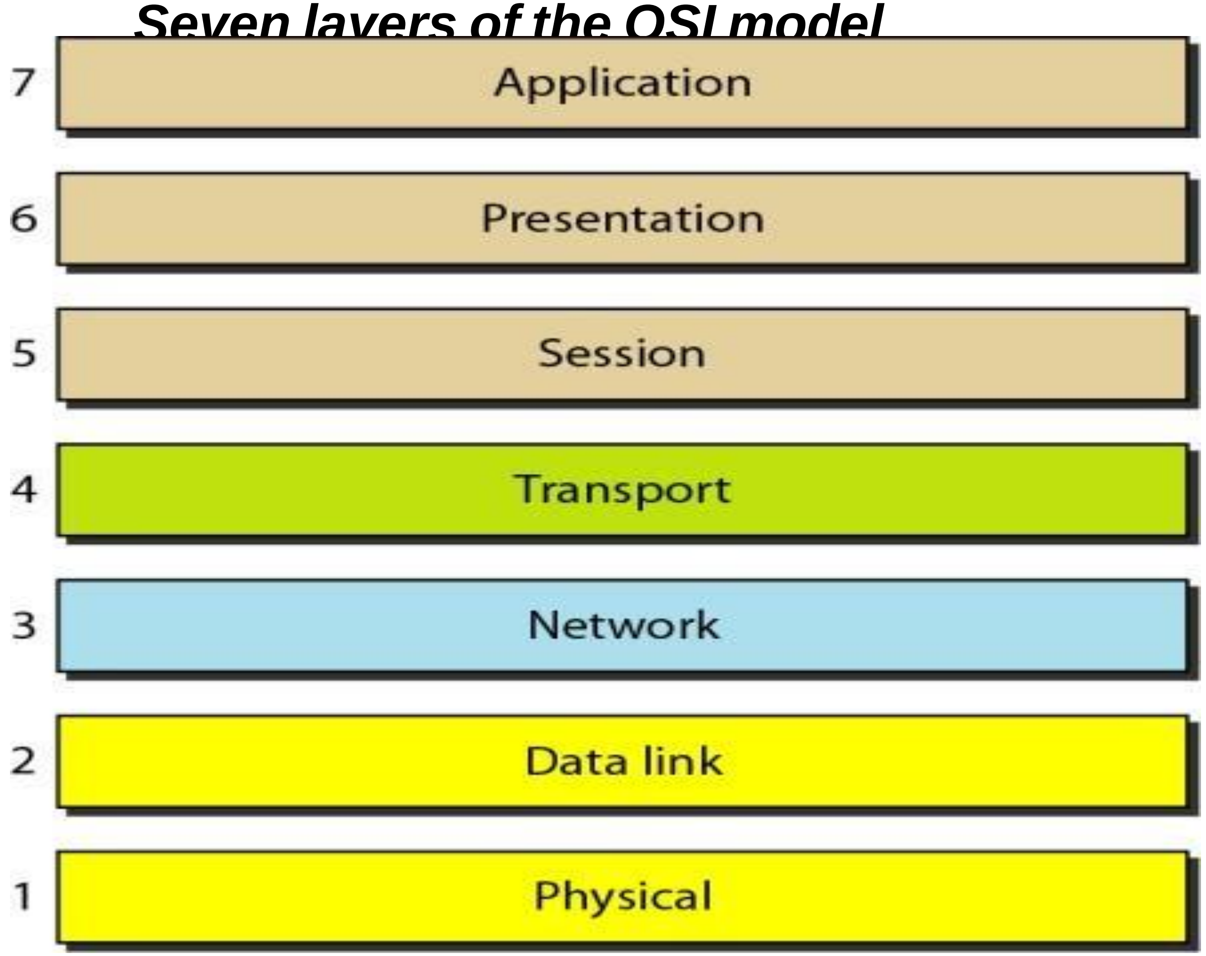
Network Model

The OSI MODEL

- ***Established in 1947, the International Standards Organization (ISO) is a multinational body dedicated to worldwide agreement on international standards.***

- ***An ISO standard that covers all aspects of network communications is the Open Systems Interconnection (OSI) model. It was first introduced in the late 1970s.***

Seven layers of the OSI model



7 Application

6 Presentation

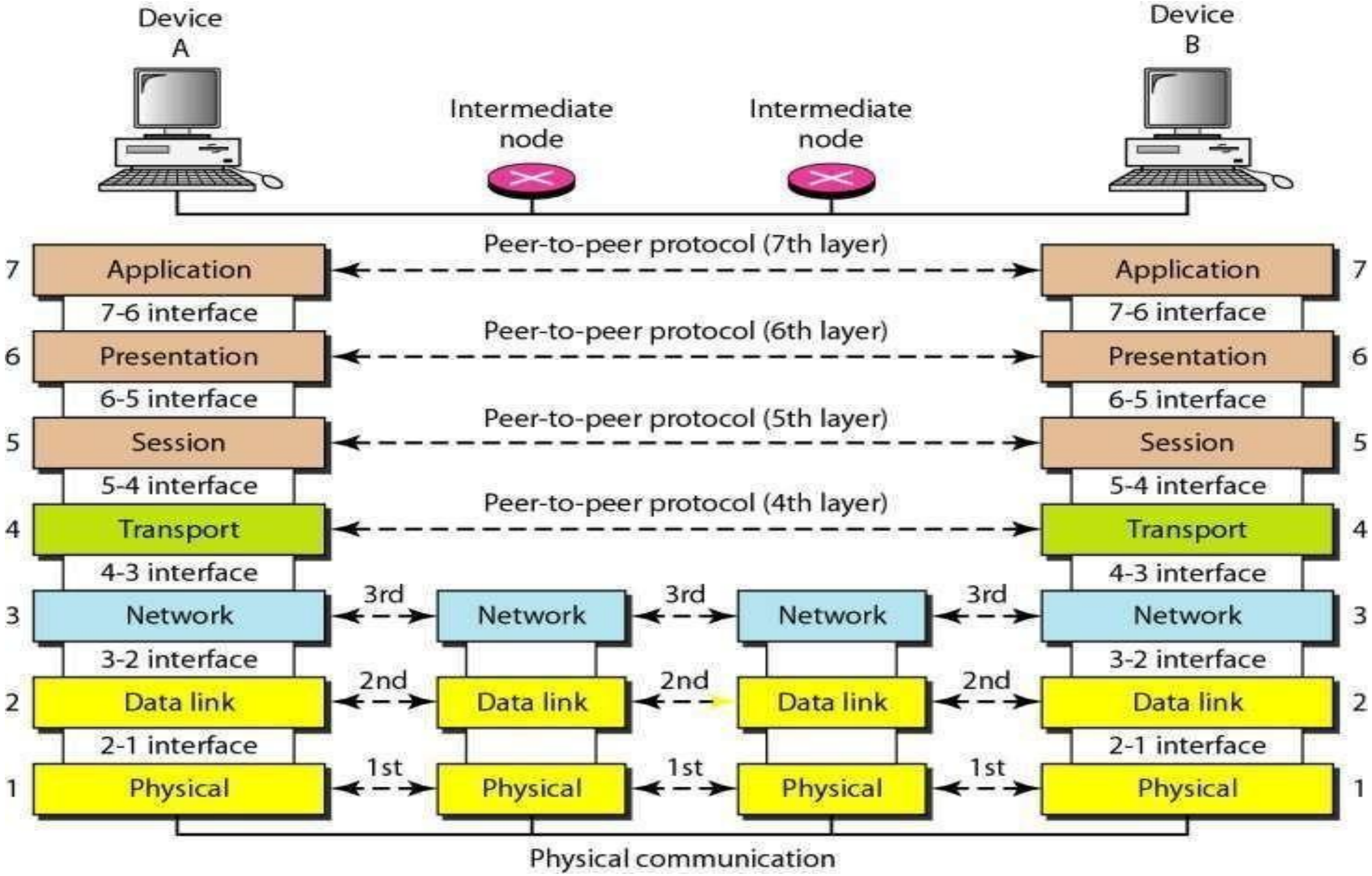
5 Session

4 Transport

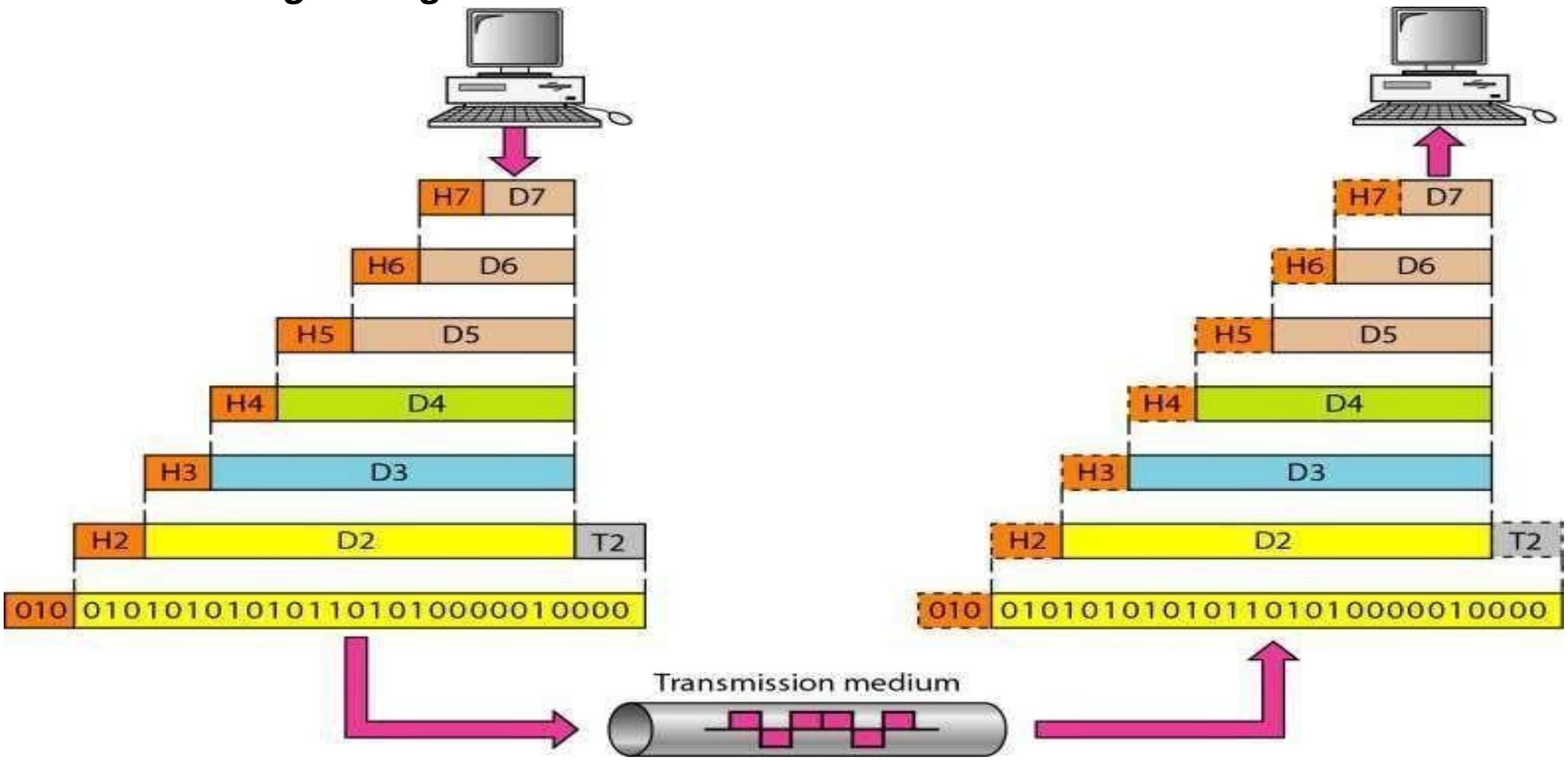
3 Network

2 Data link

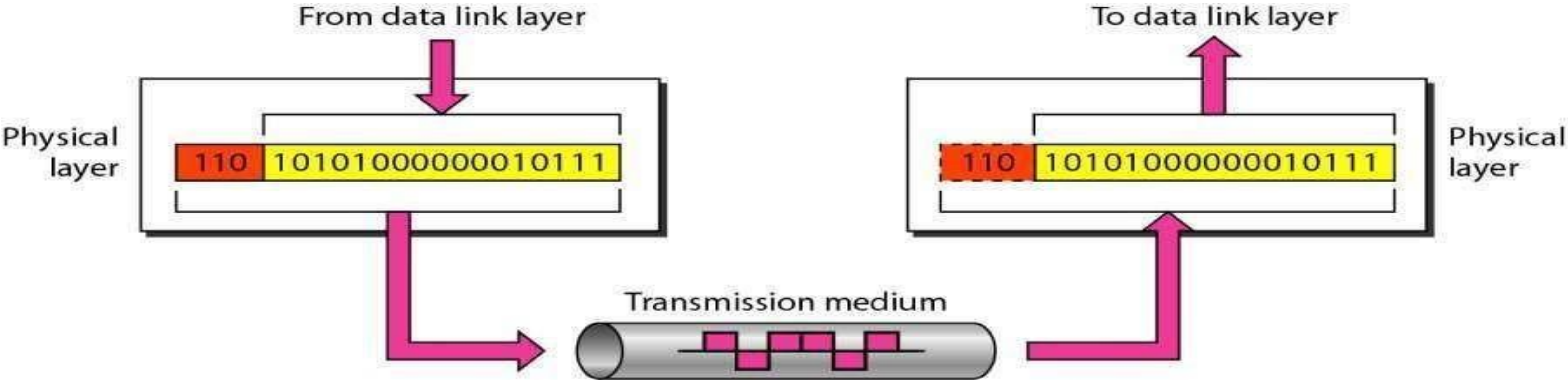
1 Physical


Device A
Device B
Intermediate node
Intermediate node
Peer-to-peer protocol (7th layer)
Peer-to-peer protocol (6th layer)
Peer-to-peer protocol (5th layer)
Peer-to-peer protocol (4th layer)
7 Application
7-6 interface
6 Presentation
6-5 interface
5 Session
5-4 interface
4 Transport
4-3 interface
3 Network
3-2 interface
2 Data link
2-1 interface
1 Physical
3rd
2nd
1st
Network
Data link
Physical
Physical communication

An exchange using the OSI model



Physicallayer

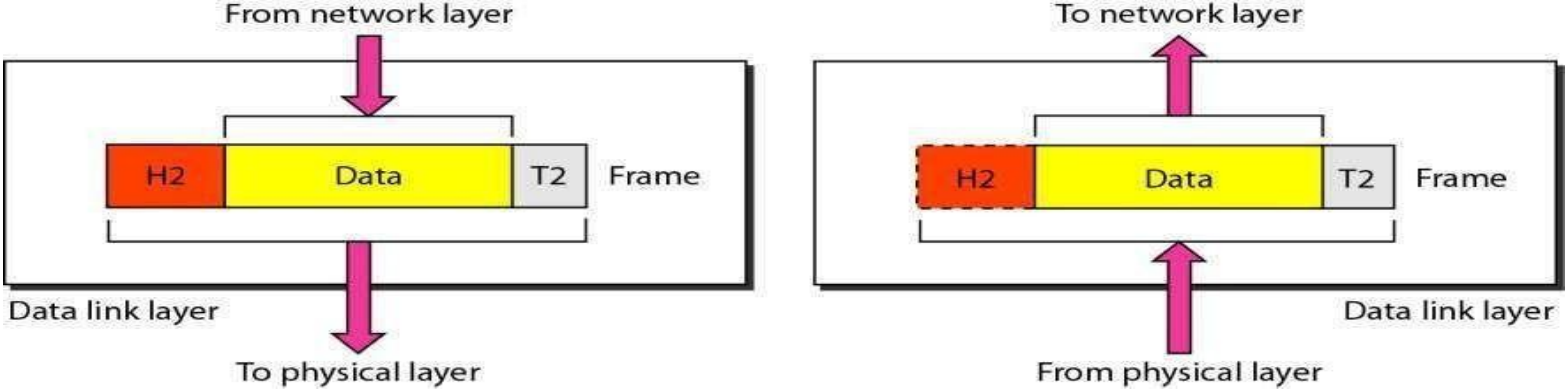


The physical layer is responsible for movements of individual bits from one hop (node) to the next.

Physical Layer Functionalities

- Physical characteristics of interfaces and medium
- Representation of bits
- Data rate
- Synchronization of bits
- Line configuration.
- Physical topology
- Transmission mode

Data link layer

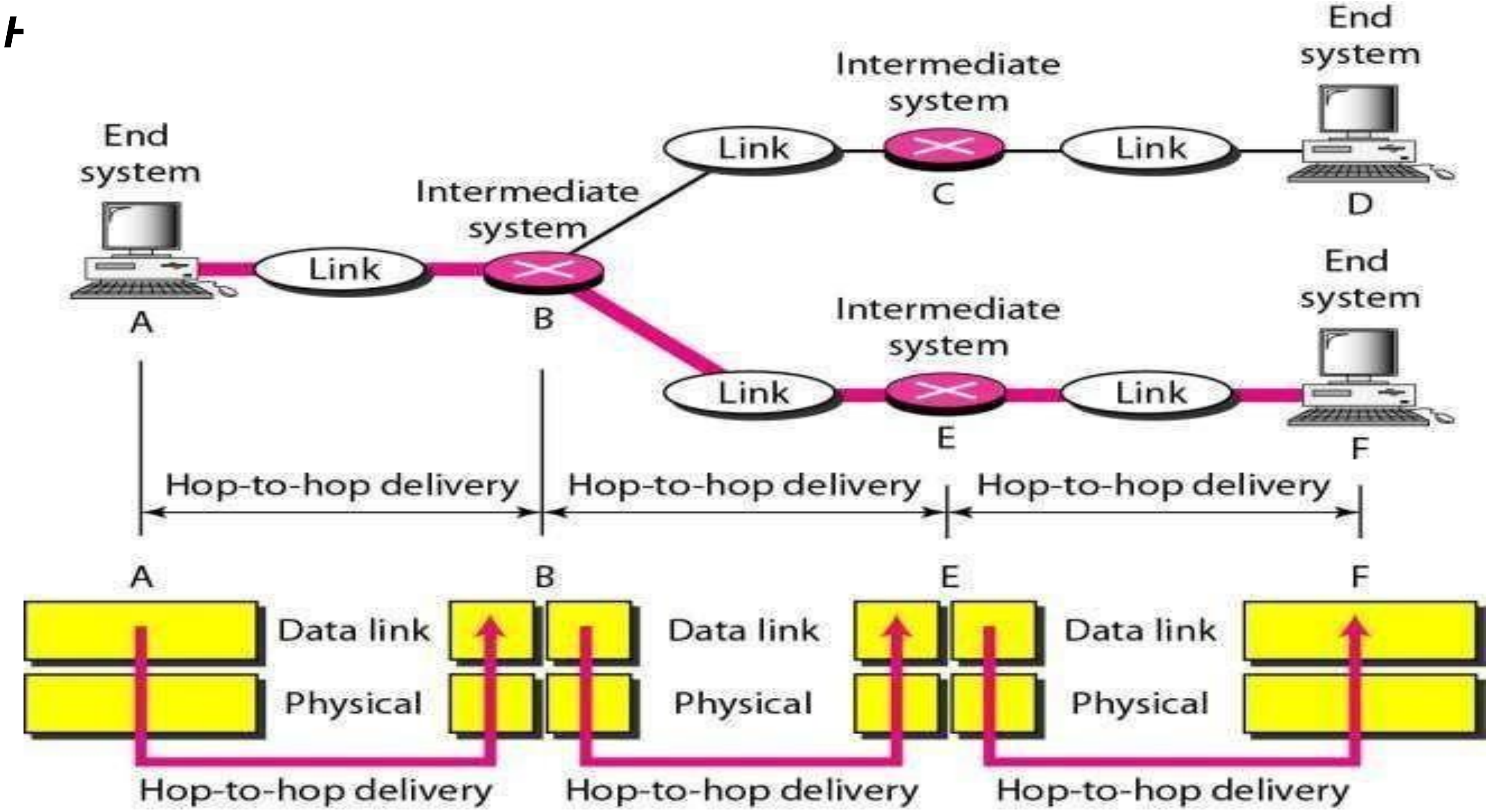


The data link layer is responsible for moving frames from one hop (node) to the next.

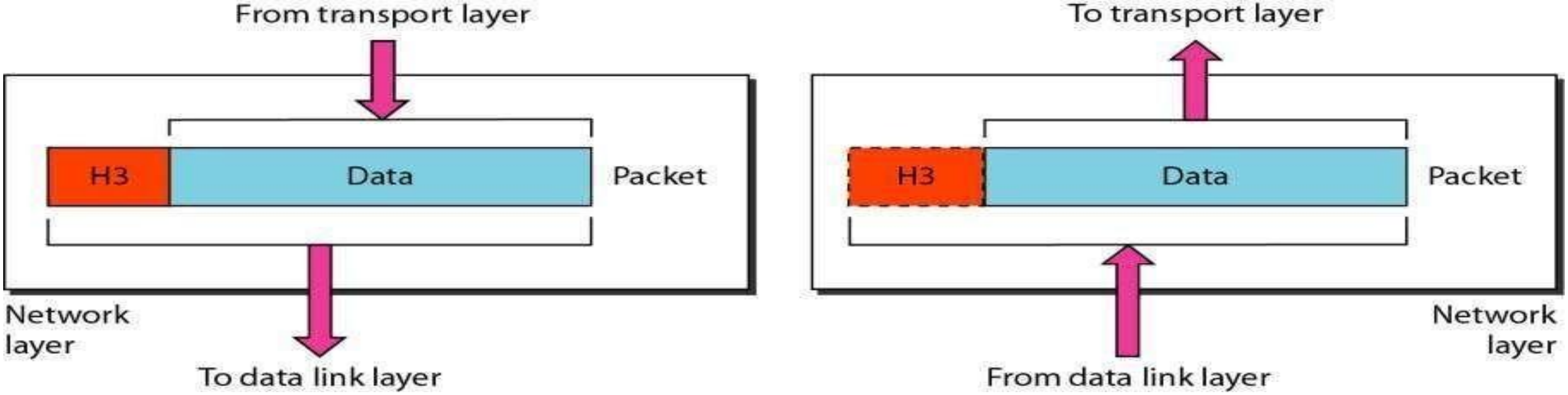
Data link layer

- Framing.
- Physical addressing
- Flow control
- Error control
- Access control

H


End system
Intermediate system
Intermediate system
End system
Intermediate system
End system
Link
Link
Link
Link
Link
A
B
C
D
E
F
Hop-to-hop delivery
Hop-to-hop delivery
Hop-to-hop delivery
A
B
E
F
Data link
Data link
Data link
Physical
Physical
Physical
Hop-to-hop delivery
Hop-to-hop delivery
Hop-to-hop delivery

Network layer

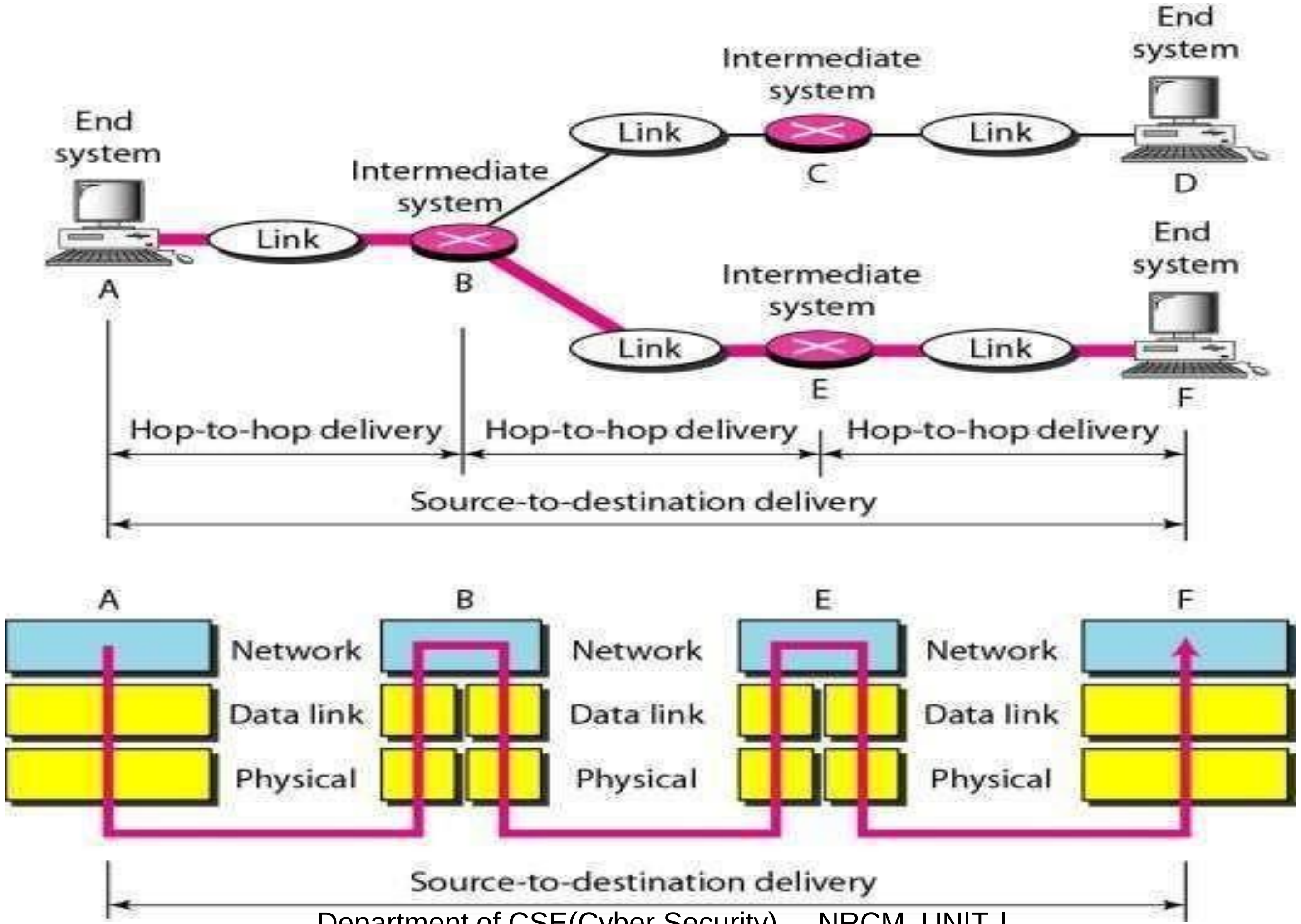


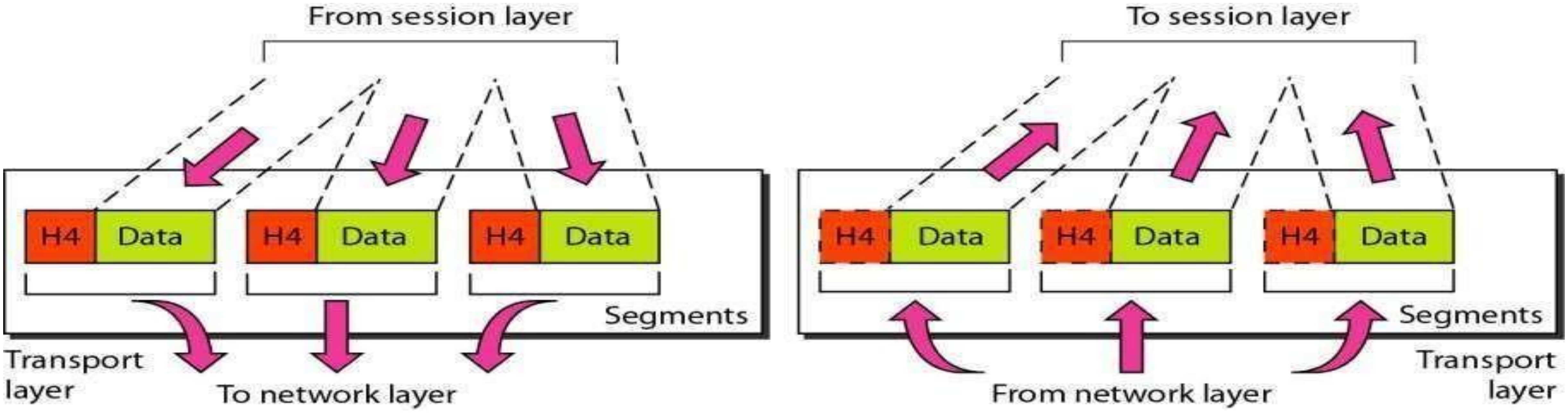
The network layer is responsible for the delivery of individual packets from the source host to the destination host.

Network Layer

- packets
- Logical addressing
- Routing

Source-to-destination delivery





The transport layer is responsible for the delivery of a message from one process to another.

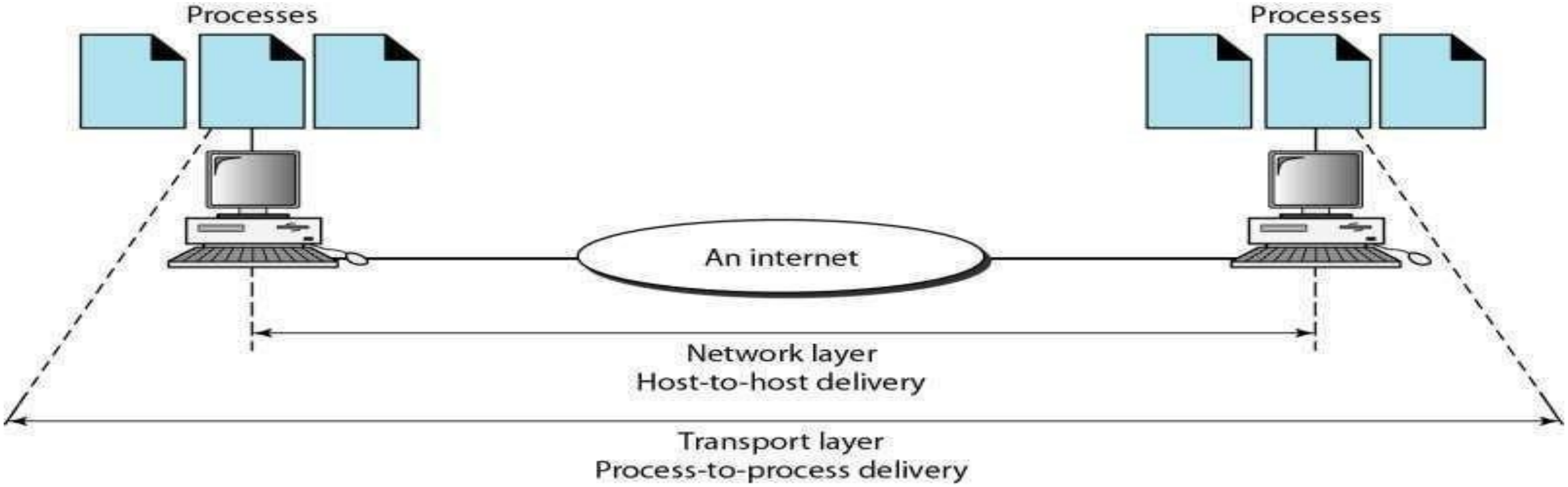
Transport Layer

- Service-point addressing.
- Segmentation and reassembly
- Connection control.
- Flow control
- Error control

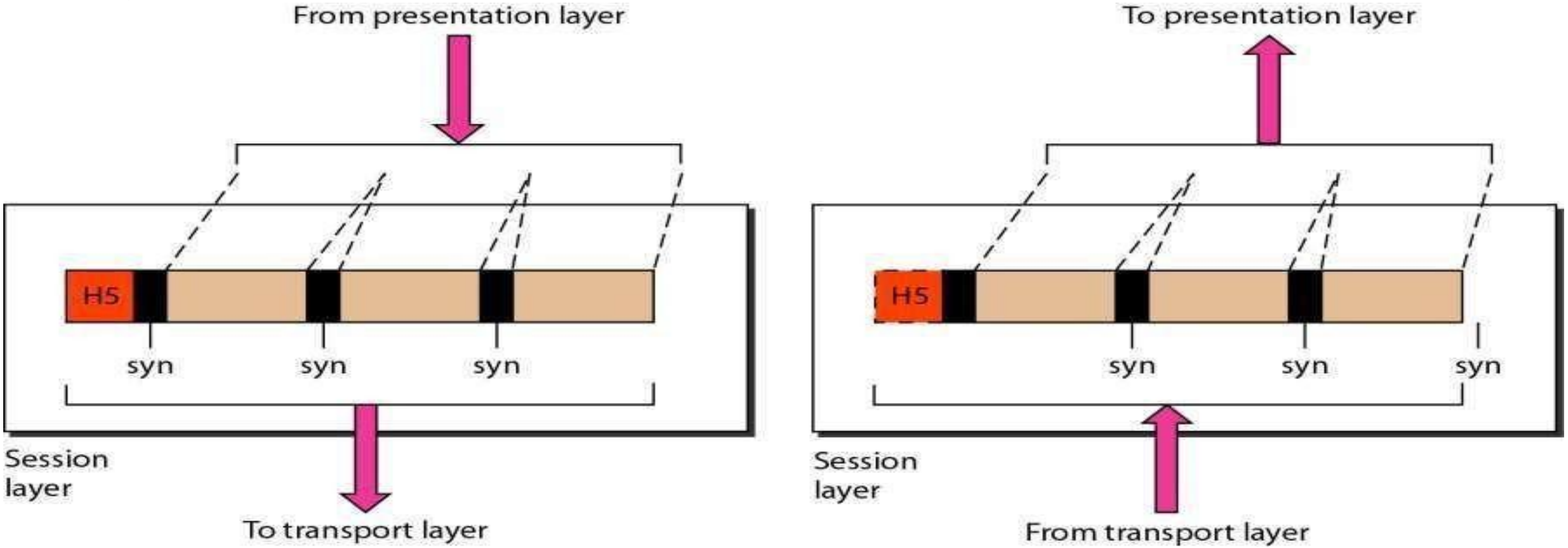
Transport Layer

- Service-point addressing.
- Segmentation and reassembly
- Connection control.
- Flow control
- Error control

Reliable process-to-process delivery of a message



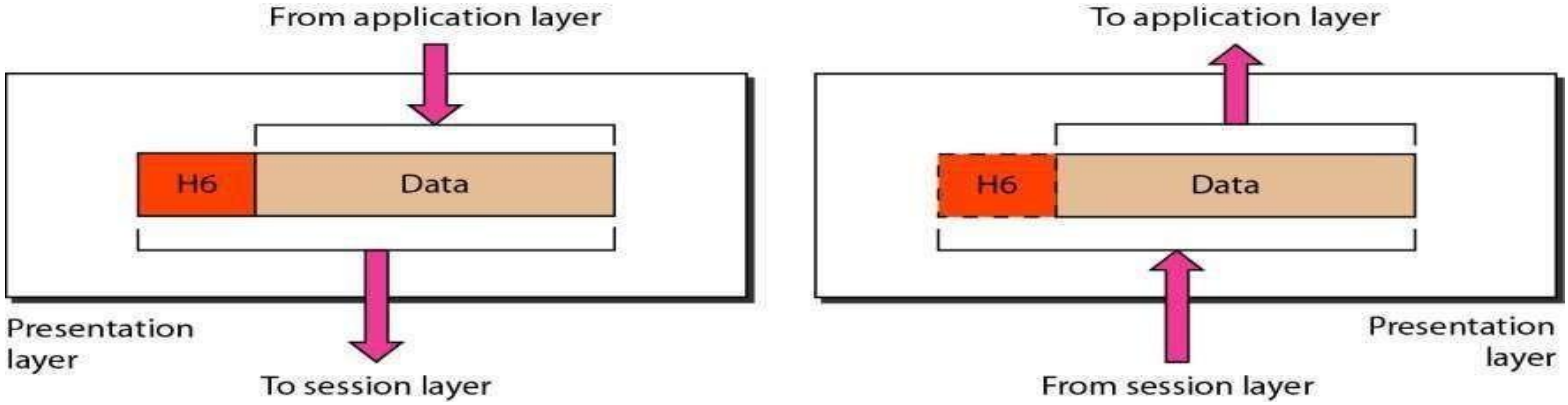
Session layer



The session layer is responsible for dialog control and synchronization.

Session Layer

- Dialog control
- Synchronization(maintaining check points)

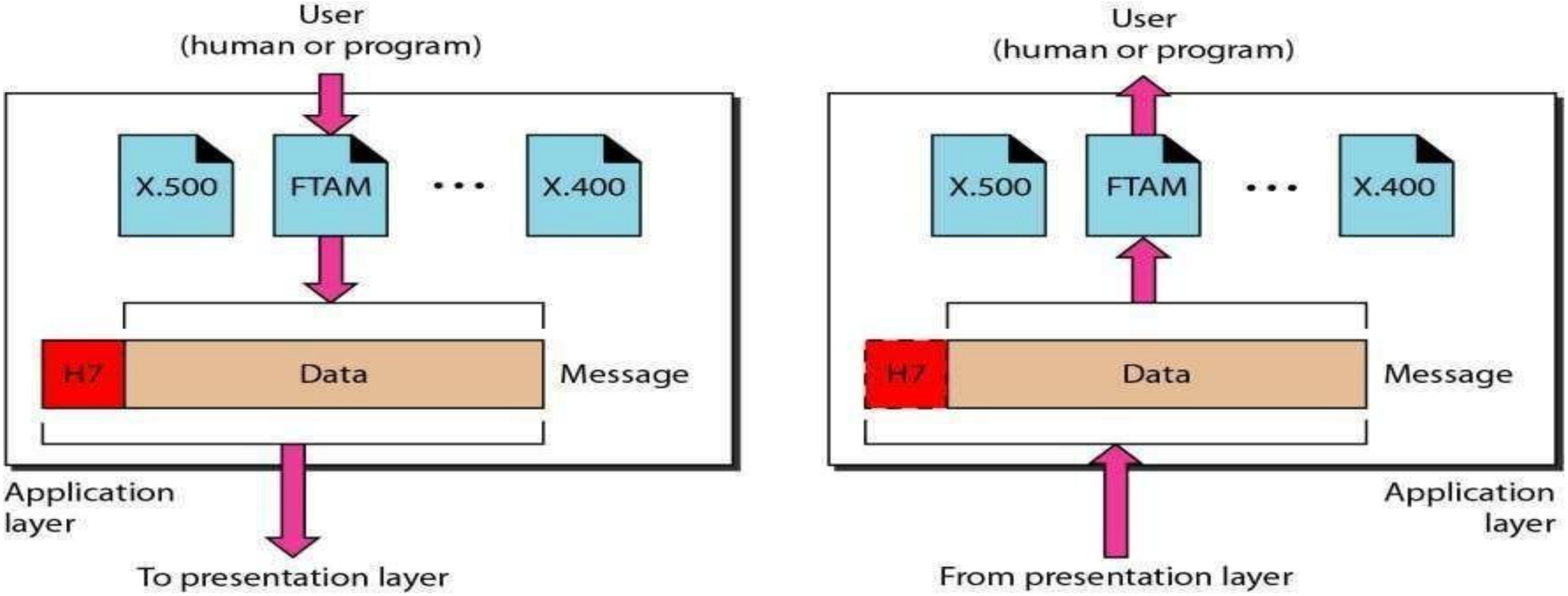


The presentation layer is responsible for translation, compression, and encryption.

Presentation Layer

- Translation.
- Encryption.
- Compression.

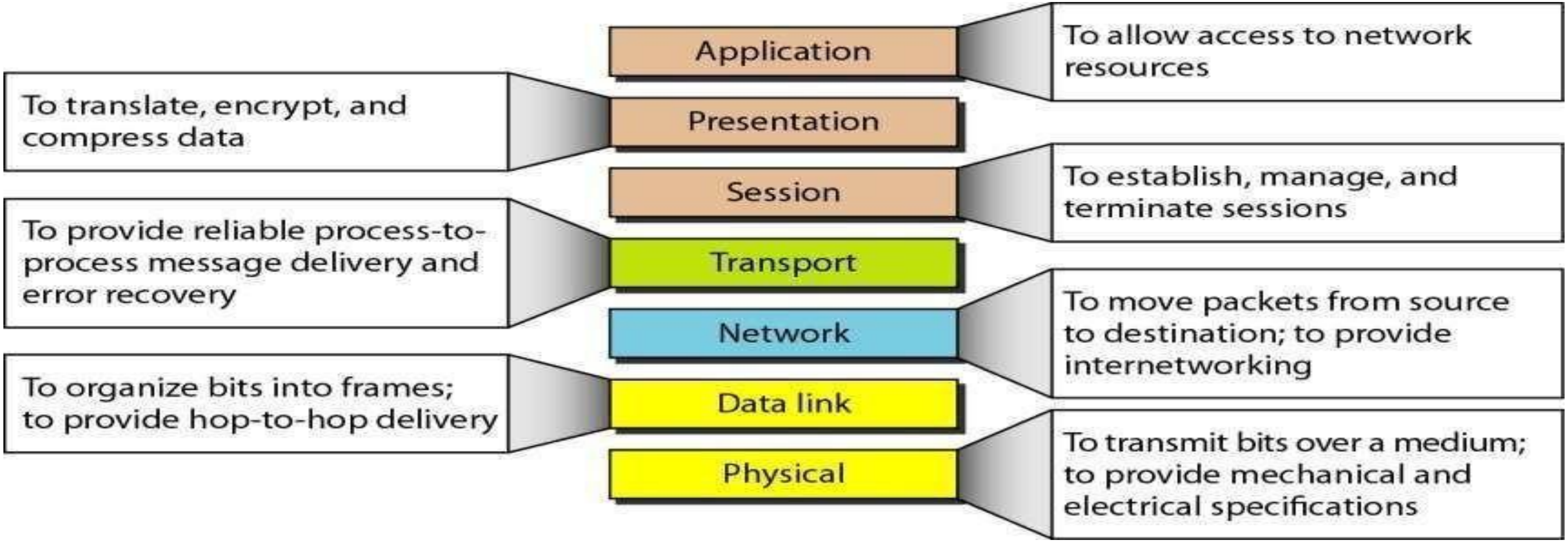
Application layer



Application Layer

- Network virtual terminal
- File transfer, access, and management.
- Mail services.
- Directory services.

Summary of layers



Application

To allow access to network resources

Presentation

To translate, encrypt, and compress data

Session

To establish, manage, and terminate sessions

Transport

To provide reliable process-to-process message delivery and error recovery

Network

To move packets from source to destination; to provide internetworking

Data link

To organize bits into frames; to provide hop-to-hop delivery

Physical

To transmit bits over a medium; to provide mechanical and electrical specifications

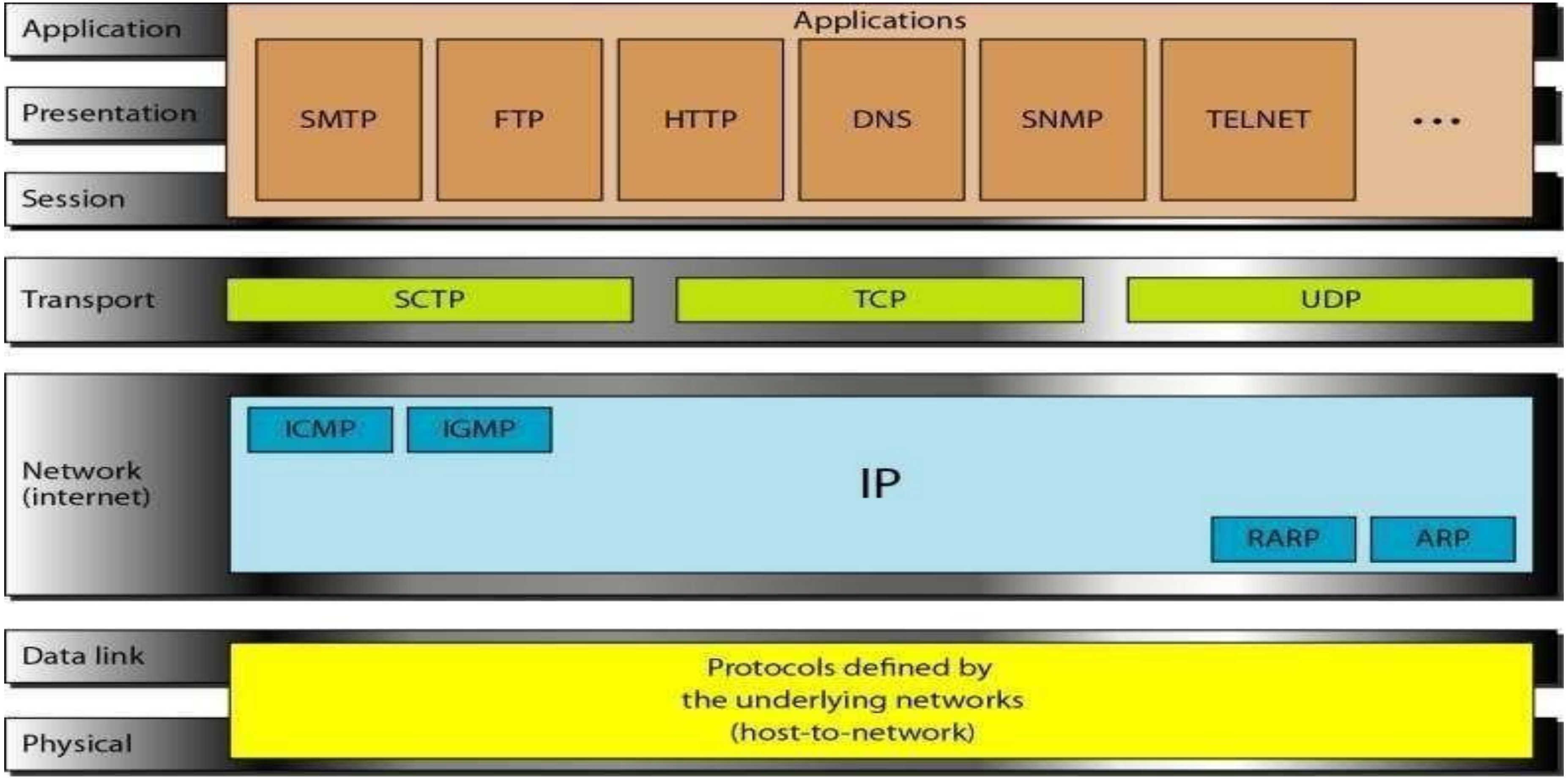
TCP/IP PROTOCOL SUITE

The layers in the TCP/IP protocol suite do not exactly match those in the OSI model.

4 layers in TCP/IP suite:

1. **Application layer**
2. **Transport layer**
3. **Internet layer**
4. **Host-to-network-layer**

TCP/IP and OSI model



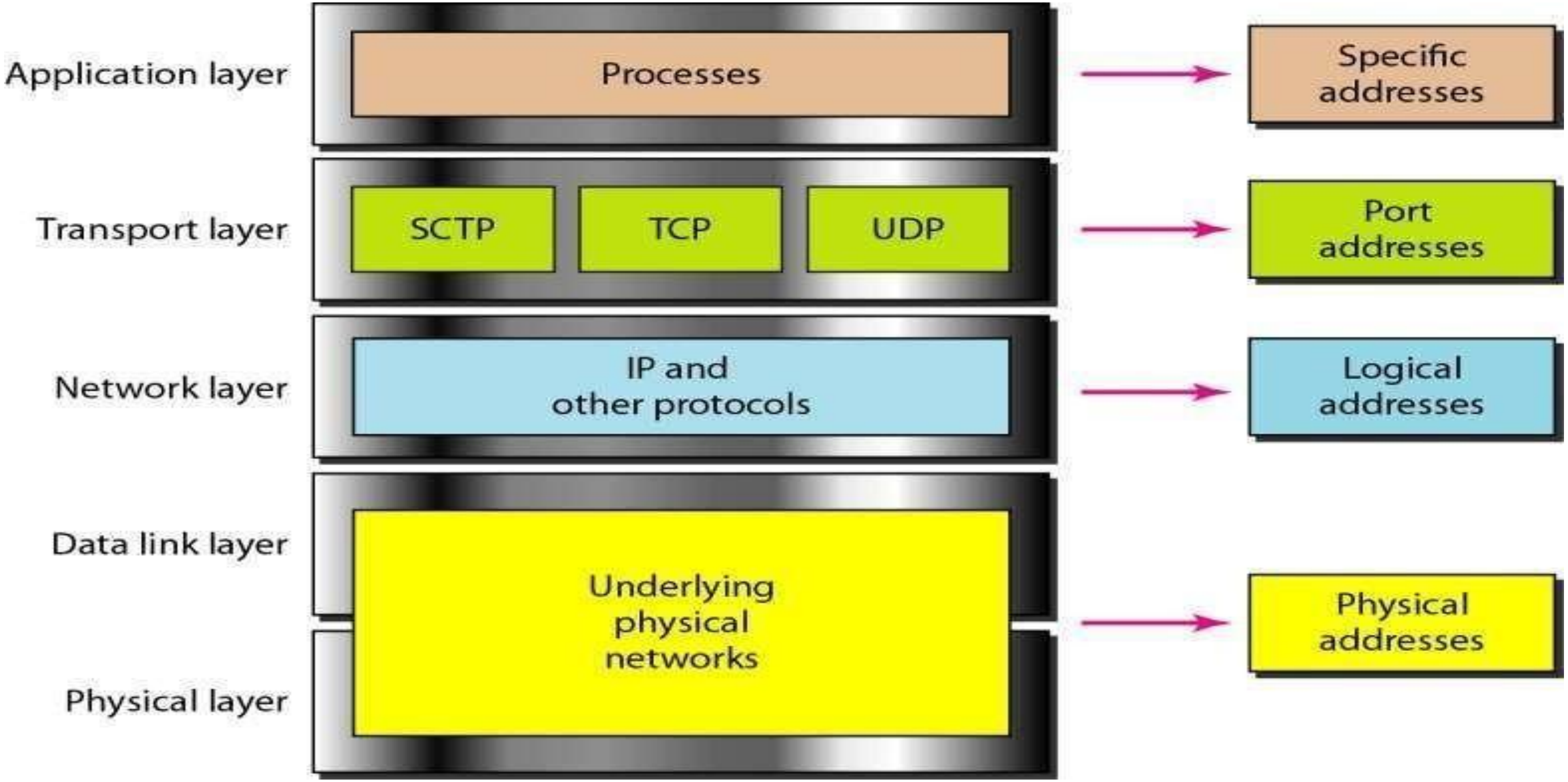
ADDRESSING

Four levels of addresses are used in an internet employing the TCP/IP protocols: physical, logical, port, and specific.

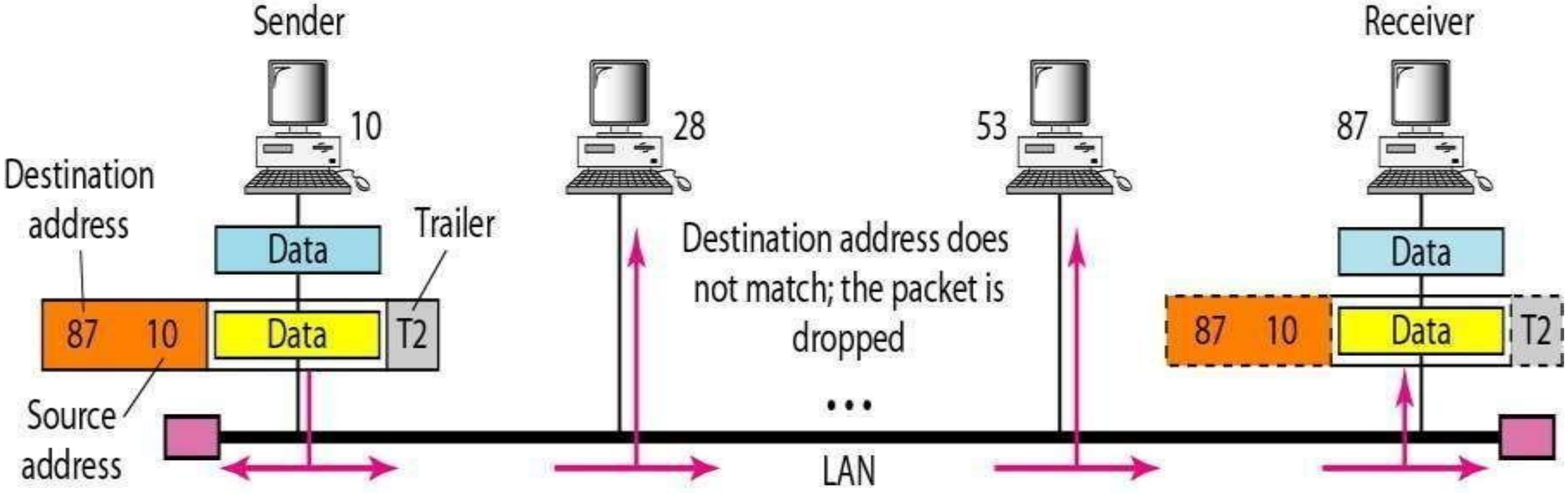
Topics discussed in this section:

Physical Addresses Logical Addresses Port Addresses Specific Addresses

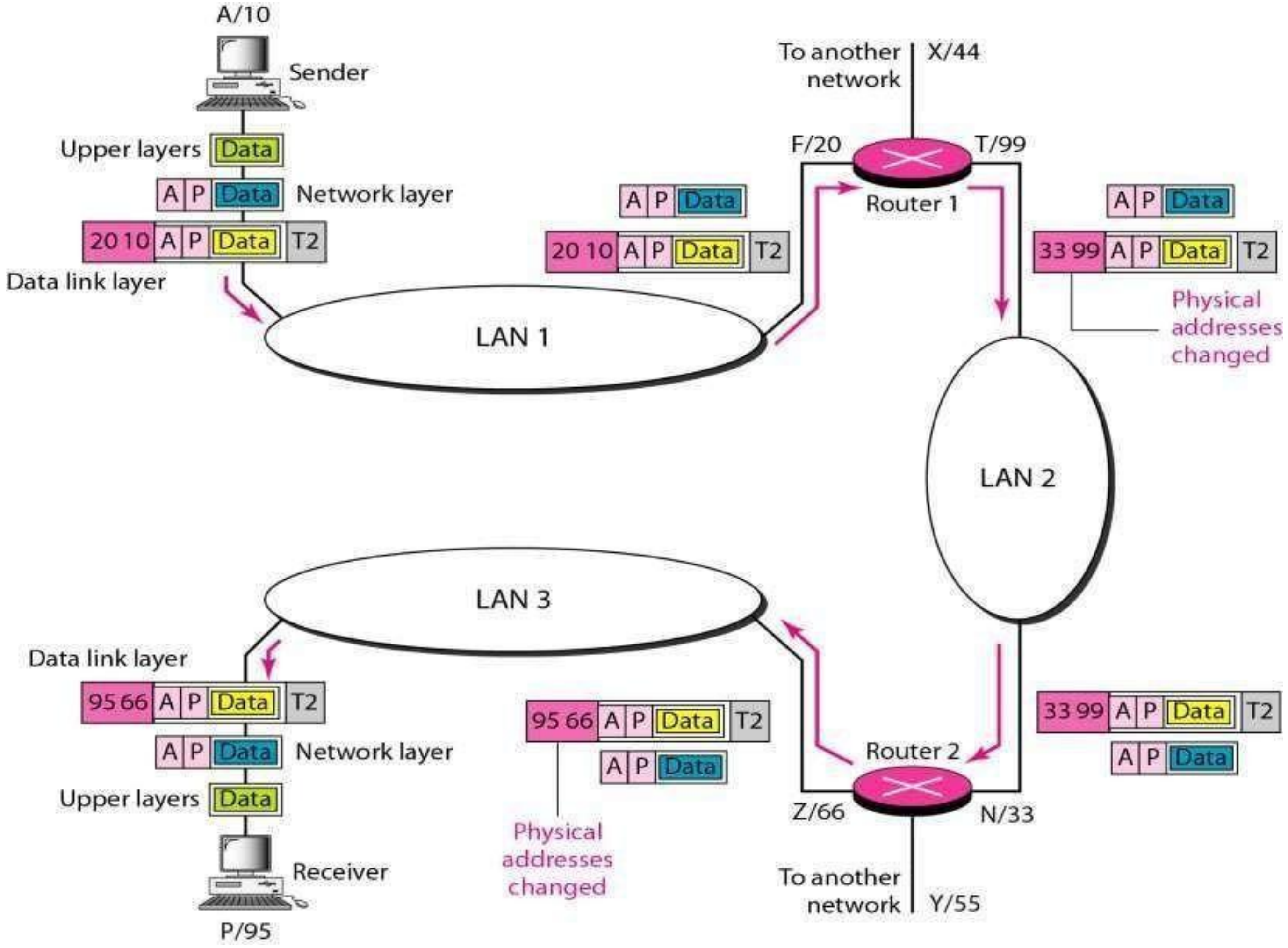
Figure 2.18 *Relationship of layers and addresses in TCP/IP*



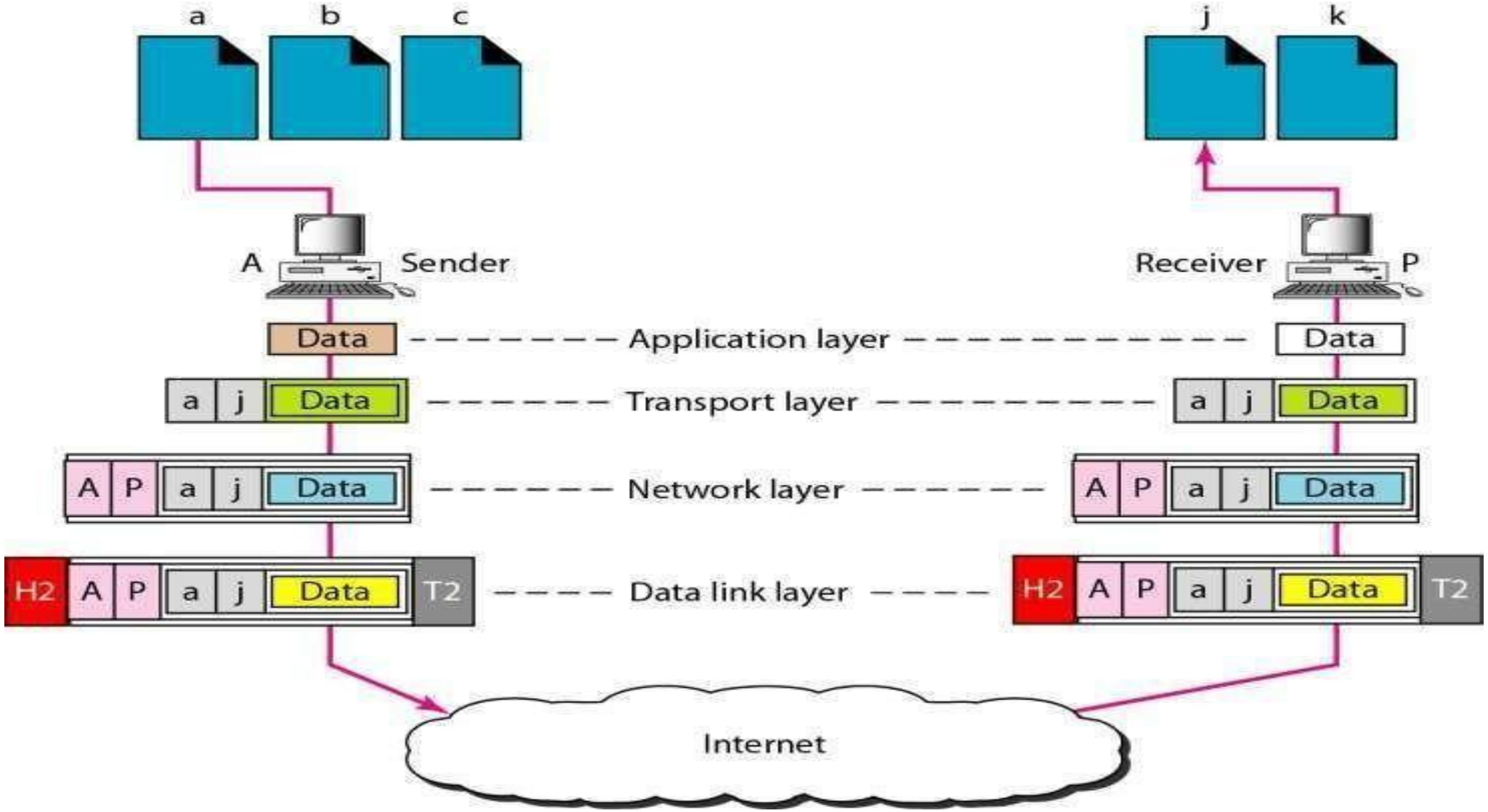
Physical addresses



IP addresses

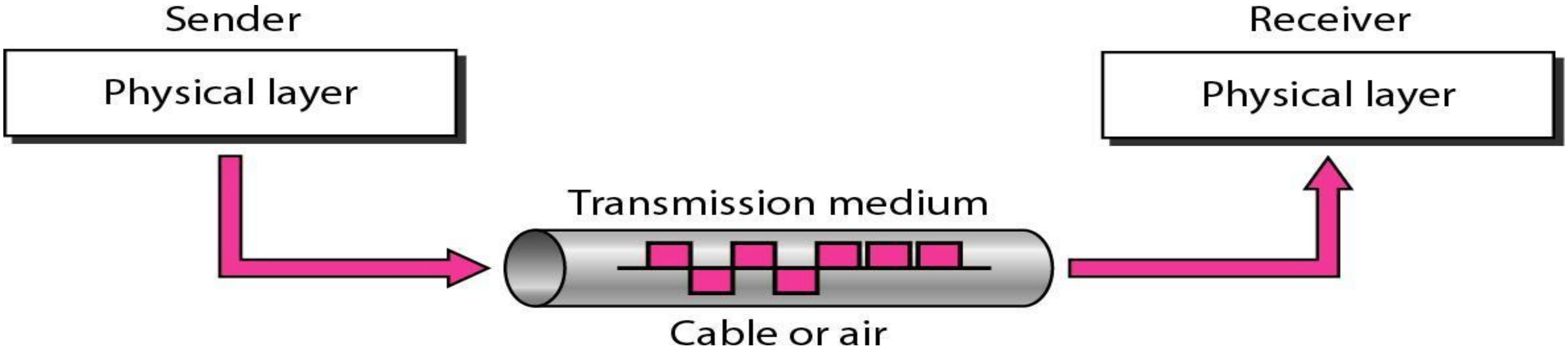


Port addresses

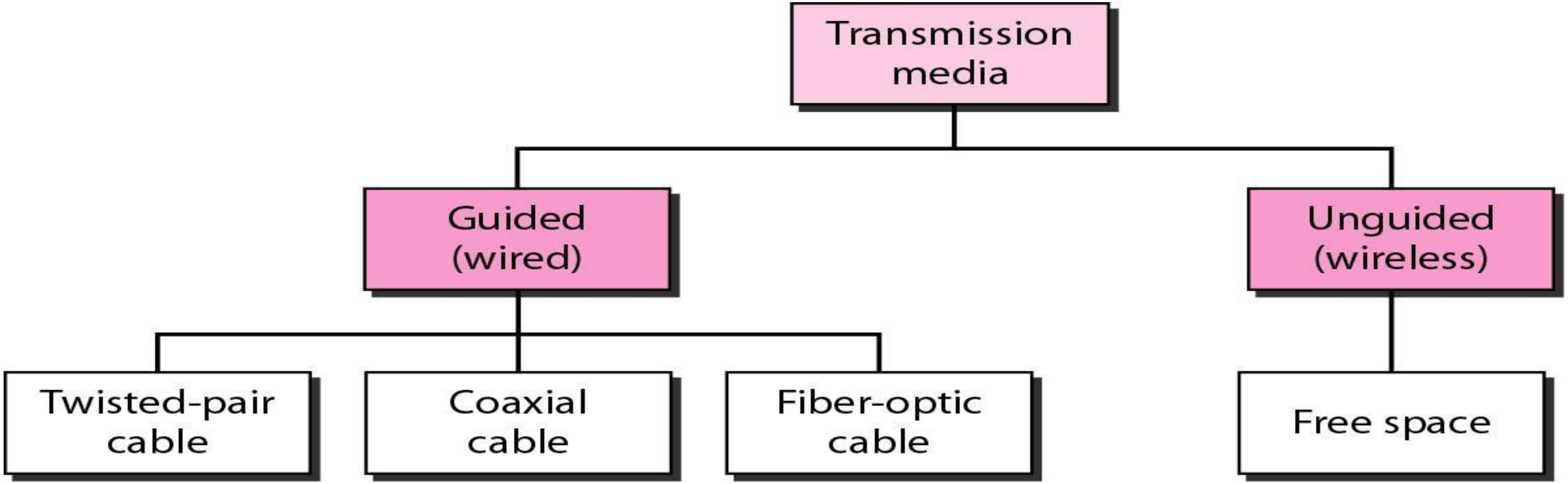


Transmission Media

A transmission medium can be broadly defined as anything that can carry information from a source to a destination

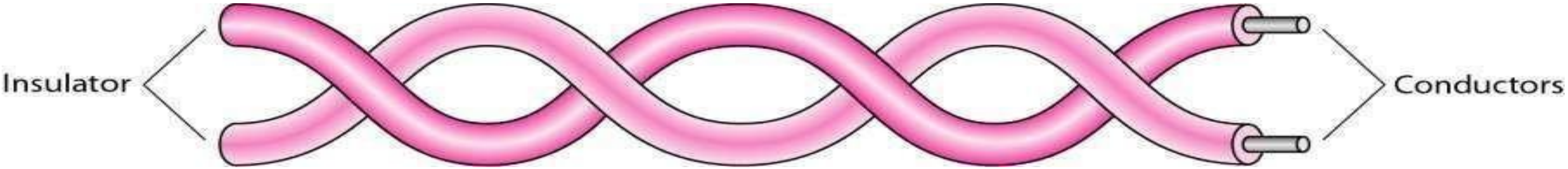


Classes of transmission media



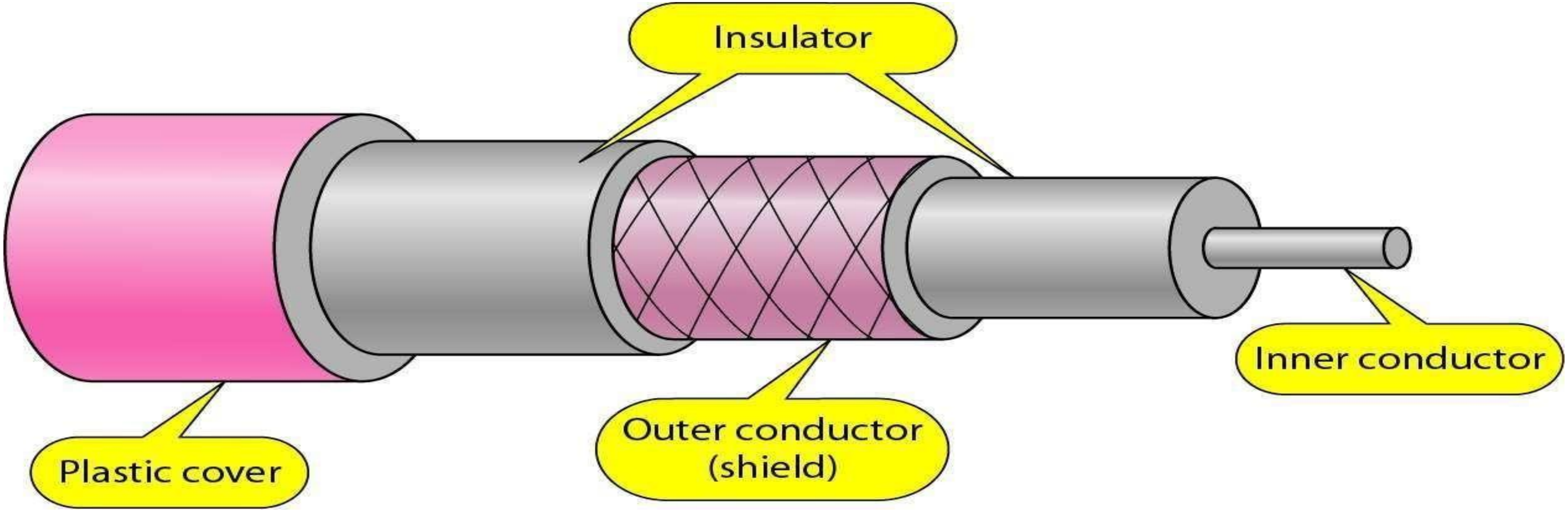
Twisted-Pair Cable

A twisted pair consists of two conductors (normally copper), each with its own plastic insulation, twisted together.



Coaxial Cable

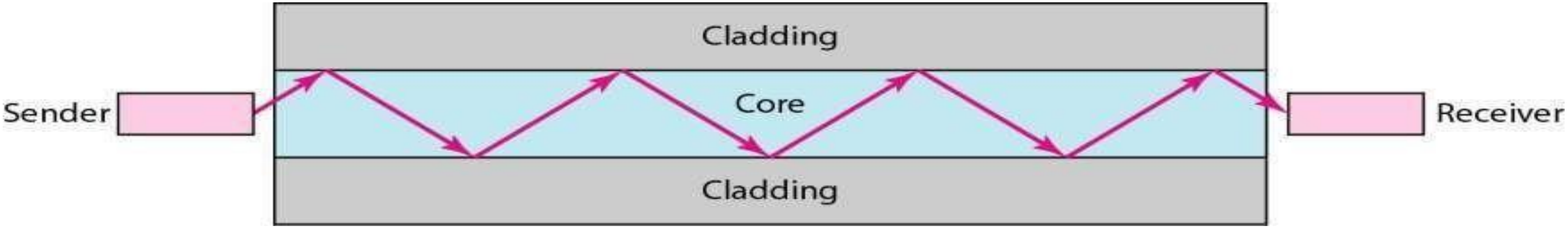
Coaxial cable (or *coax)* carries signals of higher frequency ranges than those in twisted-pair cable.



Fiber-Optic Cable

A fiber-optic cable is made of glass or plastic and transmits signals in the form of light.

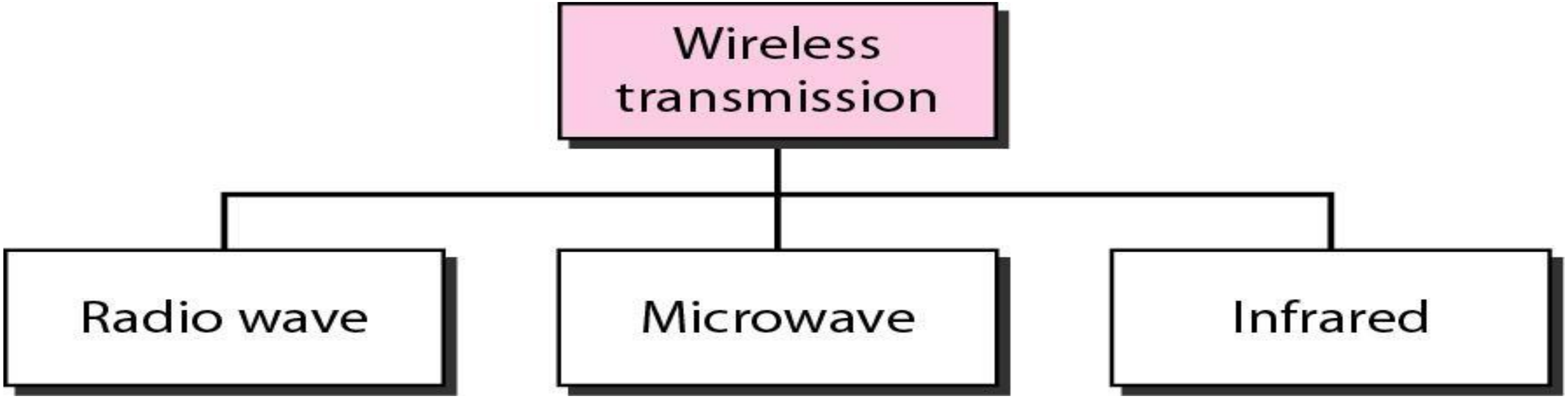
Transmission of lightray



Applications

- **Twisted-pair cable** is used for voice and data communications
- **Coaxial cable** is used in cable TV networks and traditional Ethernet LANs.
- **Fiber-optic cable** is used in backbone networks, cable TV networks, and Fast Ethernet networks

Wireless transmission waves



Radio Waves

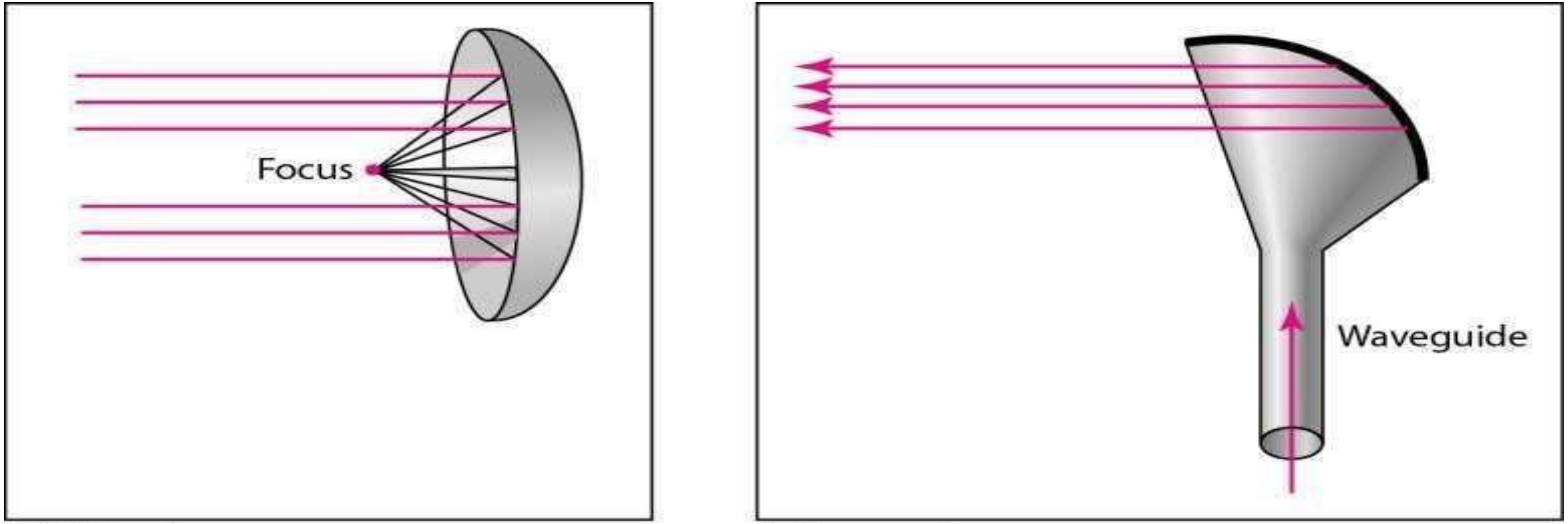
waves ranging in frequencies between 3 kHz and 1 GHz are normally called **radio waves**

Radio waves are used for multicast communications, such as radio and television, and paging systems.

Microwaves

Electromagnetic waves having frequencies between 1 and 300 GHz are called **micro-waves** Microwaves are used for **uncast communication** such as cellular telephones, satellite networks, and wireless LANs.

Unidirectional antennas



a. Dish antenna

b. Horn antenna

II UNIT

DATA LINK LAYER

Data Link Layer

- Responsibilities
  - Framing
  - Addressing
  - Flow control
  - Media Access Control
  - Error control
    - Error detection
    - Error correction

Data Link Layer

- Responsibilities
  - Framing
  - Addressing
  - Flow control
  - Media Access Control
  - Error control
    - Error detection
    - Error correction

Note

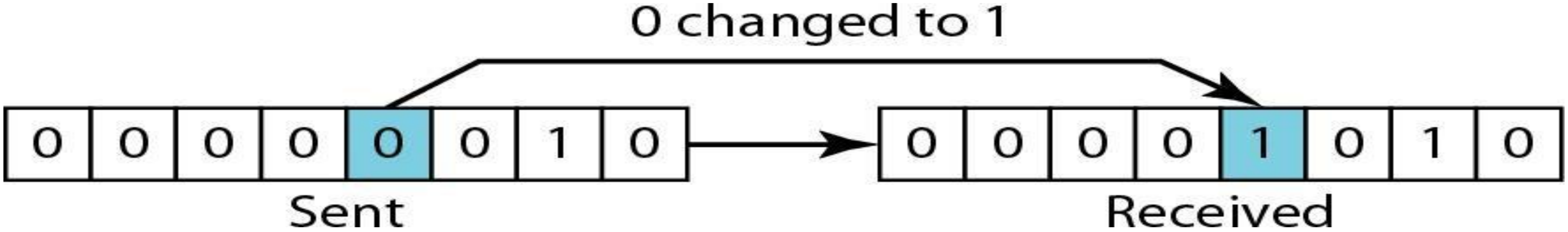
Data can be corrupted during transmission.

Some applications require that errors be detected and corrected.

Note

In a single-bit error, only 1 bit in the data unit has changed.

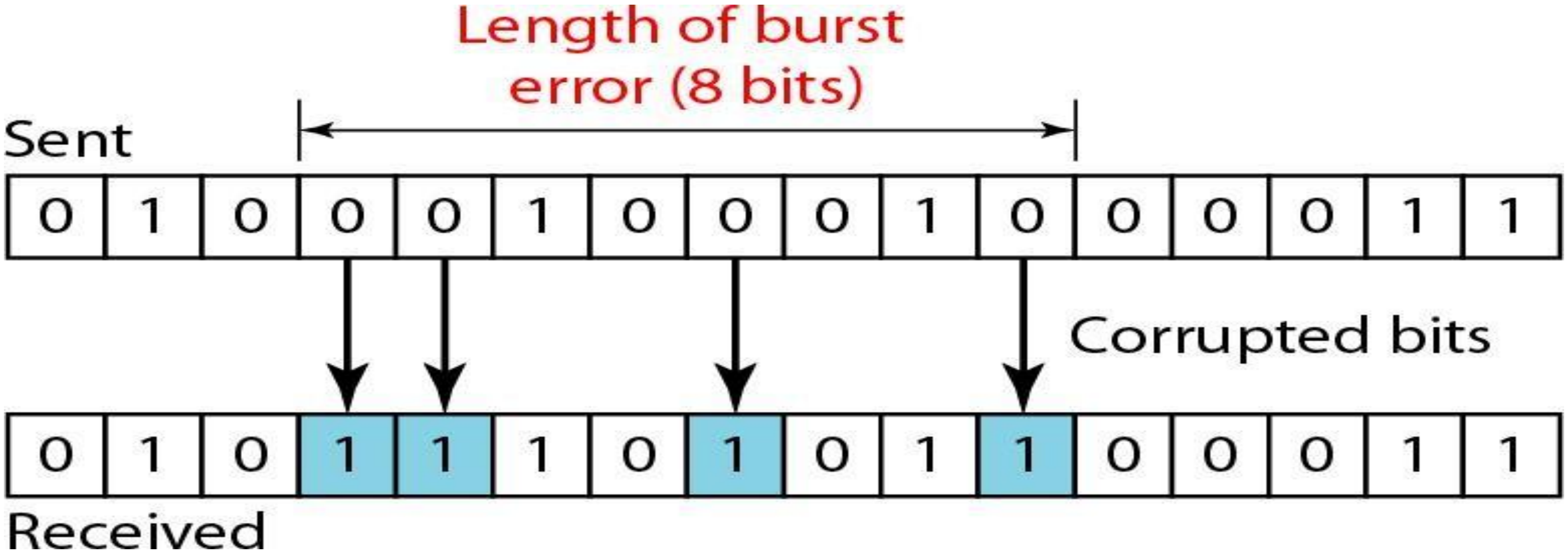
Single-bit error



Note

A **burst error** means that 2 or more bits in the data unit have changed.

Burst error of length 8

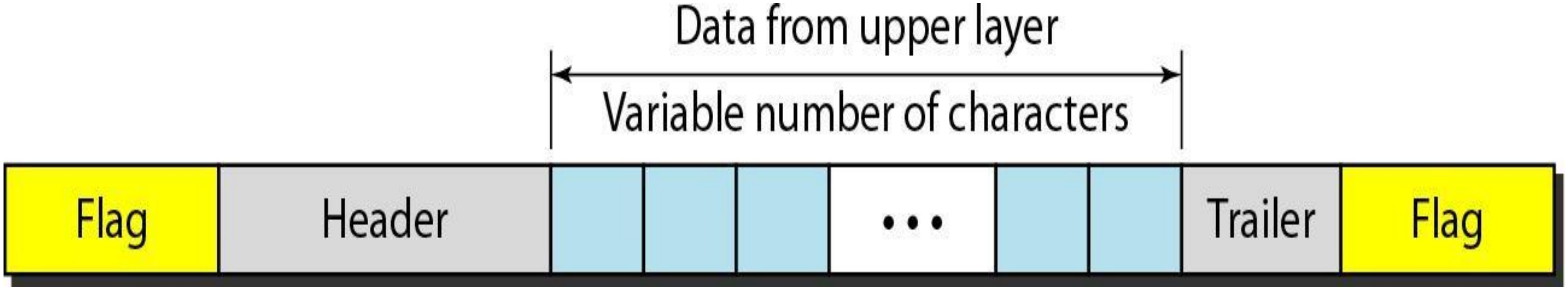


Sent	0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	1
Received	0	1	0	1	1	1	0	1	0	1	1	0	0	0	1	1

Framing

-character oriented framing

Figure ***Aframe in a character-oriented protocol***



Note

Byte stuffing is the process of adding 1 extra byte whenever there is a flag or escape character in the text.

Figure *Character stuffing* or *Byte stuffing*

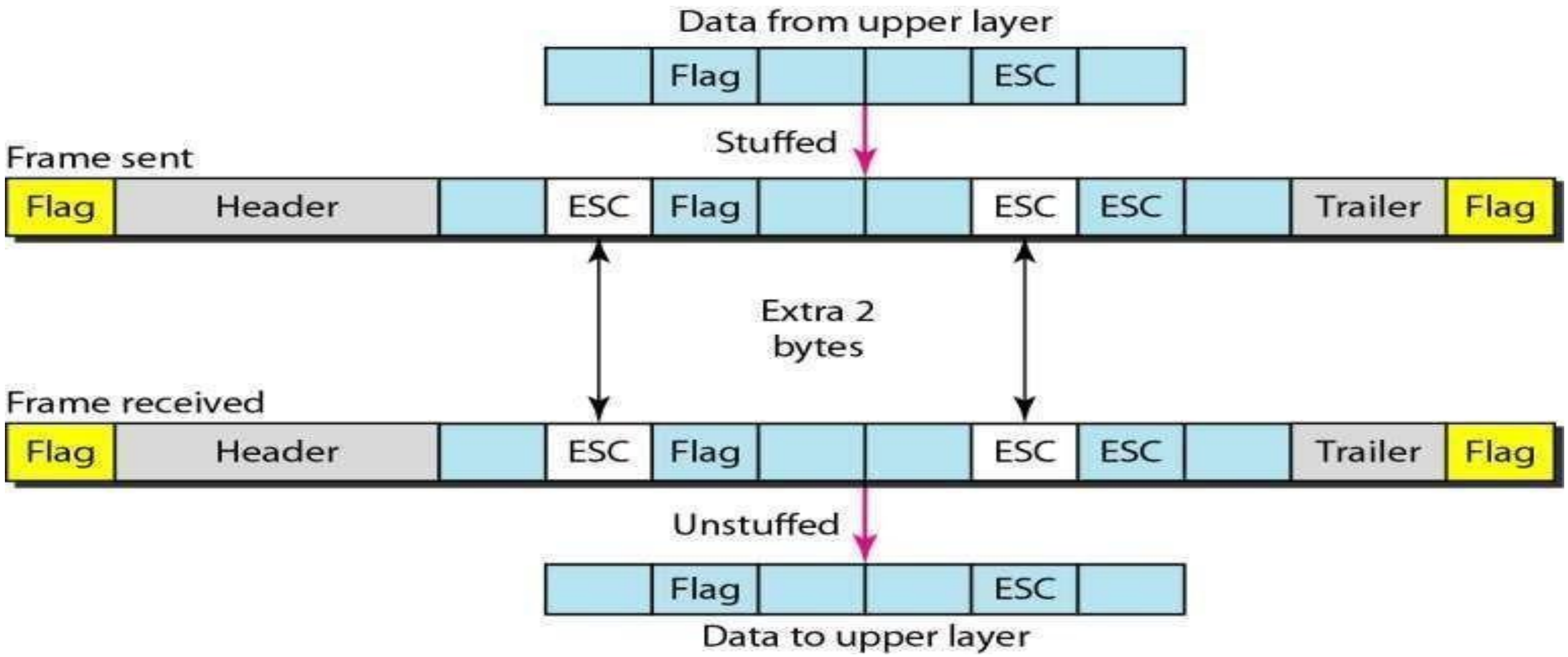
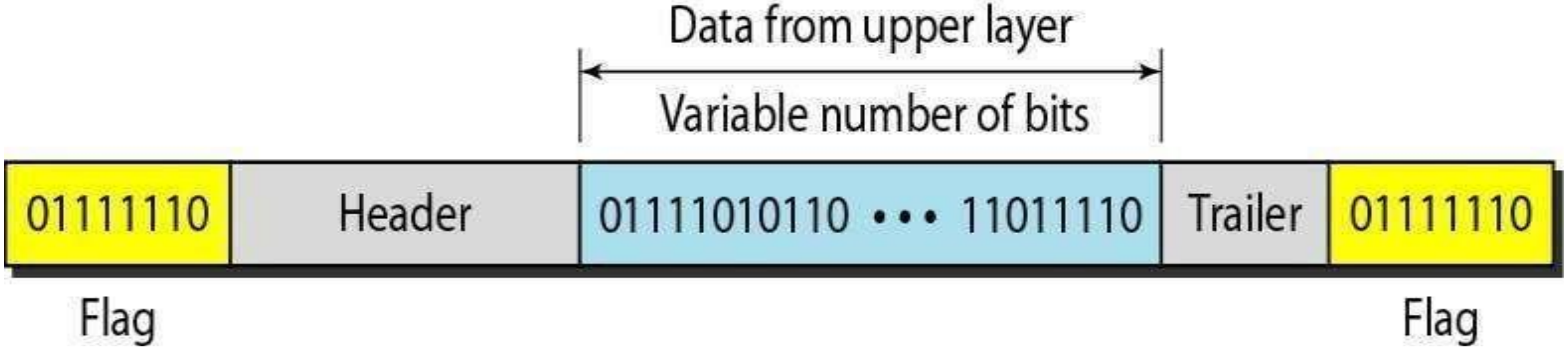


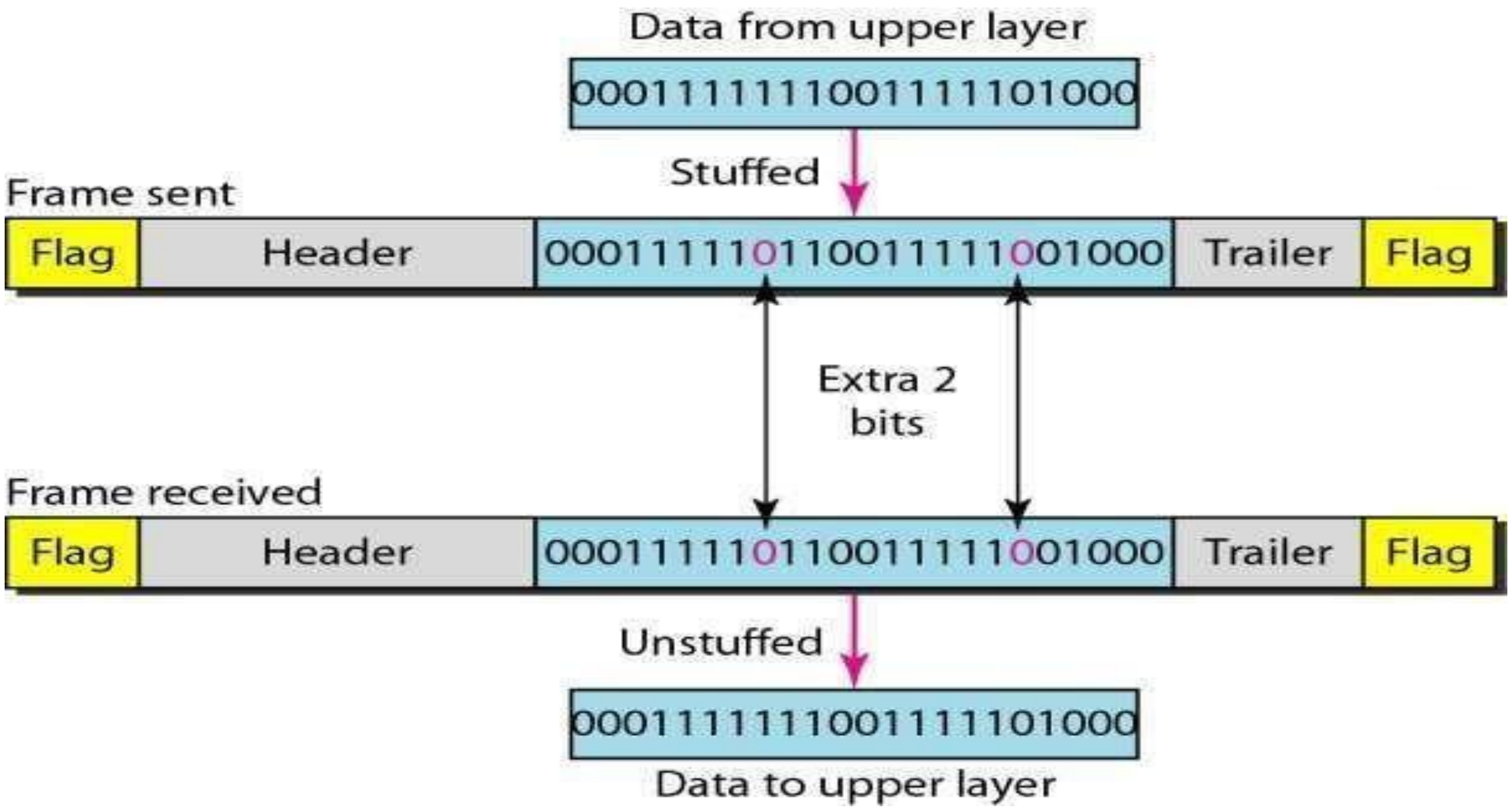
Figure *Aframe in a bit-oriented protocol*



Note

Bit stuffing is the process of adding one extra 0 whenever five consecutive 1s follow a 0 in the data, so that the receiver does not mistake the pattern 0111110 for a flag.

Figure ***Bit stuffing and unstuffing***



Note

To detect or correct errors, we need to send extra (redundant) bits with data.

Figure *Datawords and codewords in block coding*

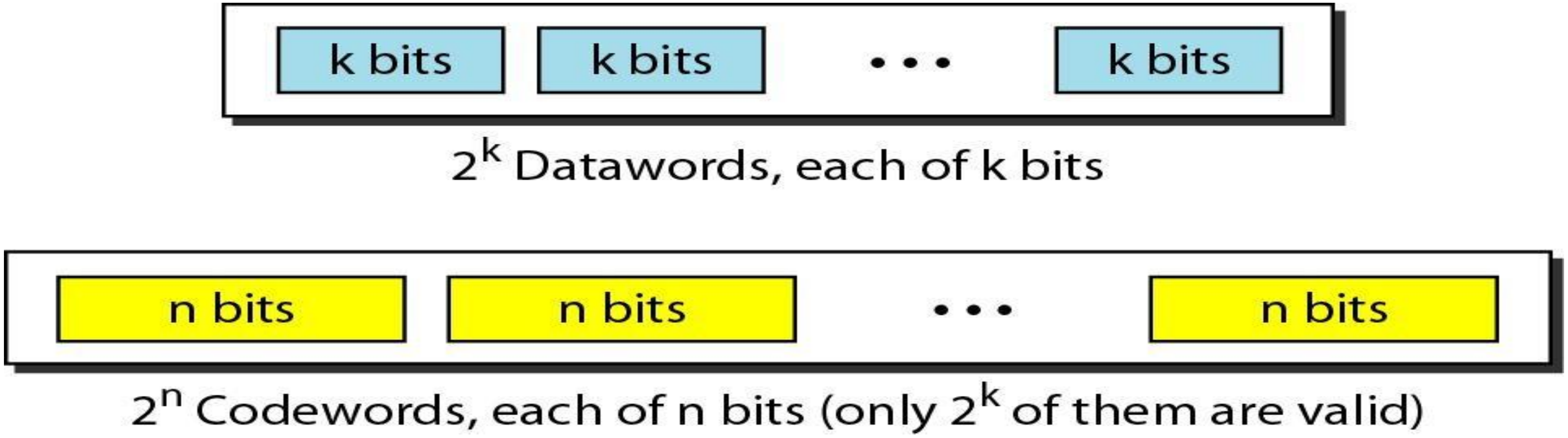

k bits
k bits
• • •
k bits
2^k Datawords, each of k bits
n bits
n bits
• • •
n bits
2^n Codewords, each of n bits (only 2^k of them are valid)

Figure *The structure of encoder and decoder*

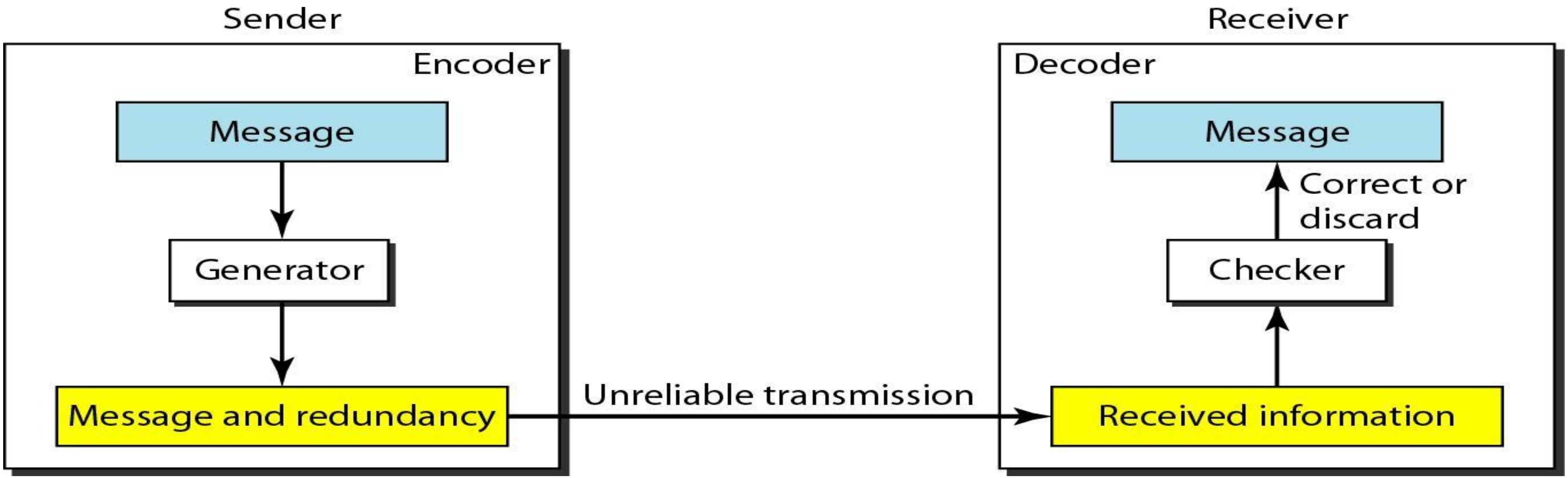
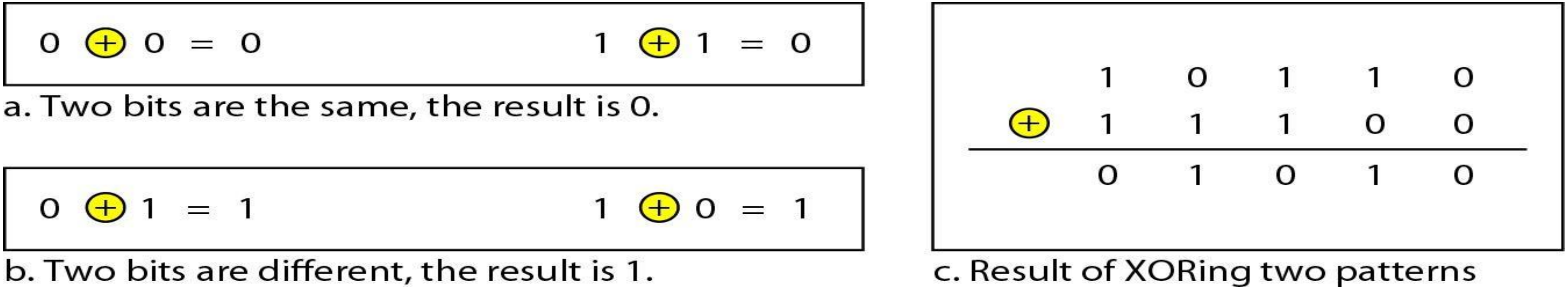


Figure *XORing of two single bits or two words*



$0 \oplus 0 = 0$ $\quad$ $1 \oplus 1 = 0$

a. Two bits are the same, the result is 0.

$0 \oplus 1 = 1$ $\quad$ $1 \oplus 0 = 1$

b. Two bits are different, the result is 1.

$$\begin{array}{r ccccc} & 1 & 0 & 1 & 1 & 0 \\ \oplus & 1 & 1 & 1 & 0 & 0 \\ \hline & 0 & 1 & 0 & 1 & 0 \end{array}$$

c. Result of XORing two patterns

Error Detection

- Enough redundancy is added to detect an error.
- The receiver knows an error occurred but does not know which bit(s) is(are) in error.
- Has less overhead than error correction.

CYCLIC CODES

***Cyclic codes* are special linear block codes with one extra**

y

$$b_1 = a_0 \quad b_2 = a_1 \quad b_3 = a_2 \quad b_4 = a_3 \quad b_5 = a_4 \quad b_6 = a_5 \quad b_0 = a_6$$

"Cyclic Redundancy check"

Figure *CRC encoder and decoder*

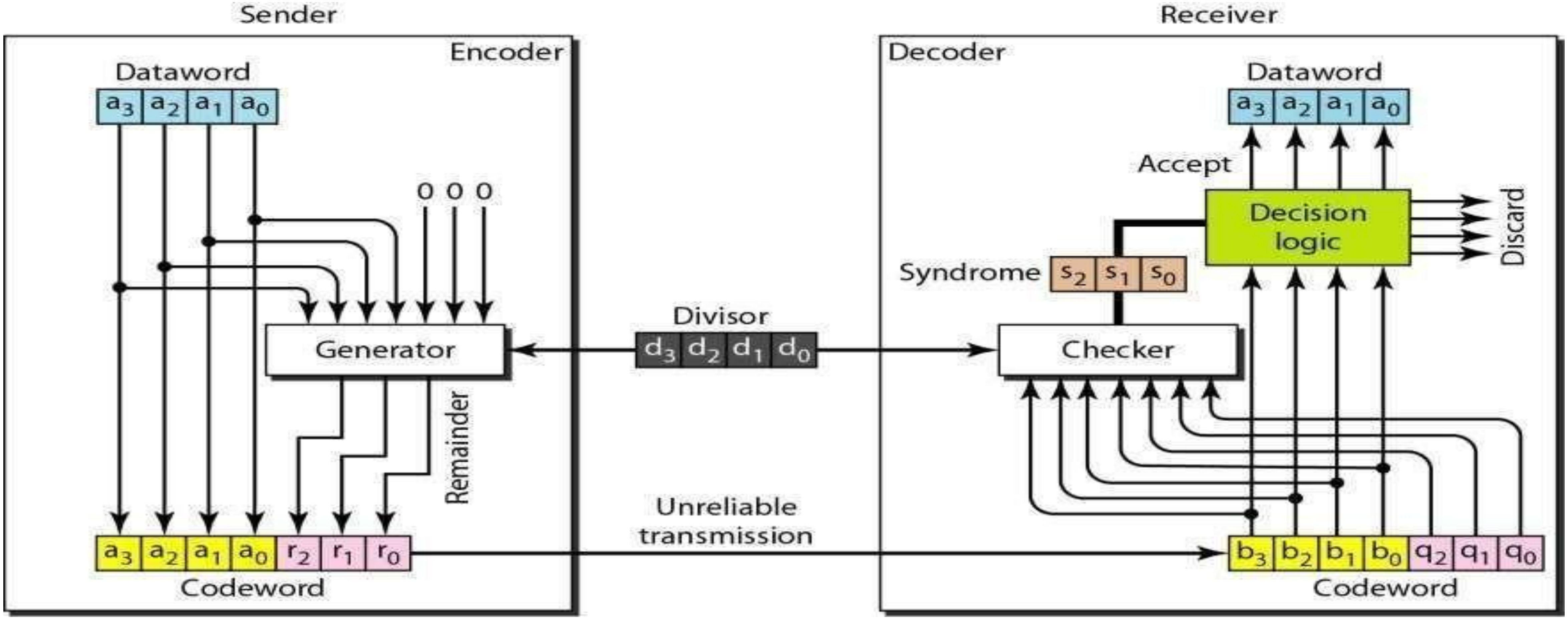


Figure *Division in CRC encoder*

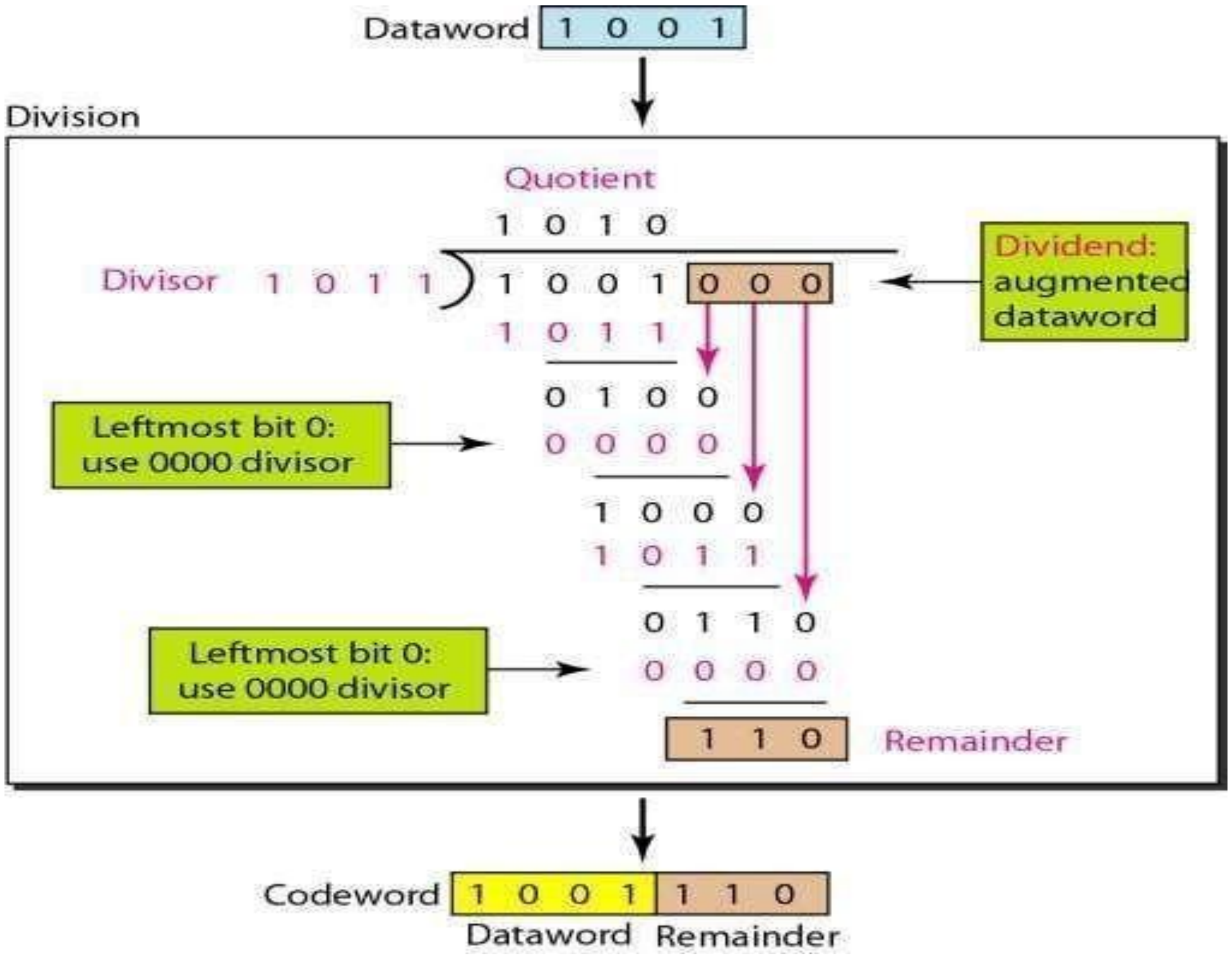


Figure *Division in the CRC decoder for two cases*

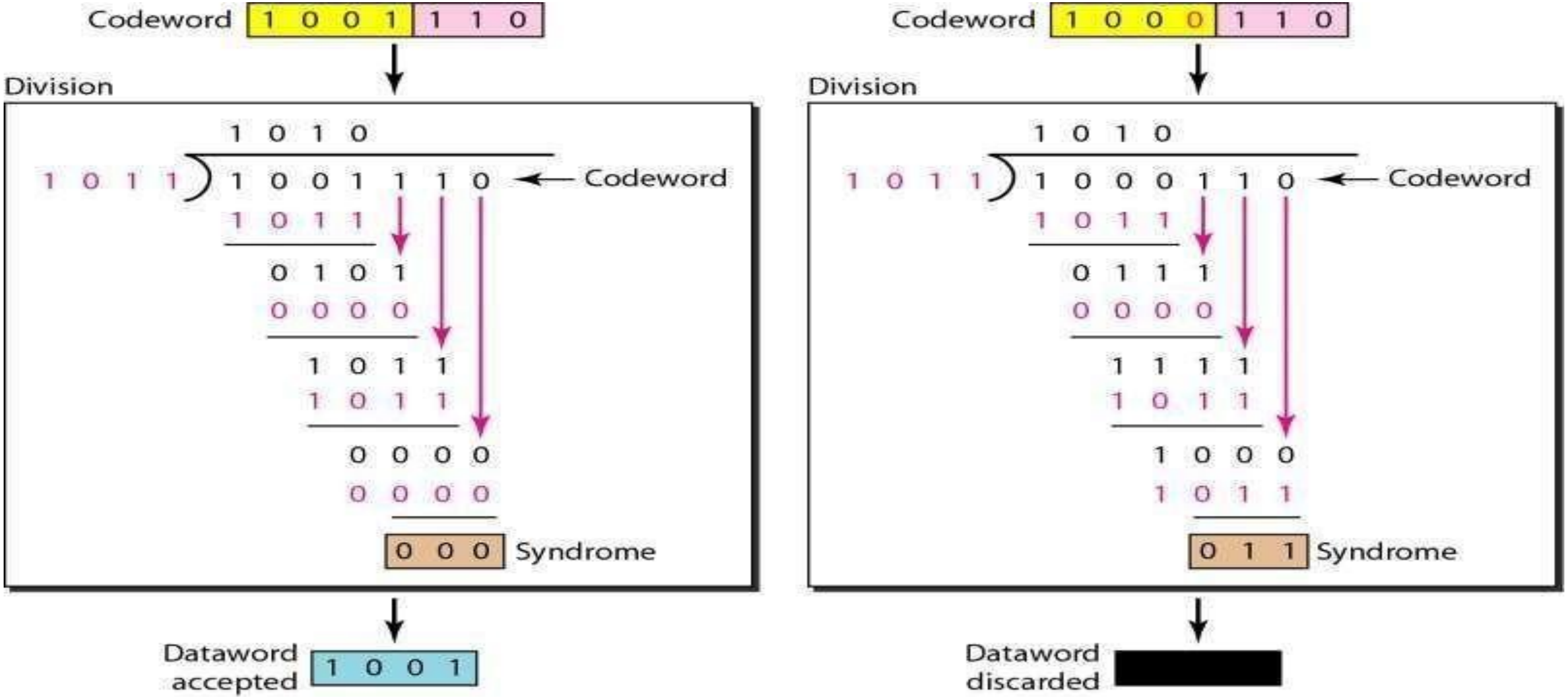
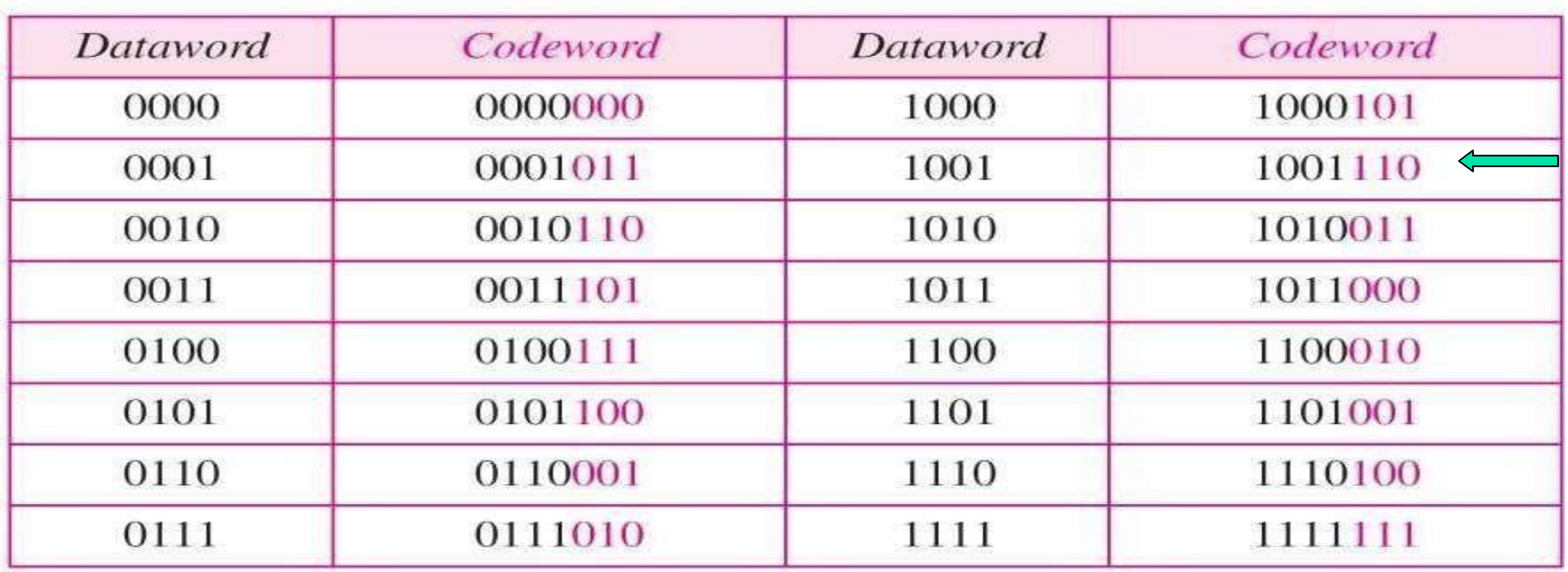


Table 10.6 *A CRC code with C(7, 4)*

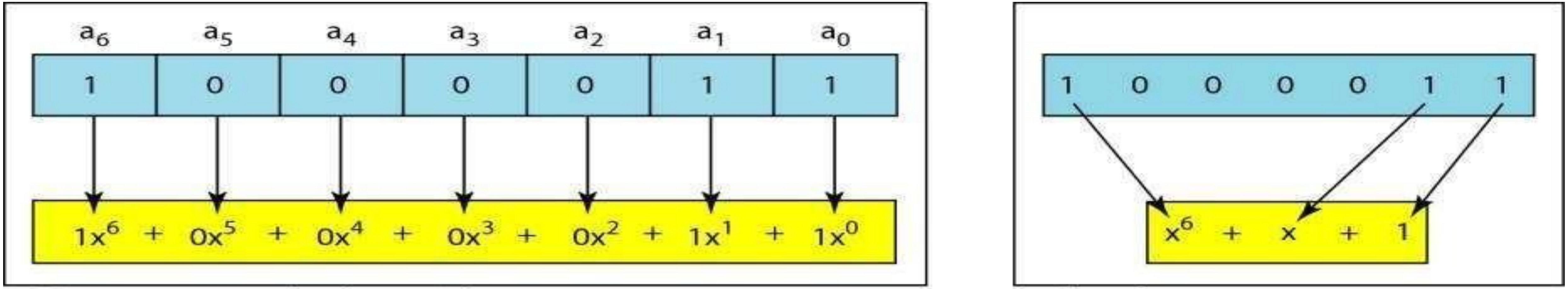


Dataword	Codeword	Dataword	Codeword
0000	0000000	1000	1000101
0001	0001011	1001	1001110
0010	0010110	1010	1010011
0011	0011101	1011	1011000
0100	0100111	1100	1100010
0101	0101100	1101	1101001
0110	0110001	1110	1110100
0111	0111010	1111	1111111

Using Polynomials

- We can use a polynomial to represent a binary word.
- Each bit from right to left is mapped onto a power term.
- The rightmost bit represents the “0” power term. The bit next to it the “1” power term, etc.
- If the bit is of value zero, the power term is deleted from the expression.

Figure *A polynomial to represent a binary word*

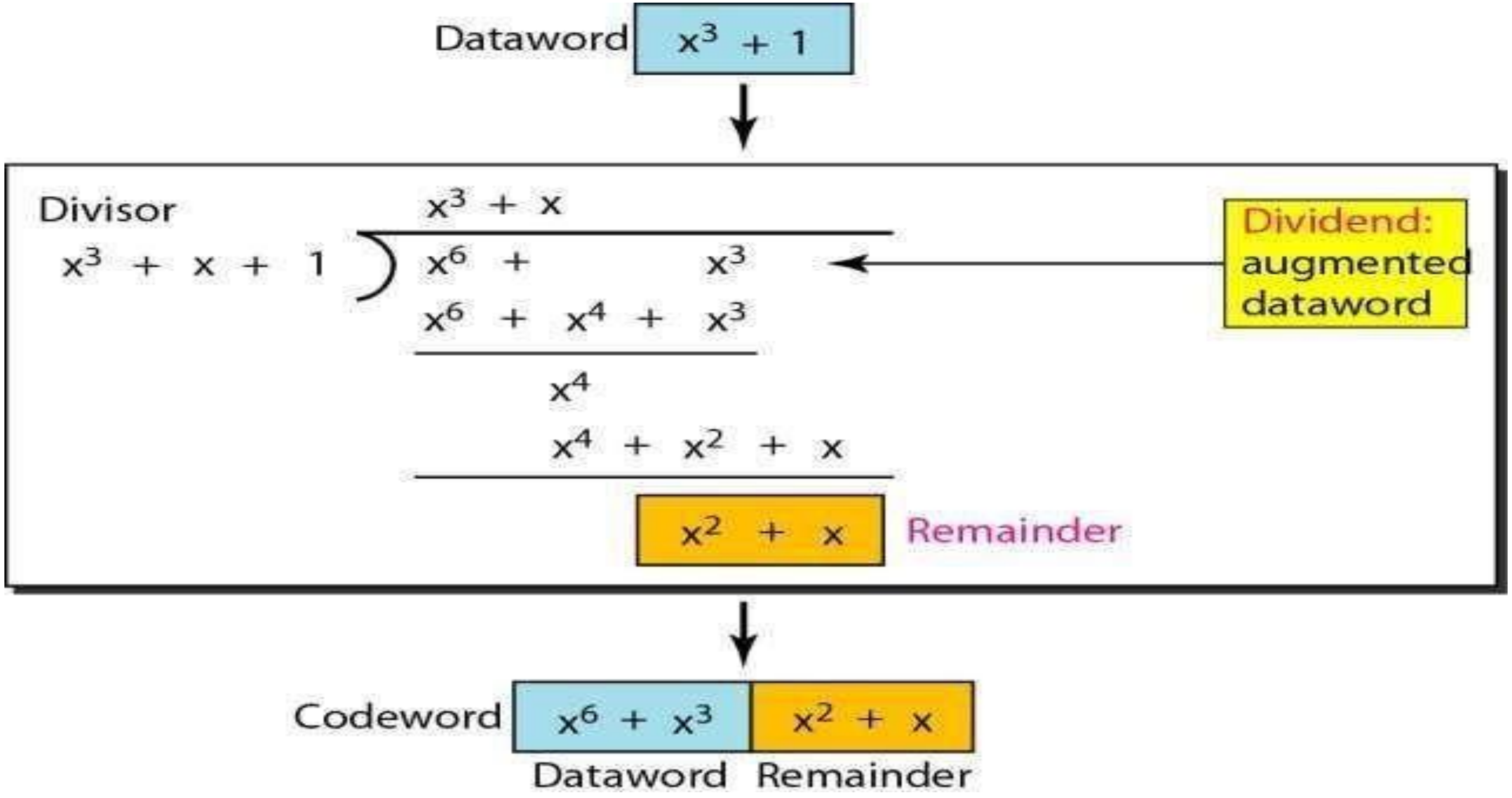


a_6 a_5 a_4 a_3 a_2 a_1 a_0
1 0 0 0 0 1 1
$1x^6 + 0x^5 + 0x^4 + 0x^3 + 0x^2 + 1x^1 + 1x^0$
1 0 0 0 0 1 1
$x^6 + x + 1$

a. Binary pattern and polynomial

b. Short form

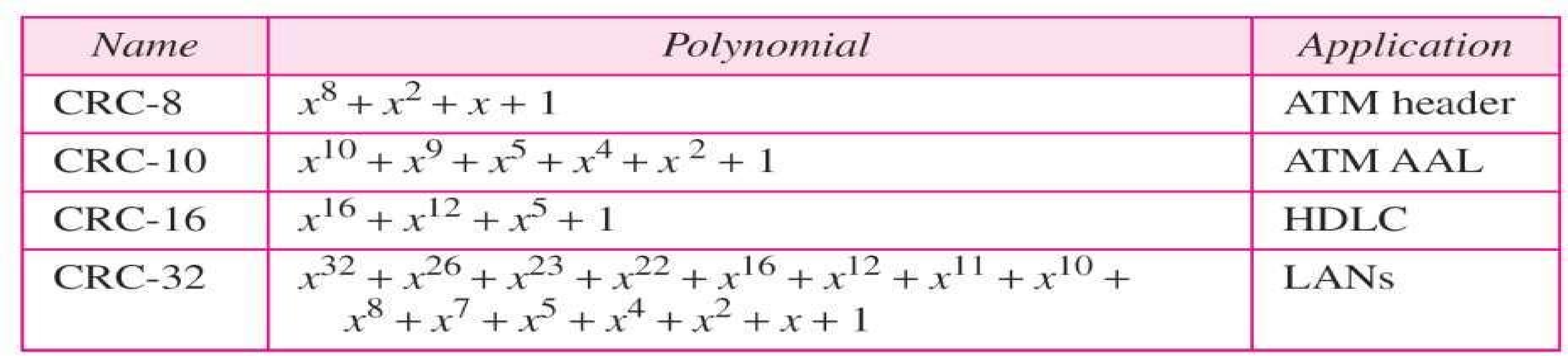
Figure 10.22 *CRC division using polynomials*



Note

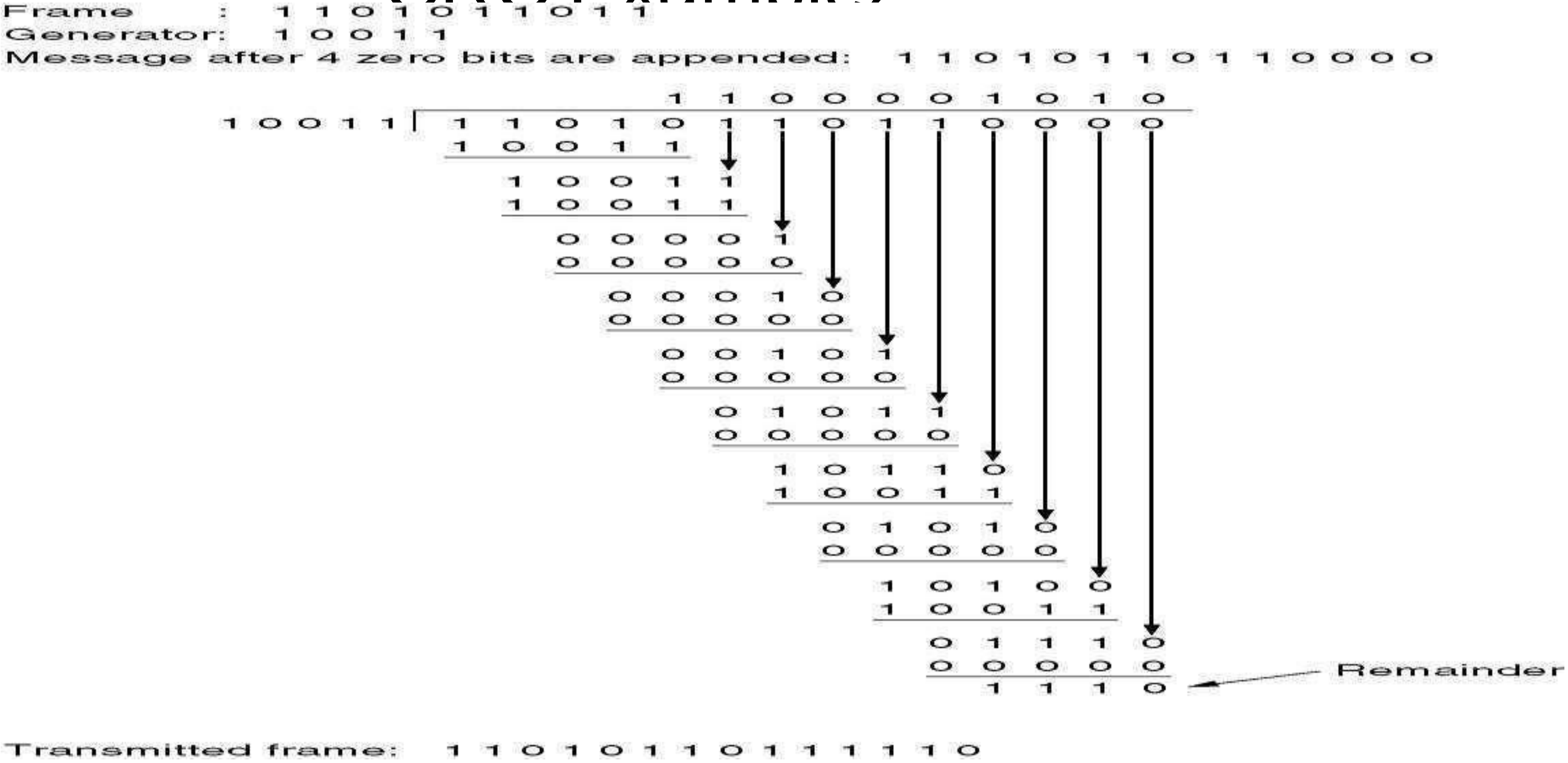
The divisor in a cyclic code is normally called the generator polynomial or simply the generator.

Table 10.7 ***Standard polynomials***



Name	*Polynomial*	*Application*
CRC-8	$x^8 + x^2 + x + 1$	ATM header
CRC-10	$x^{10} + x^9 + x^5 + x^4 + x^2 + 1$	ATM AAL
CRC-16	$x^{16} + x^{12} + x^5 + 1$	HDLC
CRC-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} +$ $x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	LANs

CRC Example2



Frame : 1 1 0 1 0 1 1 0 1 1

Generator: 1 0 0 1 1

Message after 4 zero bits are appended: 1 1 0 1 0 1 1 0 1 1 0 0 0 0

```
                    1 1 0 0 0 0 1 0 1 0
           ____________________________
1 0 0 1 1 | 1 1 0 1 0 1 1 0 1 1 0 0 0 0
            1 0 0 1 1
            ---------
              1 0 0 1 1
              1 0 0 1 1
              ---------
                0 0 0 0 1
                0 0 0 0 0
                ---------
                  0 0 0 1 0
                  0 0 0 0 0
                  ---------
                    0 0 1 0 1
                    0 0 0 0 0
                    ---------
                      0 1 0 1 1
                      0 0 0 0 0
                      ---------
                        1 0 1 1 0
                        1 0 0 1 1
                        ---------
                          0 1 0 1 0
                          0 0 0 0 0
                          ---------
                            1 0 1 0 0
                            1 0 0 1 1
                            ---------
                              0 1 1 1 0
                              0 0 0 0 0
                              ---------
                                1 1 1 0   <---- Remainder
```

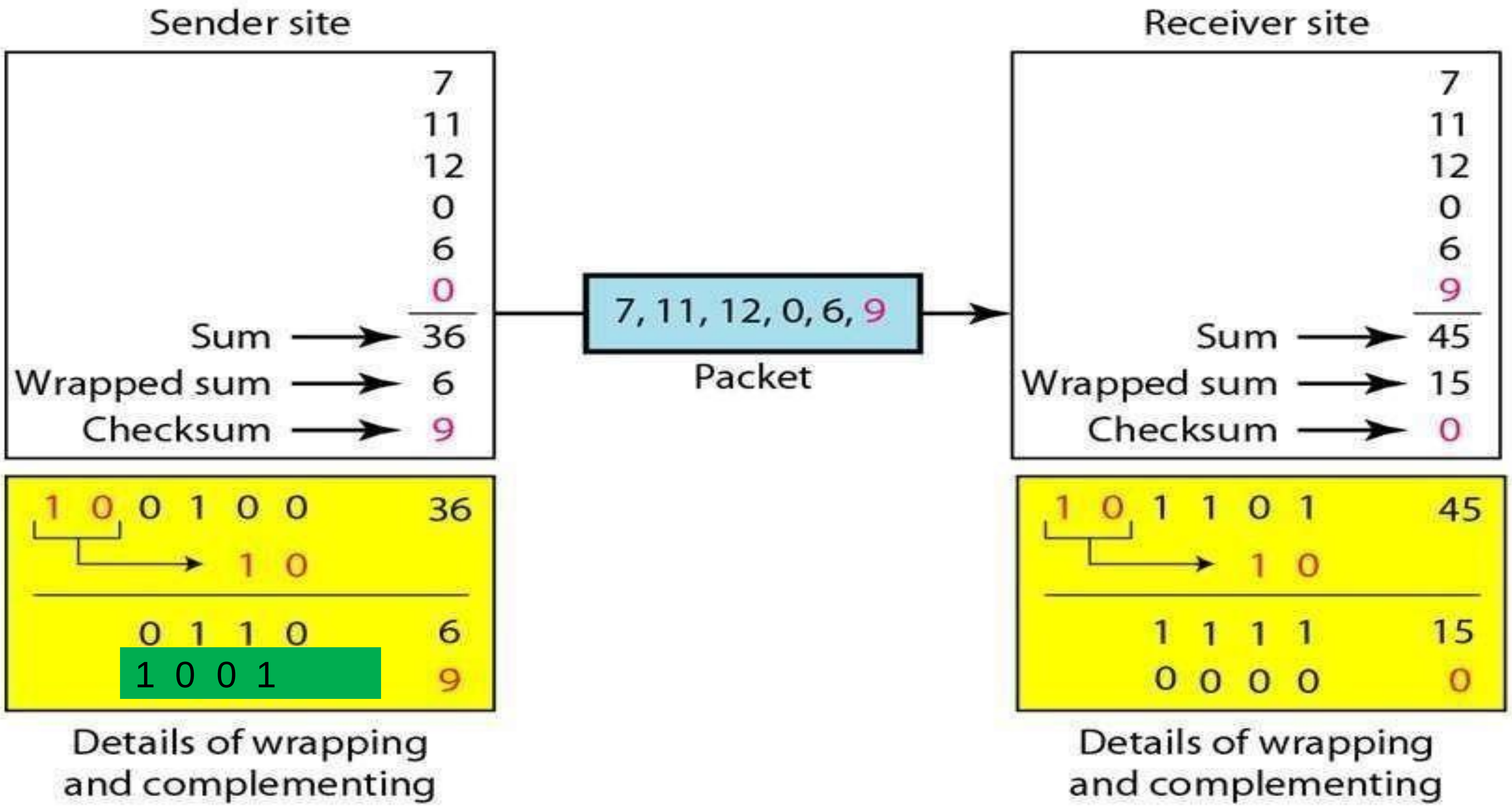
Transmitted frame: 1 1 0 1 0 1 1 0 1 1 1 1 1 0

CHECKSUM

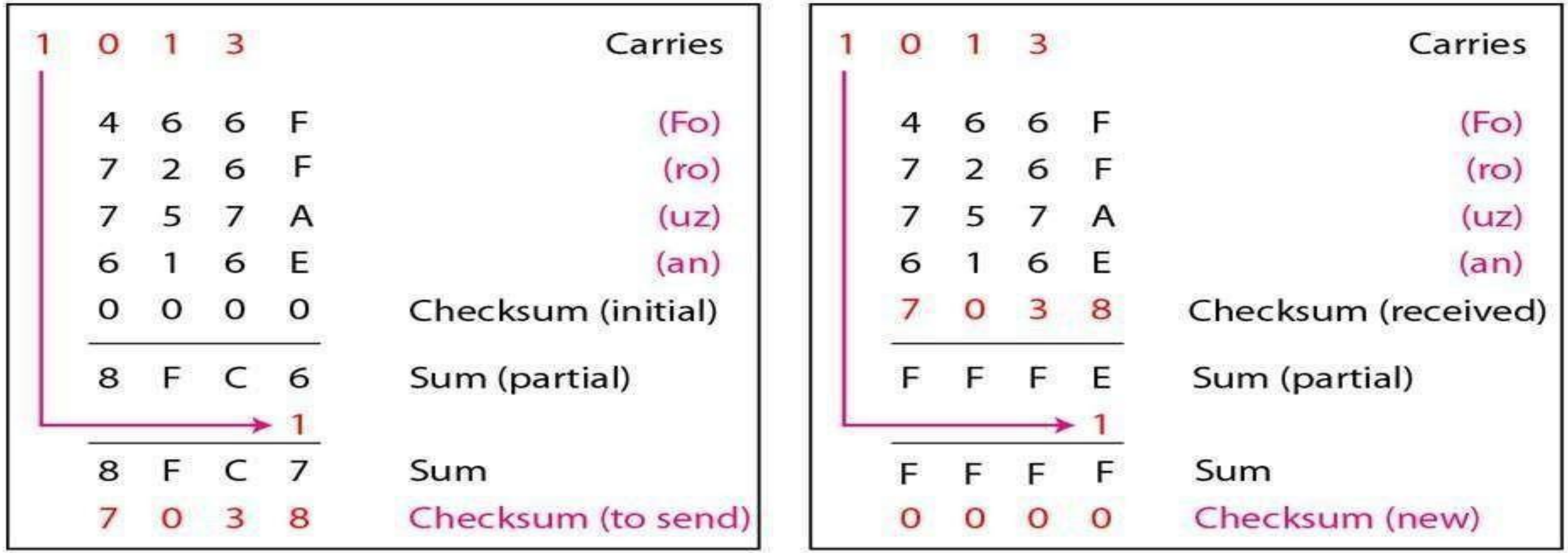
CHECKSUM

checksum is used detect errors and used in the Internet by several protocols although not at the data link layer.

checksum is used detect errors and used in the Internet by several protocols although not at the data link layer.


Sender site
7
11
12
0
6
0
Sum
36
Wrapped sum
6
Checksum
9
7, 11, 12, 0, 6, 9
Packet
Receiver site
7
11
12
0
6
9
Sum
45
Wrapped sum
15
Checksum
0
1 0 0 1 0 0
36
1 0
0 1 1 0
6
1 0 0 1
9
1 0 1 1 0 1
45
1 0
1 1 1 1
15
0 0 0 0
0
Details of wrapping and complementing
Details of wrapping and complementing

checksum



1	0	1	3		Carries
	4	6	6	F	(Fo)
	7	2	6	F	(ro)
	7	5	7	A	(uz)
	6	1	6	E	(an)
	0	0	0	0	Checksum (initial)
	8	F	C	6	Sum (partial)
				1	
	8	F	C	7	Sum
	7	0	3	8	Checksum (to send)

a. Checksum at the sender site

1	0	1	3		Carries
	4	6	6	F	(Fo)
	7	2	6	F	(ro)
	7	5	7	A	(uz)
	6	1	6	E	(an)
	7	0	3	8	Checksum (received)
	F	F	F	E	Sum (partial)
				1	
	F	F	F	F	Sum
	0	0	0	0	Checksum (new)

a. Checksum at the receiver site

FLOWAND ERROR CONTROL

The most important responsibilities of the data link layer are flow control and error control. Collectively, these functions are known as data link control.

Topics discussed in this section:

Flow Control Error
Control

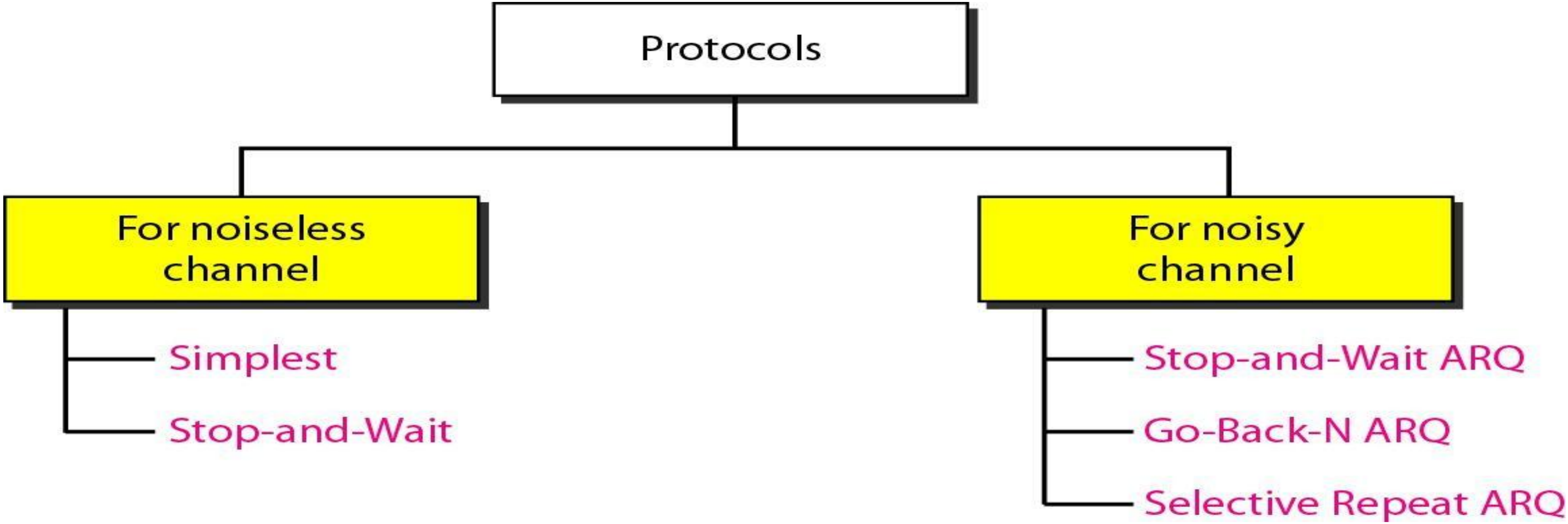
Note

Flow control refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgment.

Note

Error control in the data link layer is based on automatic repeat request, which is the retransmission of data.

Figure *Taxonomy of protocols discussed in this chapter*



NOISELESS CHANNELS

Let us first assume we have an ideal channel in which no frames are lost, duplicated, or corrupted. We introduce two protocols for this type of channel.

Topics discussed in this section:

Simplest Protocol
Stop-and-Wait Protocol

simplest protocol *with no flow or error control*

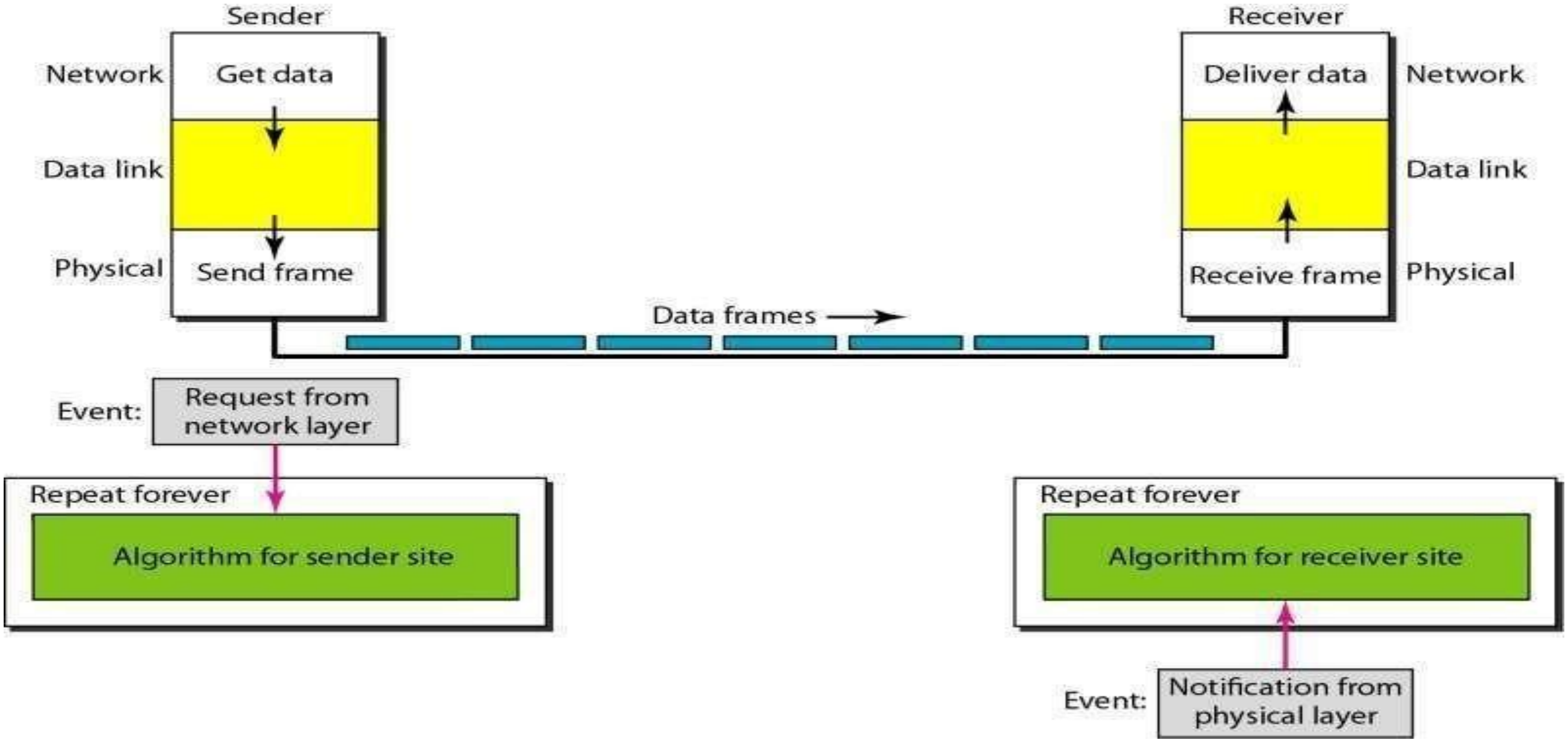


Figure *Flow diagram*

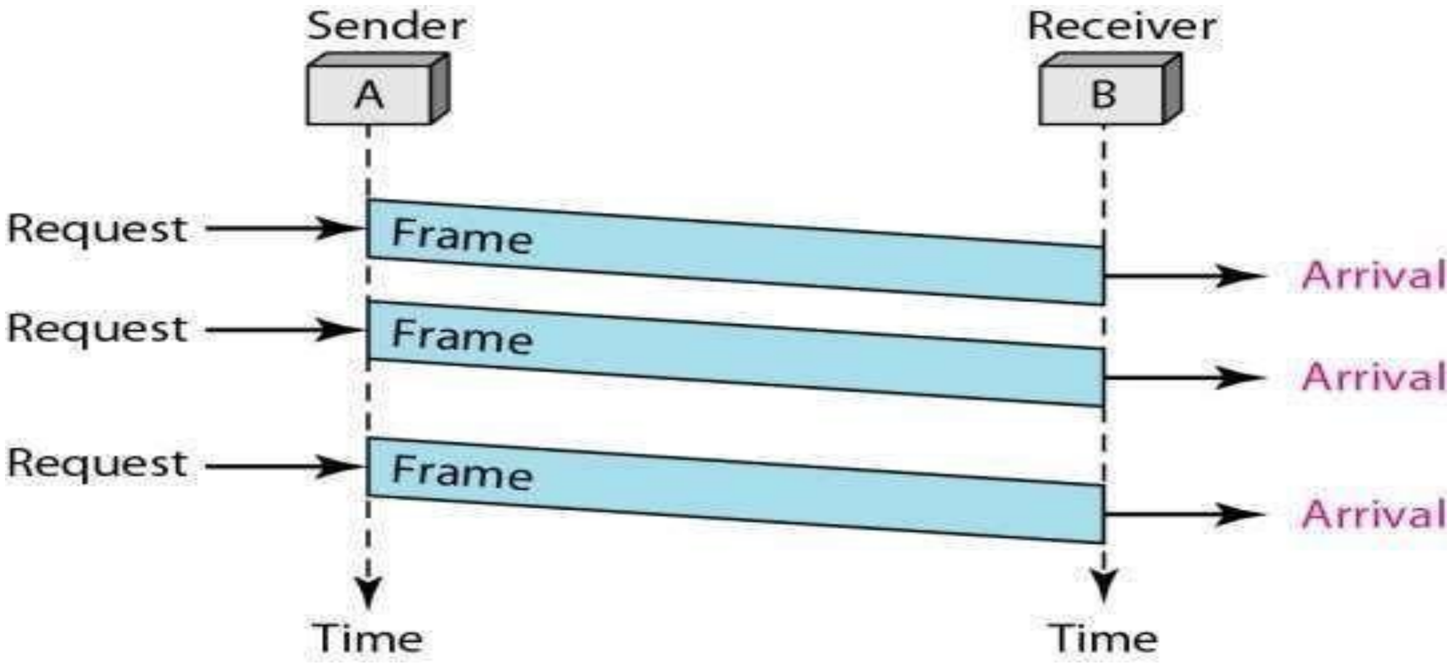


Figure *Design of Stop-and-Wait Protocol*

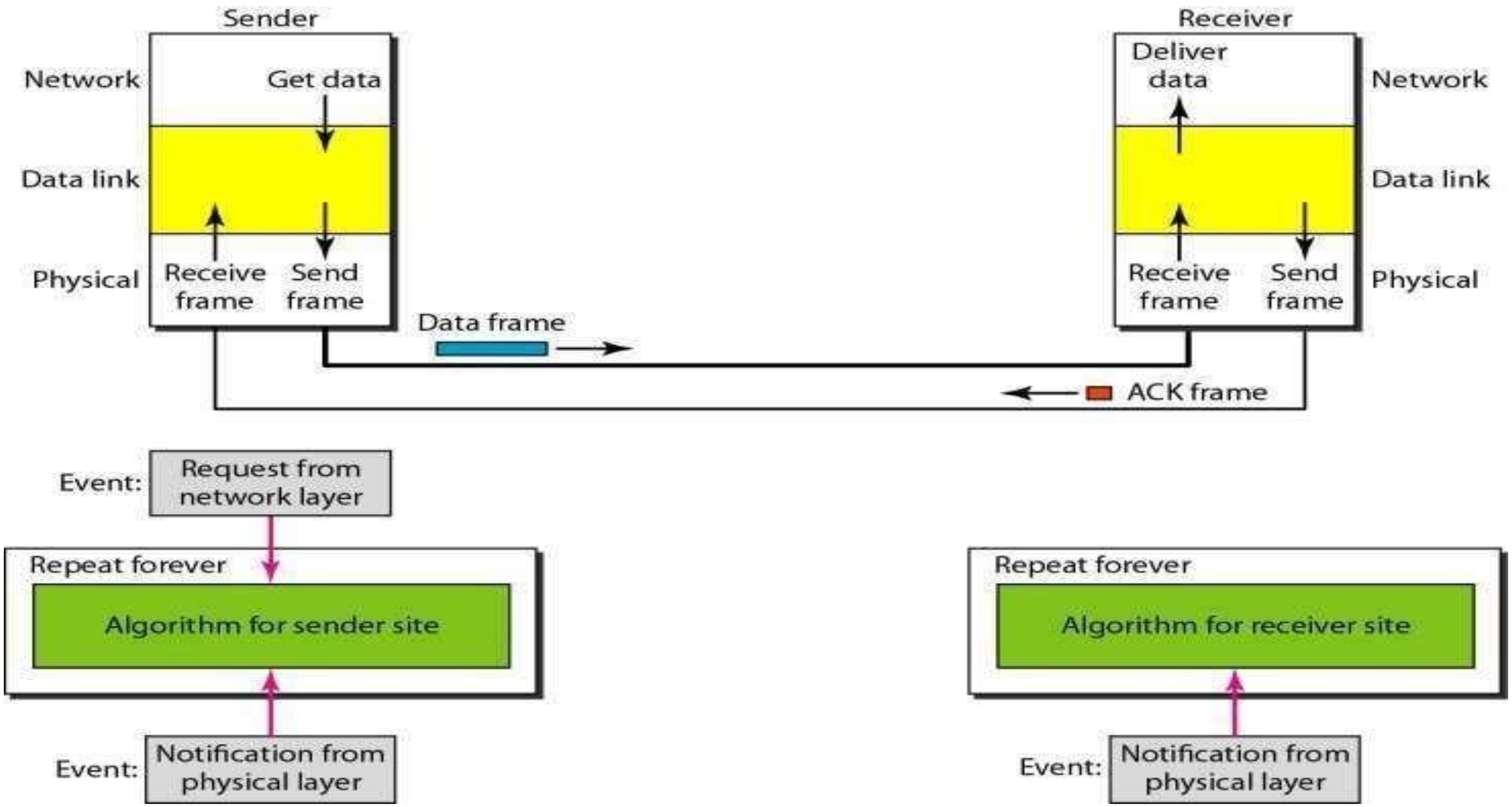
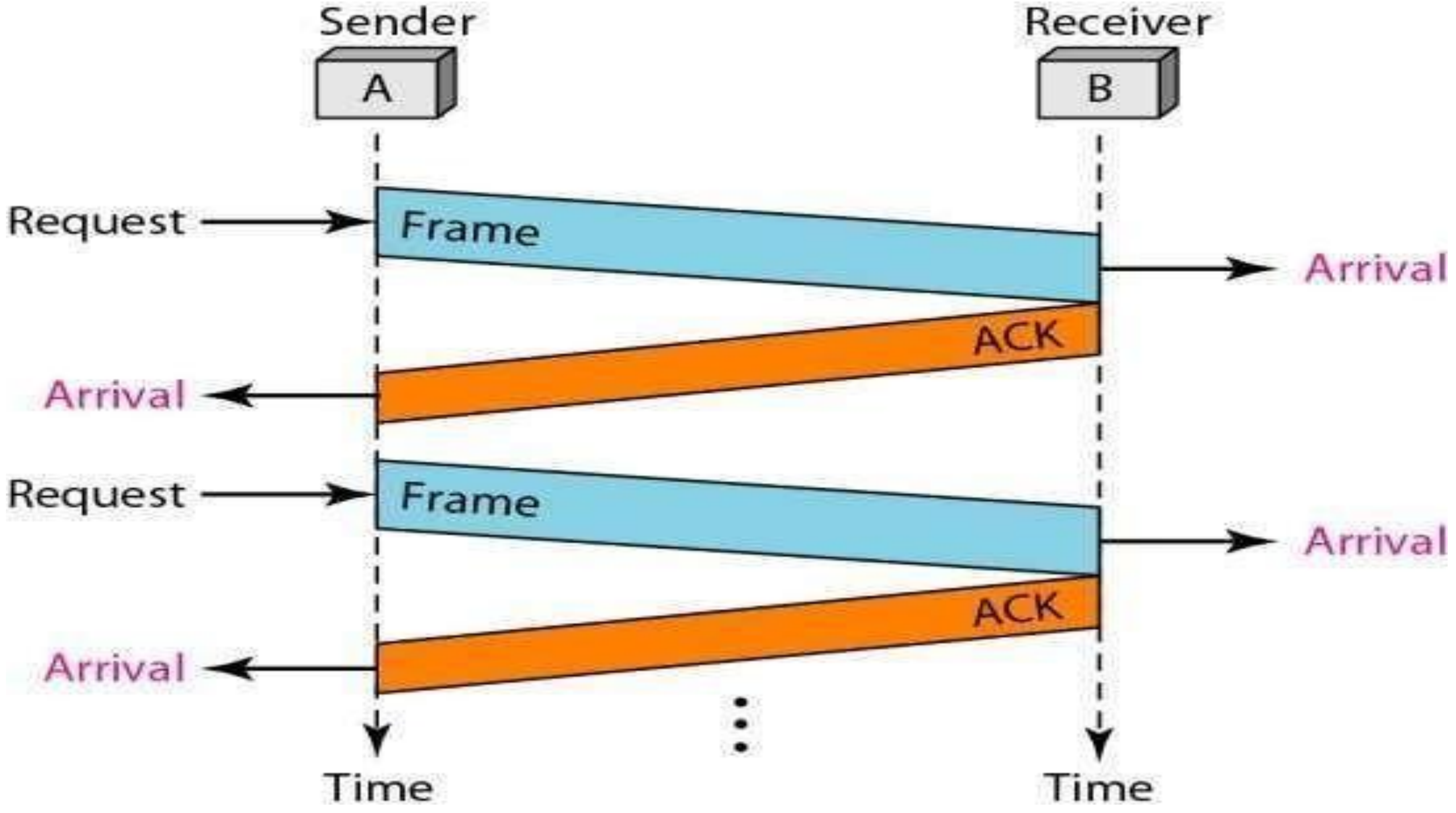


Figure *Flow diagram*



NOISY CHANNELS

Although the Stop-and-Wait Protocol gives us an idea of how to add flow control to its predecessor, noiseless channels are nonexistent. We discuss three protocols in this section that use error control.

Topics discussed in this section:

Topics discussed in this section:

Stop-and-Wait Automatic Repeat Request Go-Back-N Automatic
Repeat Request Selective Repeat Automatic Repeat Request

Note

Error correction in Stop-and-Wait ARQ is done by keeping a copy of the sent frame and retransmitting of the frame when the timer expires.

Note

In Stop-and-Wait ARQ, we use sequence numbers to number the frames.

Figure *Design of the Stop-and-WaitARQ Protocol*

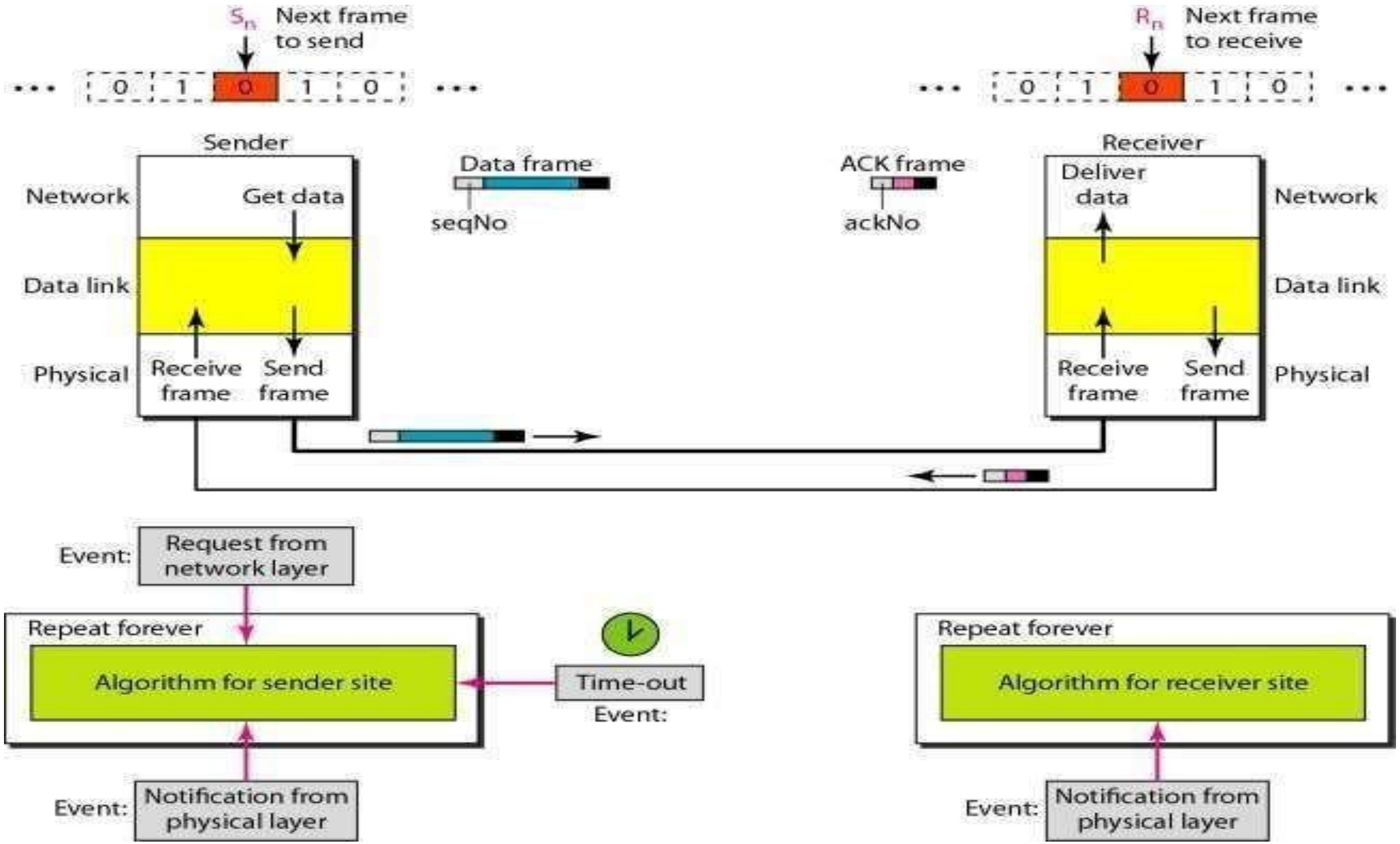


Figure ***Flow diagram Stop-and-Wait ARQ Protocol***

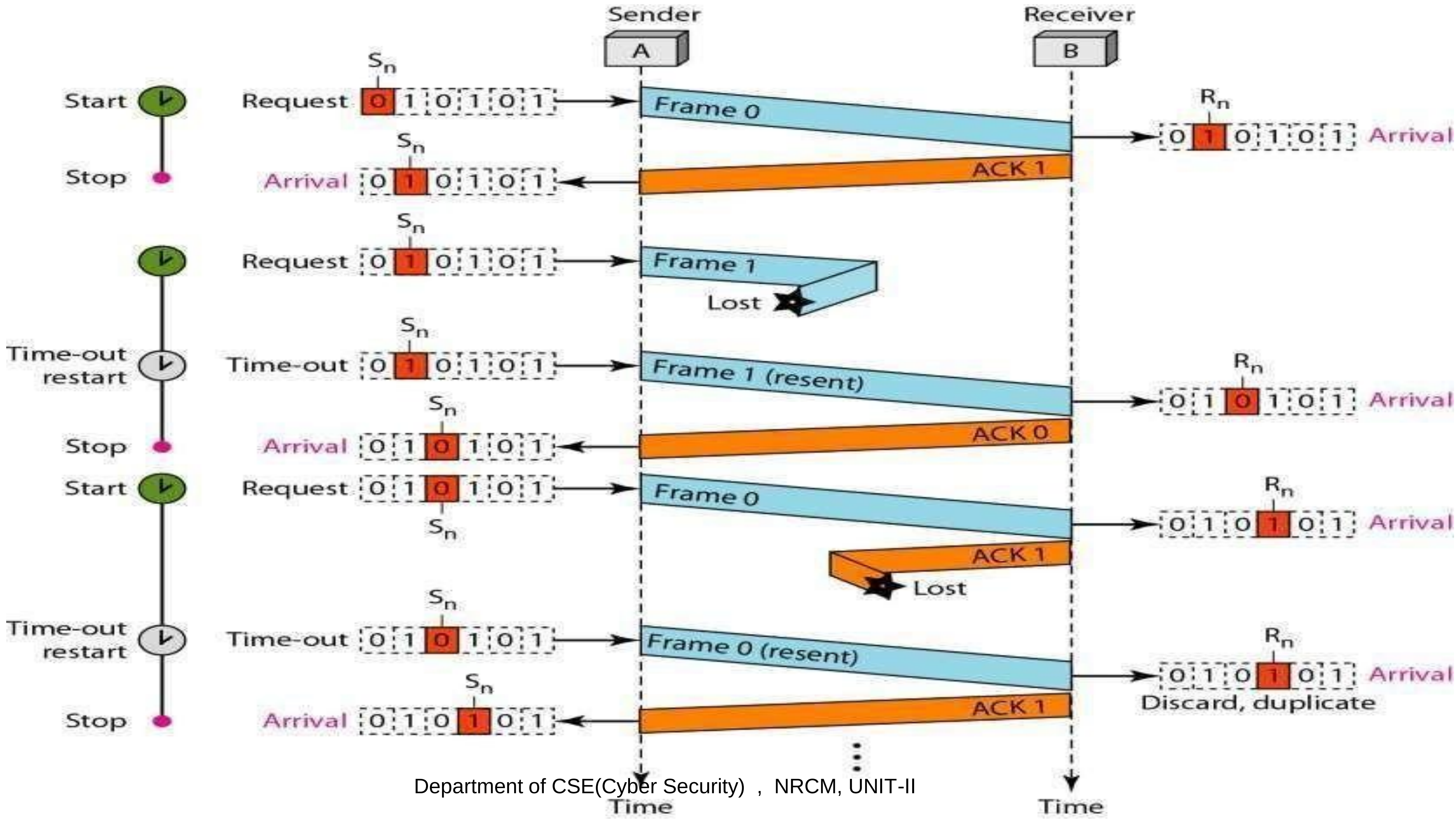
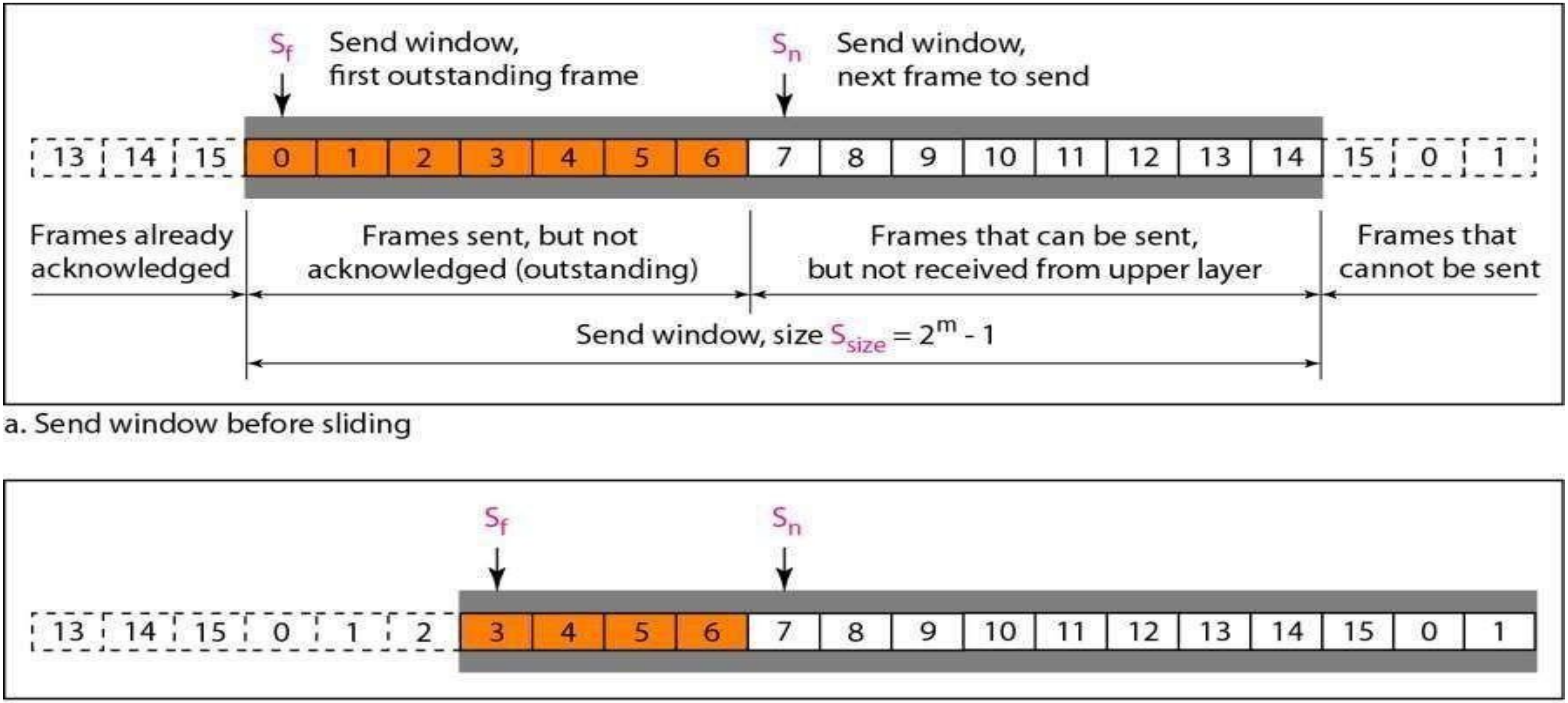


Figure Send window for Go-Back-N ARQ


S_f Send window, first outstanding frame
S_n Send window, next frame to send
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1
Frames already acknowledged
Frames sent, but not acknowledged (outstanding)
Frames that can be sent, but not received from upper layer
Frames that cannot be sent
Send window, size $S_{size} = 2^m - 1$
S_f
S_n
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1

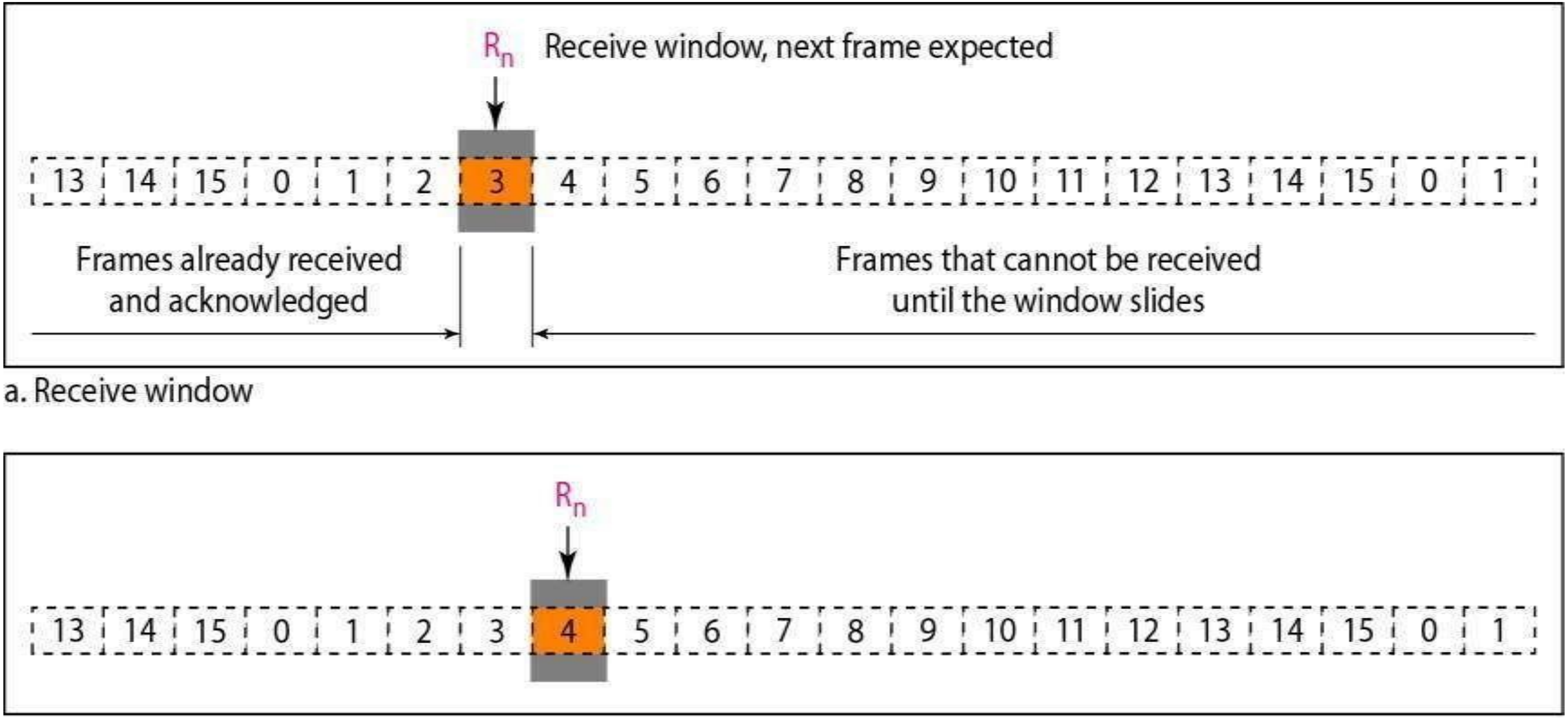
a. Send window before sliding

b. Send window after sliding

Note

The send window is an abstract concept defining an imaginary box of size $2^m - 1$ with three variables: S_f, S_n, and S_{size}.

Figure Receive window for Go-Back-N ARQ


R_n
Receive window, next frame expected
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1
Frames already received and acknowledged
Frames that cannot be received until the window slides
R_n
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1

a. Receive window

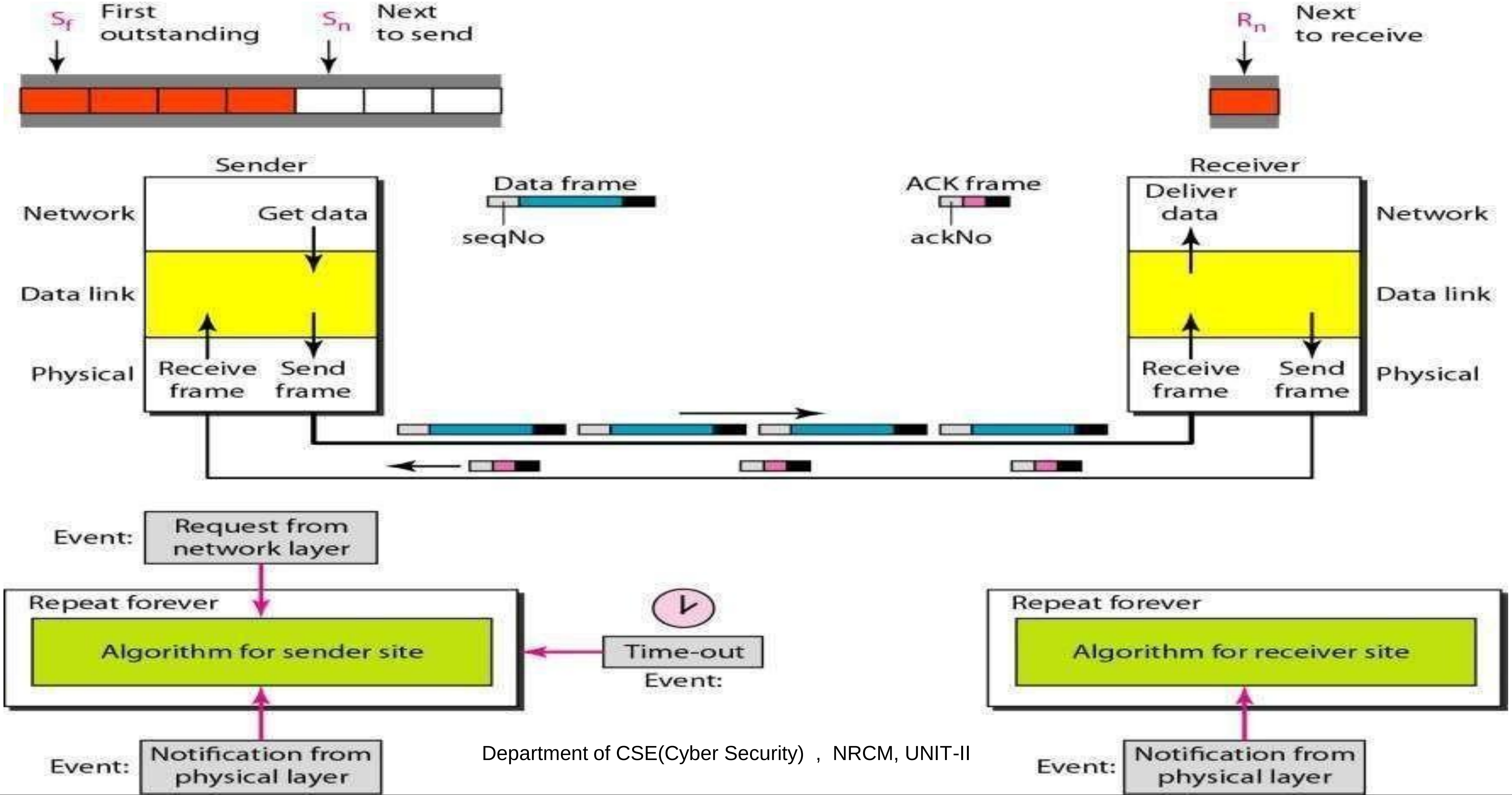
b. Window after sliding

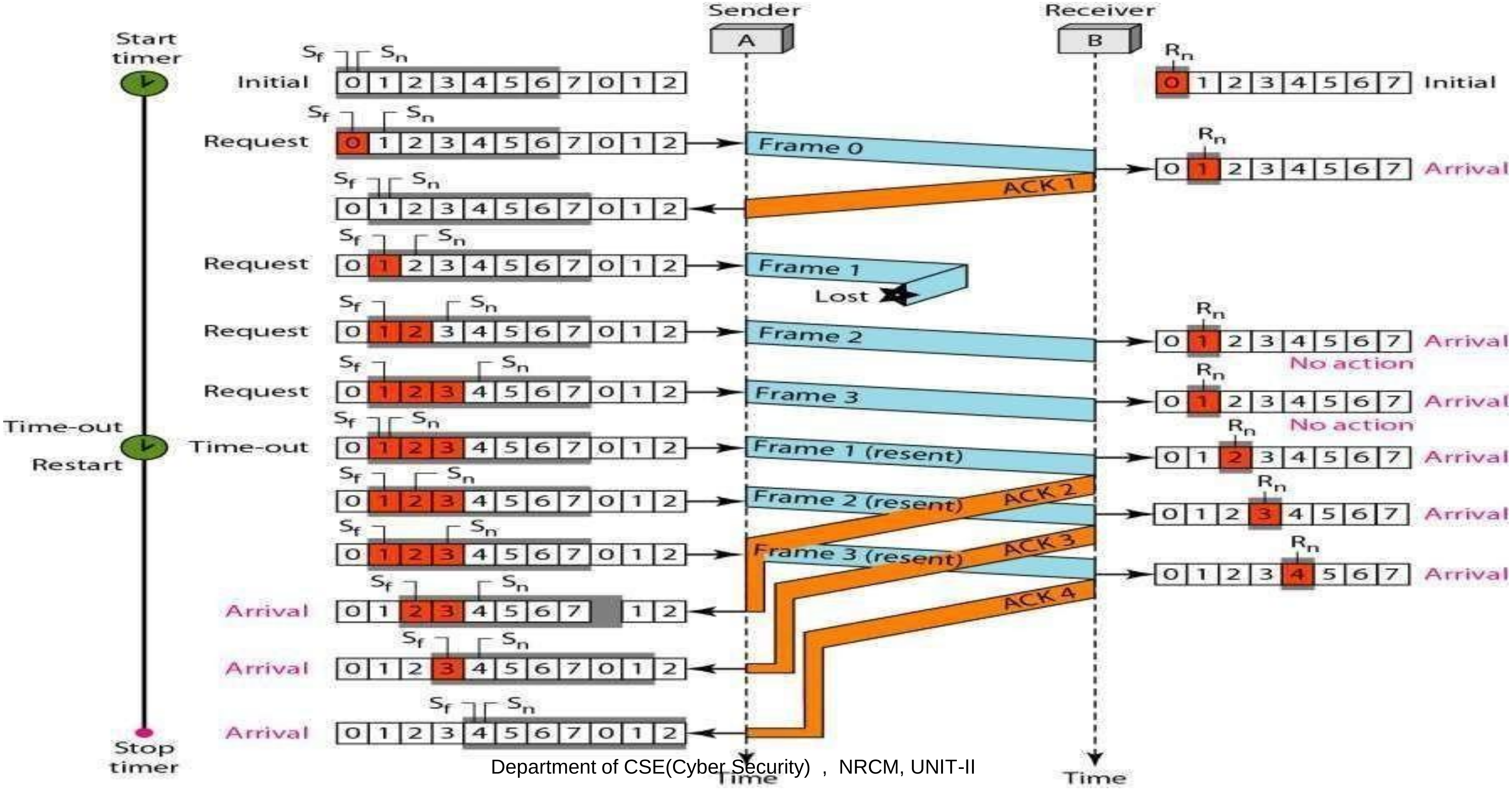
Note

- The receive window is an abstract concept defining an imaginary box of size 1 with one single variable R_n.

- The window slides when a correct frame has arrived; sliding occurs one slot at a time.

Figure *Design of Go-Back-N ARQ*

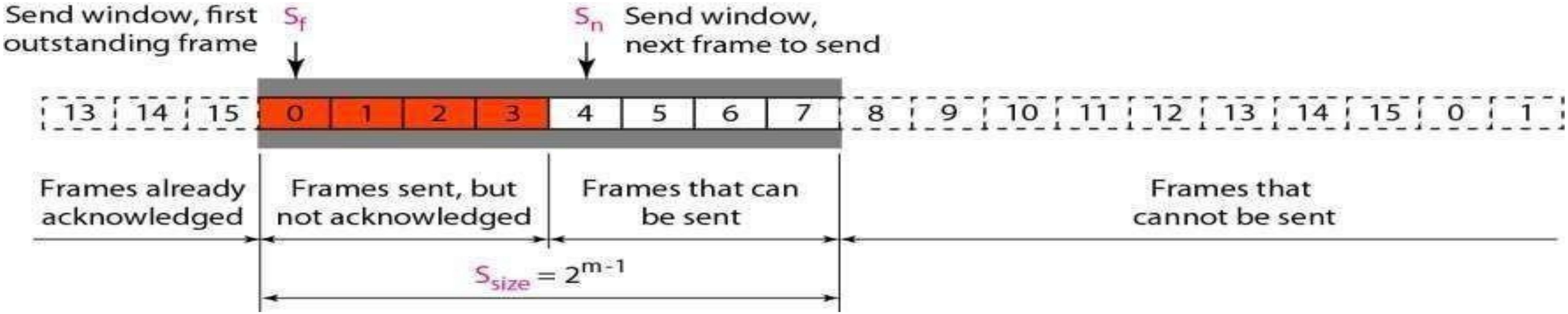



Sender
A
Receiver
B
Start timer
Initial
Request
Frame 0
ACK 1
Arrival
Frame 1
Lost
Frame 2
No action
Frame 3
Time-out
Restart
Frame 1 (resent)
Frame 2 (resent)
ACK 2
Frame 3 (resent)
ACK 3
ACK 4
Stop timer
Time
Time
S_f
S_n
R_n

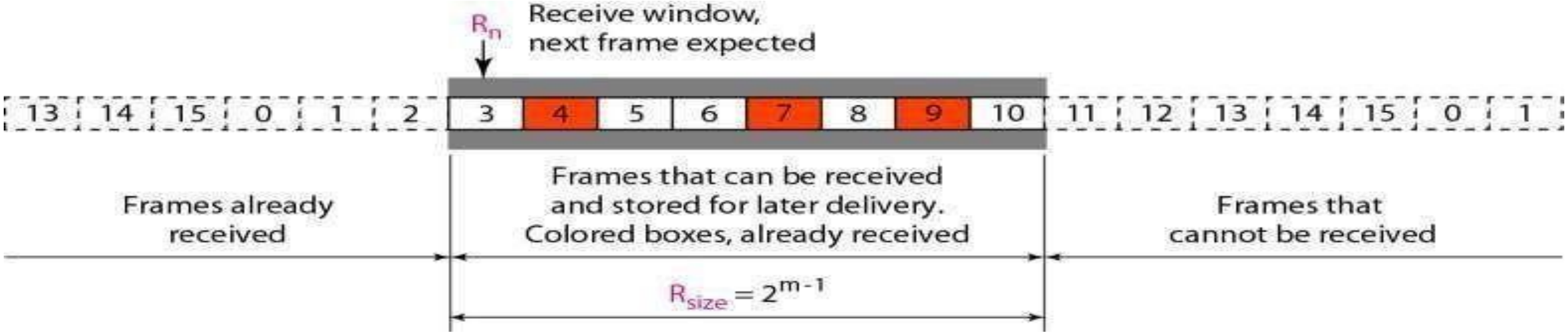
Note

Stop-and-Wait ARQ is a special case of Go-Back-N ARQ in which the size of the send window is 1.

Selective Repeat ARQ

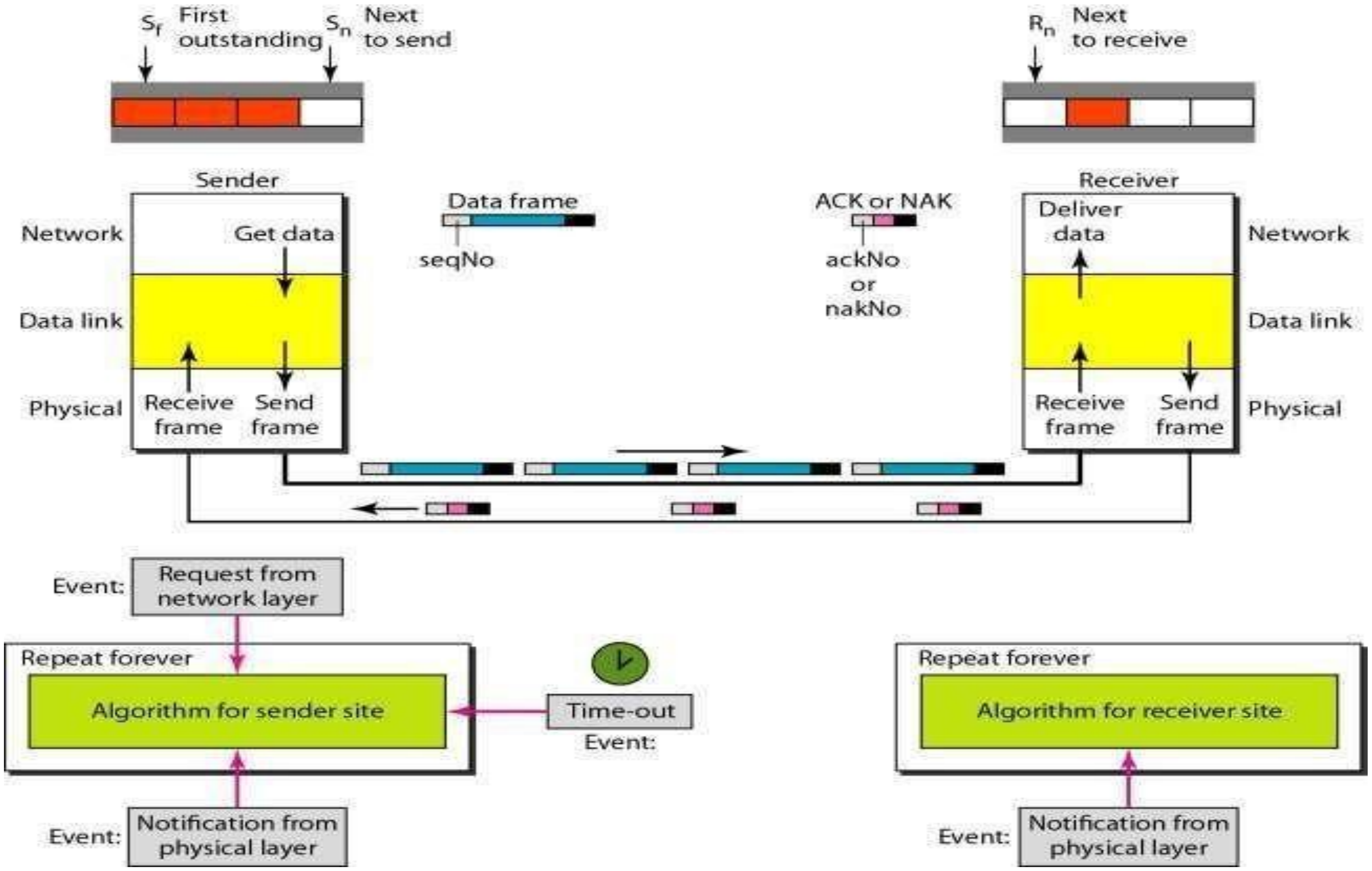

Send window, first outstanding frame
S_f
S_n
Send window, next frame to send
13
14
15
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
0
1
Frames already acknowledged
Frames sent, but not acknowledged
Frames that can be sent
Frames that cannot be sent
$S_{size} = 2^{m-1}$

Receive window for Selective Repeat ARQ



R_n
Receive window, next frame expected
13
14
15
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
0
1
Frames already received
Frames that can be received and stored for later delivery. Colored boxes, already received
Frames that cannot be received
$R_{size} = 2^{m-1}$

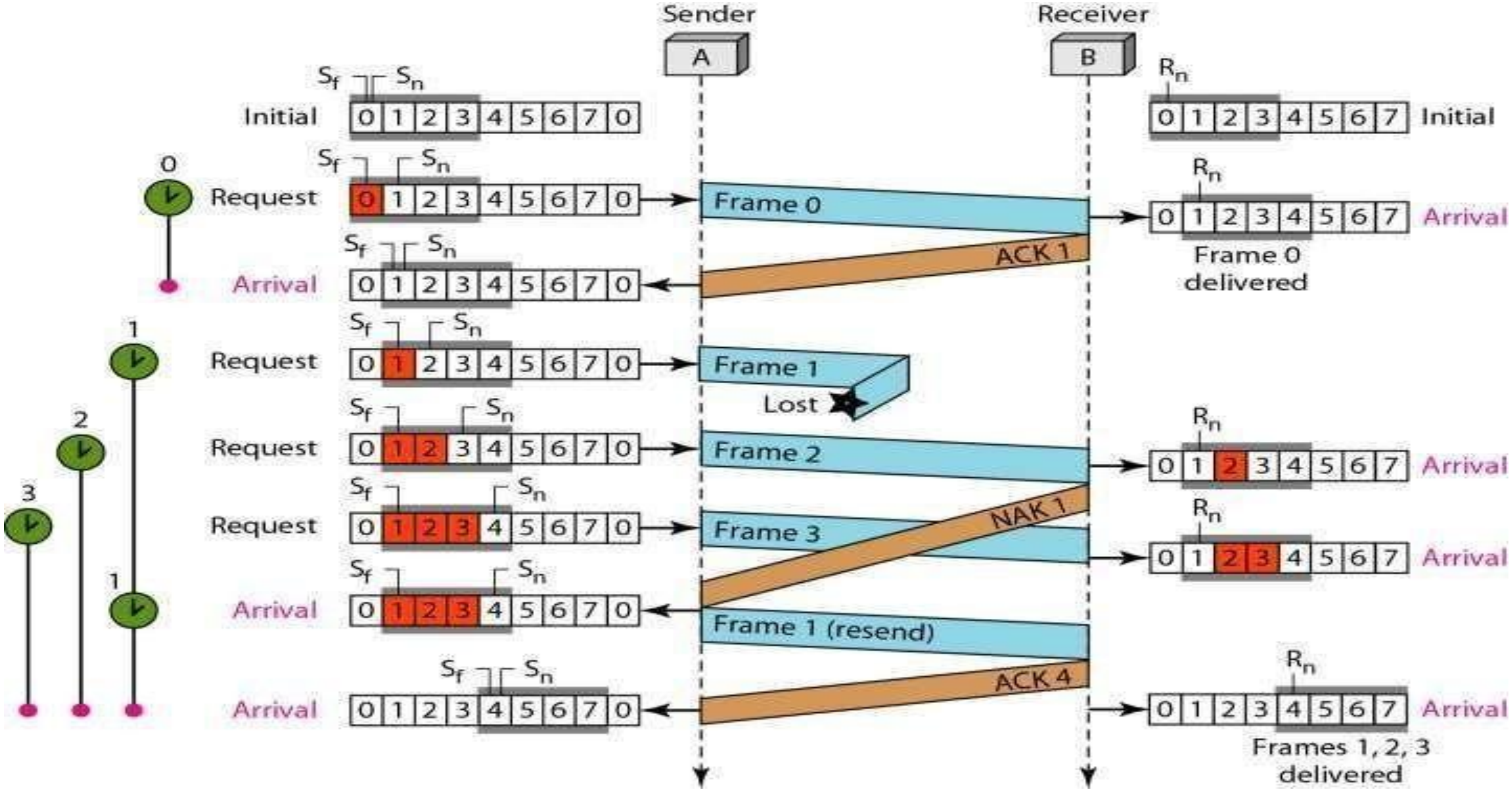
Figure 11.20 *Design of Selective Repeat ARQ*



Note

In Selective Repeat ARQ, the size of the sender and receiver window must be at most one-half of 2^m.

Figure *Flow diagram for* *Selective Repeat ARQ*



11-6 HDLC

11-6 HDLC

- High-level Data Link Control (HDLC) is a bit-oriented protocol for communication over point-to-point and multipoint links. It implements the ARQ mechanisms we discussed in this chapter.

High-level Data Link Control (HDLC) is a bit-oriented protocol for communication over point-to-point and multipoint links. It implements the ARQ mechanisms we discussed in this chapter.

Figure 11.27 *HDLC frames*

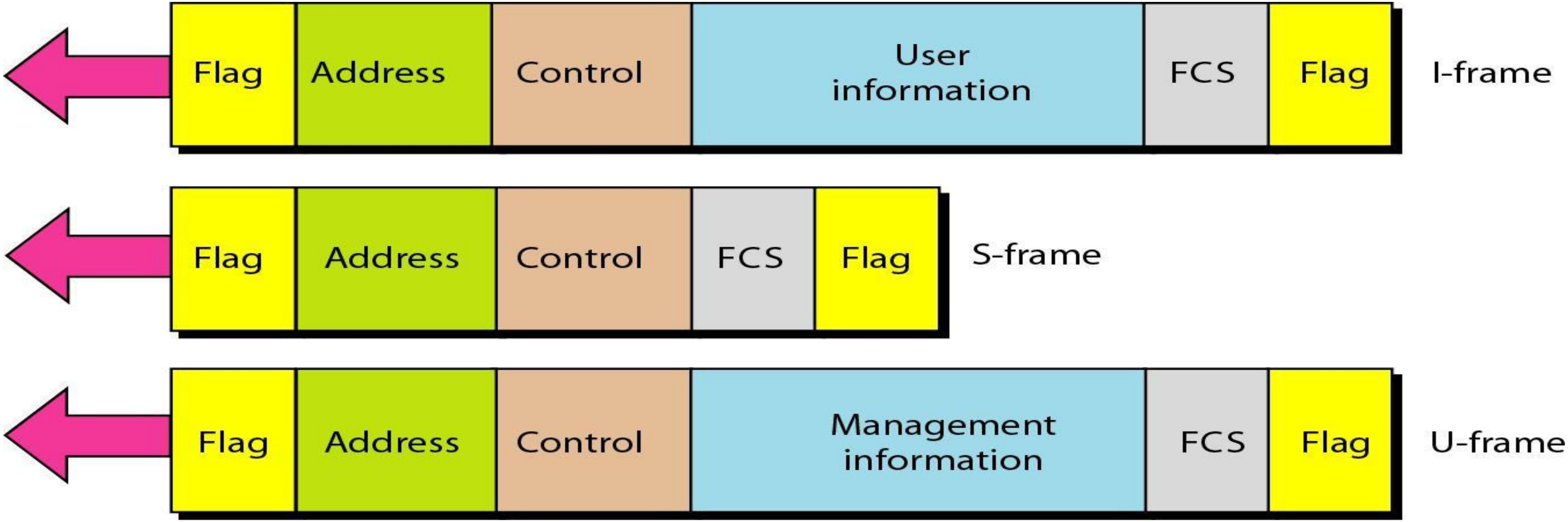
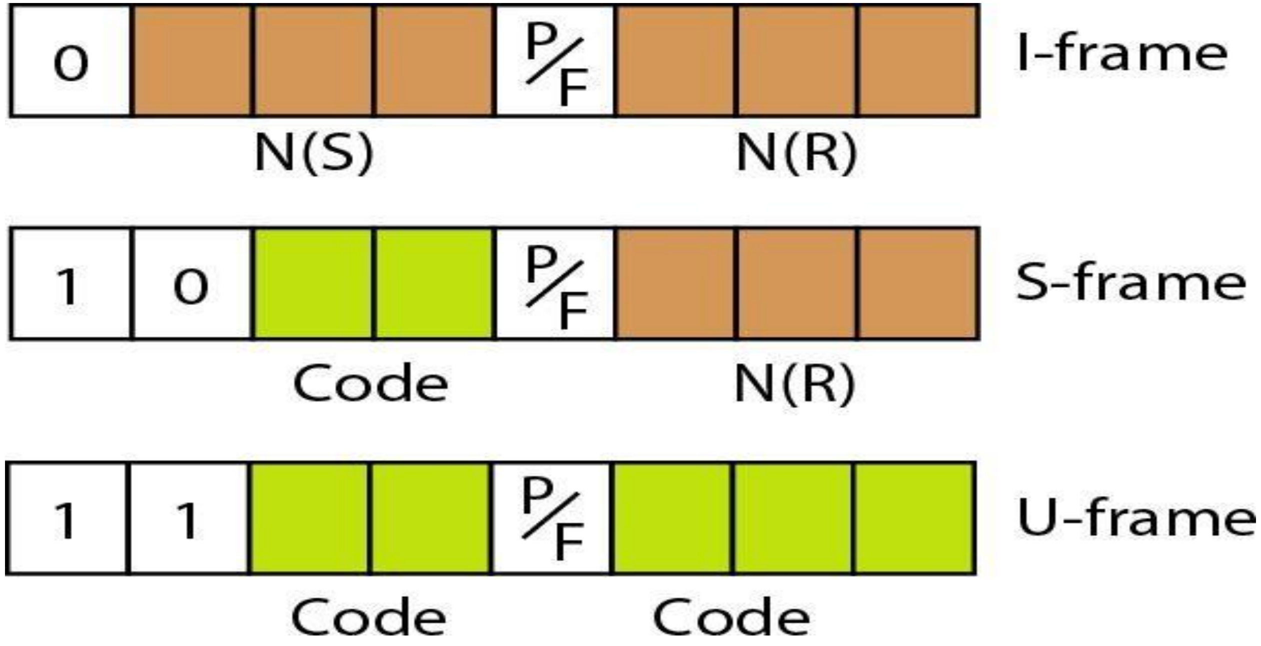


Figure 11.28 *Control field format for the different frame types*



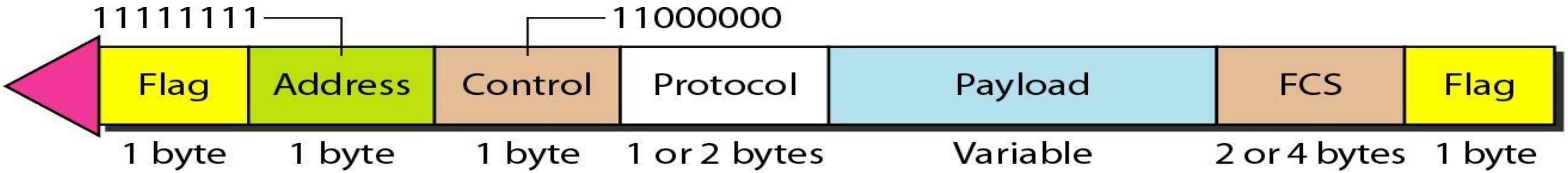
POINT-TO-POINT PROTOCOL

POINT-TO-POINT PROTOCOL

- Although HDLC is a general protocol that can be used for both point-to-point and multipoint configurations, one of the most common protocols for point-to-point access is the Point-to-Point Protocol (PPP). PPP is a byte-oriented protocol.

Although HDLC is a general protocol that can be used for both point-to-point and multipoint configurations, one of the most common protocols for point-to-point access is the Point-to-Point Protocol (PPP). PPP is a byte-oriented protocol.

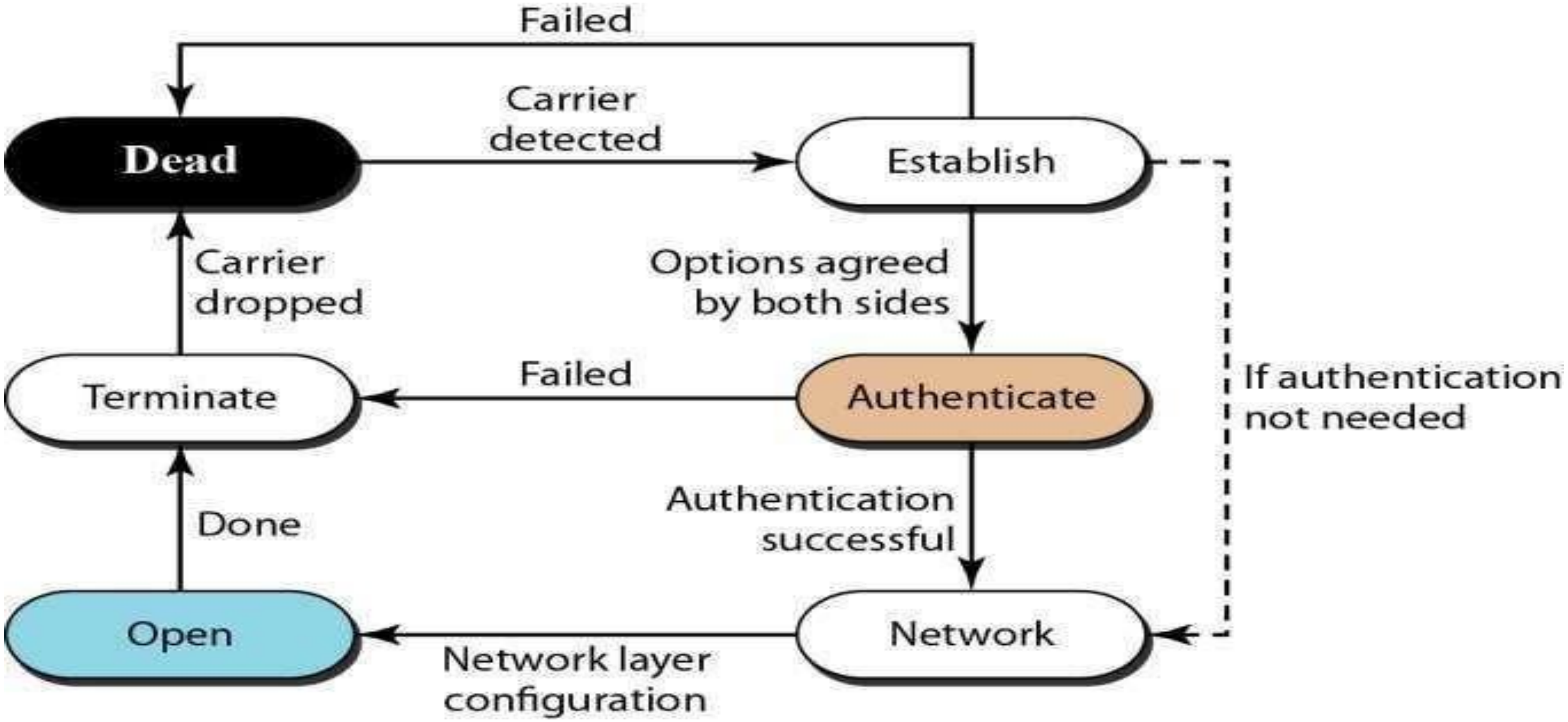
PPP frame format



Note

PPP is a byte-oriented protocol using byte stuffing with the escape byte 01111101.

Transition phases

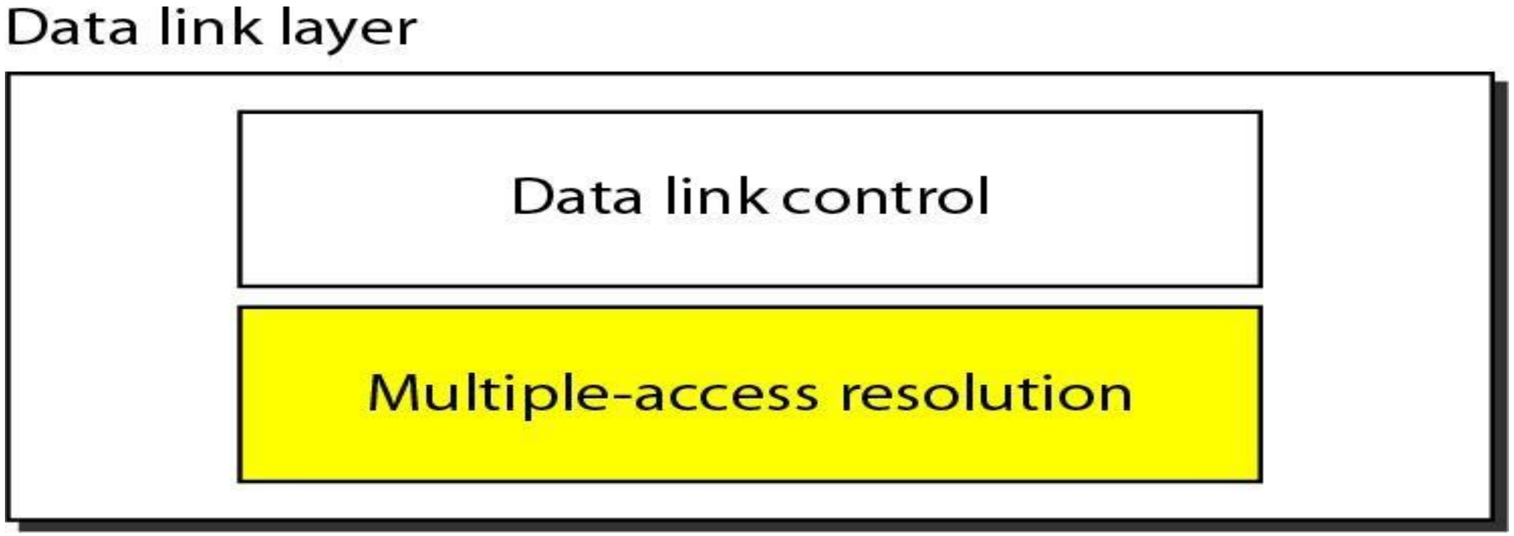

Failed
Carrier detected
Dead
Establish
Carrier dropped
Options agreed by both sides
Failed
Terminate
Authenticate
If authentication not needed
Done
Authentication successful
Open
Network
Network layer configuration

Medium Access Control

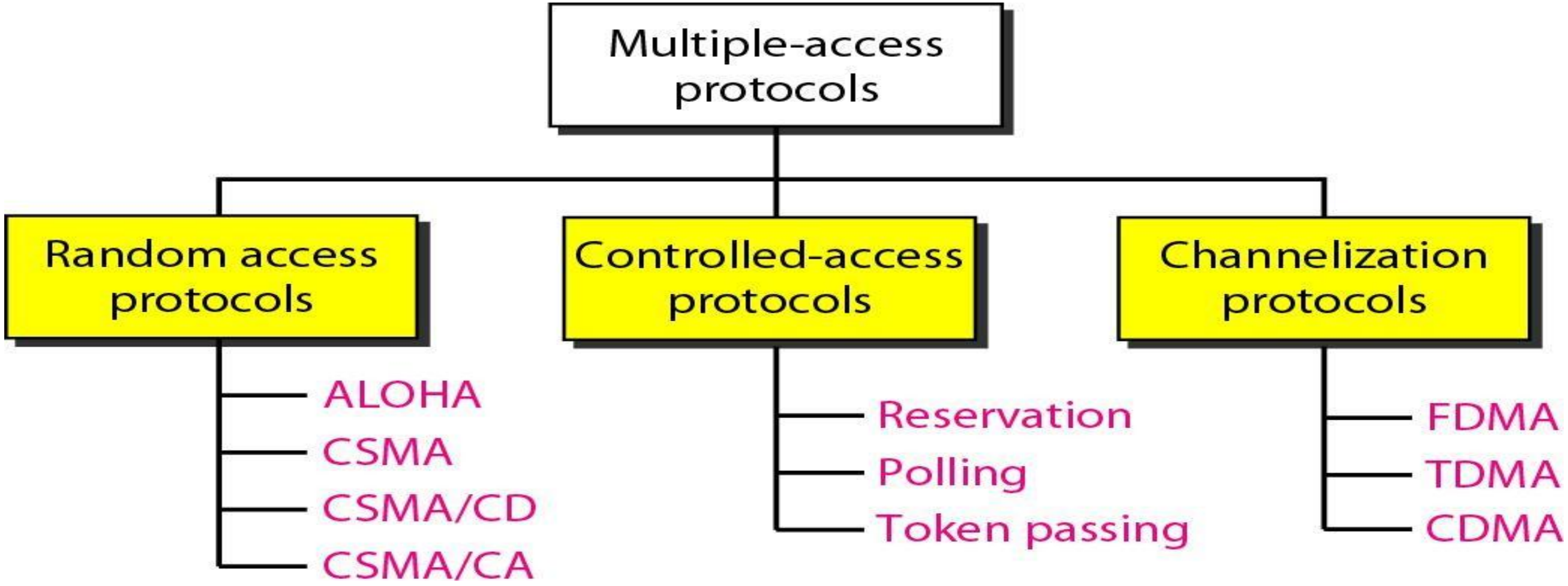
Chapter 12

Multiple Access

Data link layer divided into two functionality-oriented sublayers



Taxonomy of multiple-access protocols discussed in this chapter



RANDOM ACCESS

RANDOM ACCESS

In random access or contention methods, no station is superior to another station and none is assigned the control over another. No station permits, or does not permit, another station to send.

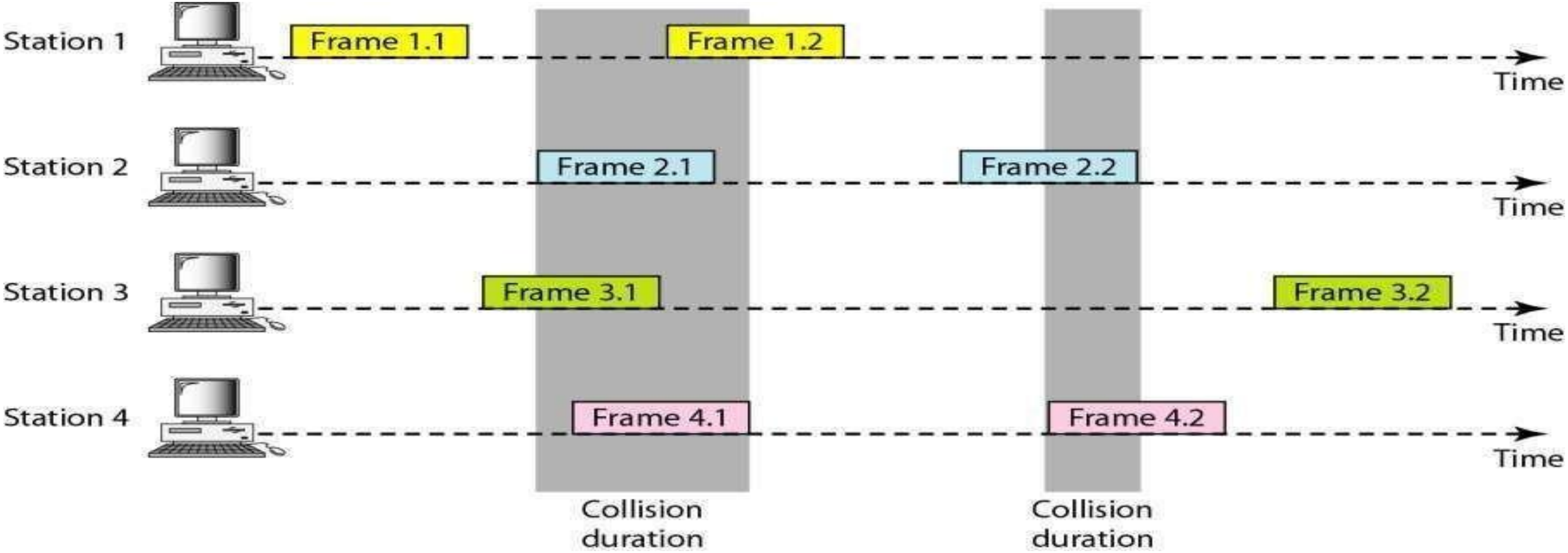
In random access or contention methods, no station is superior to another station and none is assigned the control over another. No station permits, or does not permit, another station to send.

Topics discussed in this section:

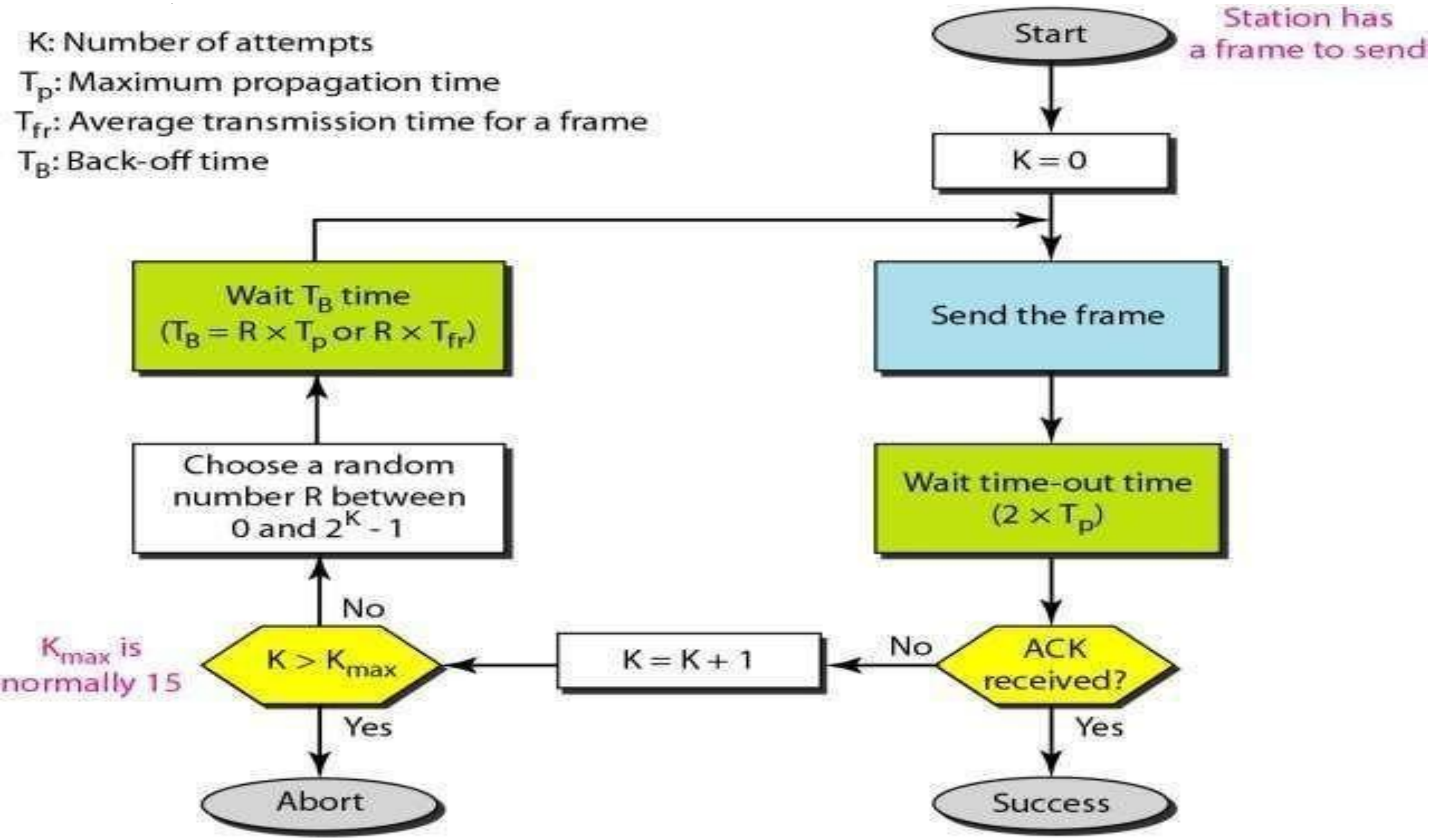
Topics discussed in this section:

ALOHA
Carrier Sense Multiple Access
Carrier Sense Multiple Access with Collision Detection
Carrier Sense Multiple Access with Collision Avoidance

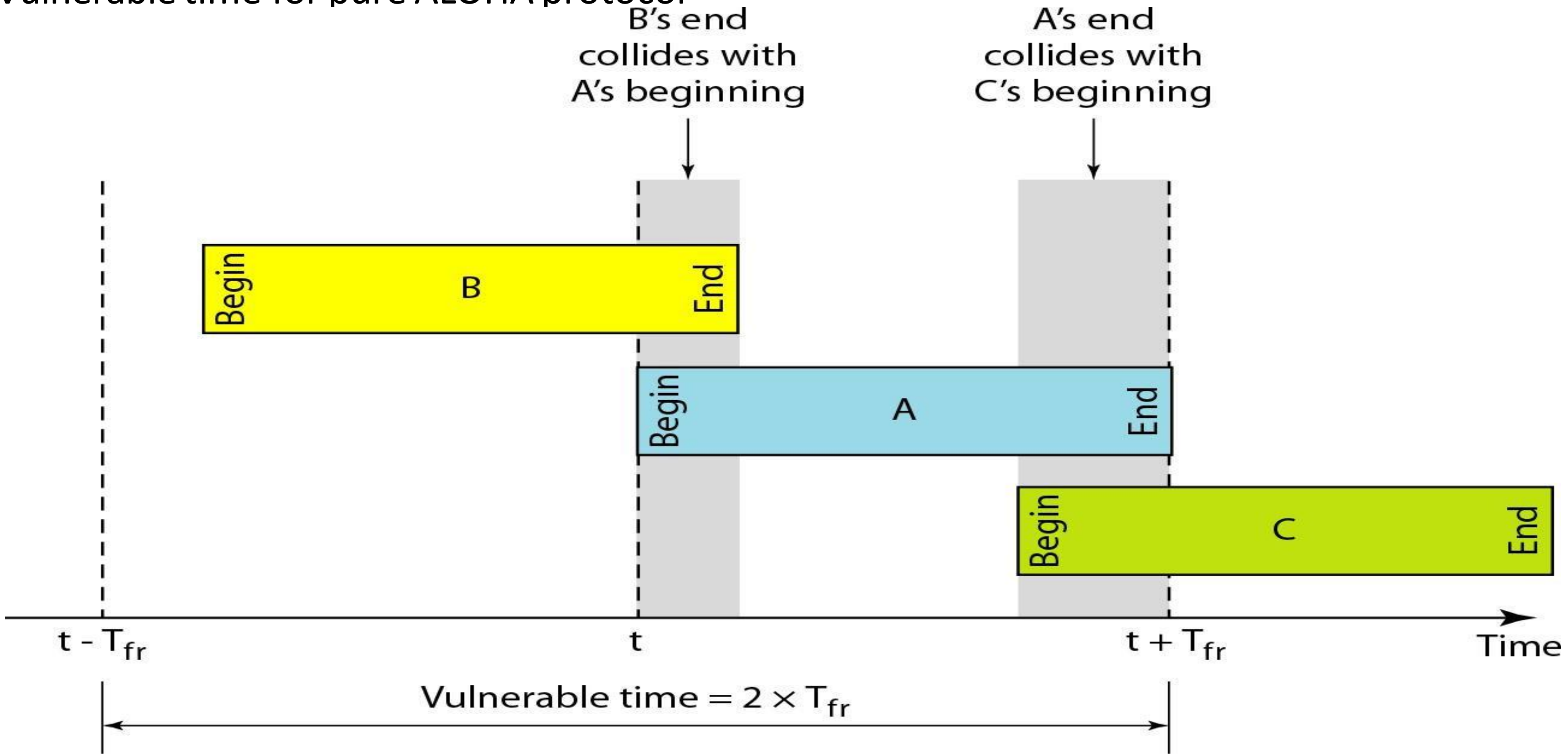
Frames in a pure ALOHA network



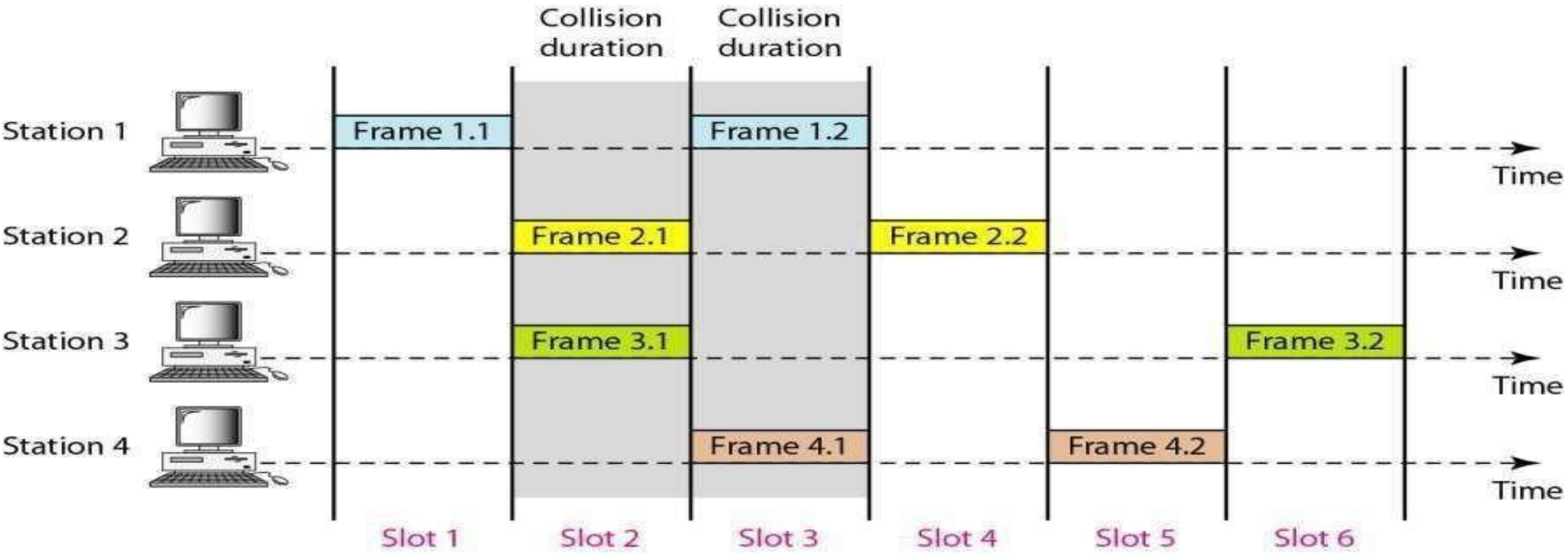
Pr


K: Number of attempts
T_p: Maximum propagation time
T_{fr}: Average transmission time for a frame
T_B: Back-off time
Start
Station has a frame to send
K = 0
Send the frame
Wait time-out time ($2 \times T_p$)
ACK received?
No
K = K + 1
Yes
Success
$K > K_{max}$
K_{max} is normally 15
No
Choose a random number R between 0 and $2^K - 1$
Wait T_B time ($T_B = R \times T_p$ or $R \times T_{fr}$)
Yes
Abort

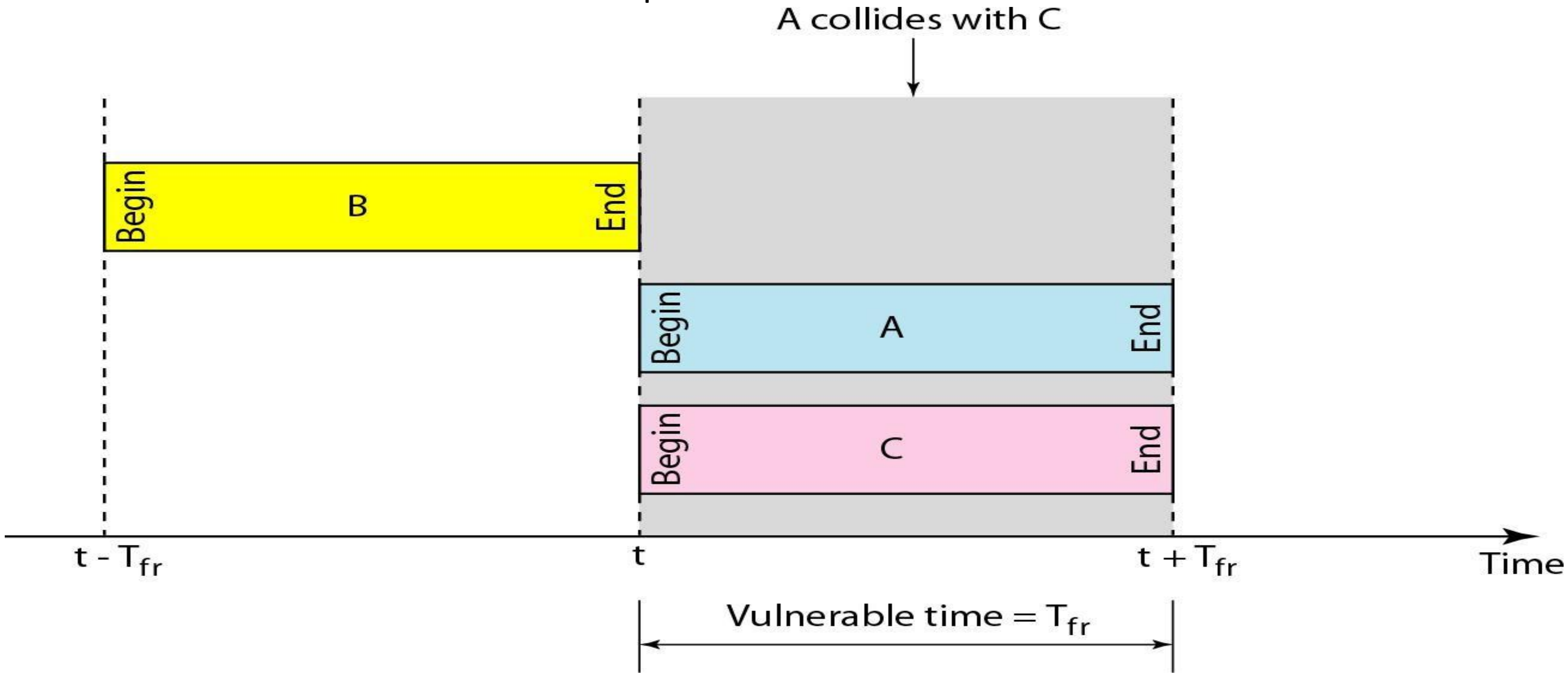
Vulnerable time for pure ALOHA protocol


B's end collides with A's beginning
A's end collides with C's beginning
Begin
B
End
Begin
A
End
Begin
C
End
$t - T_{fr}$
t
$t + T_{fr}$
Time
Vulnerable time = $2 \times T_{fr}$

Frames in a slotted ALOHA network



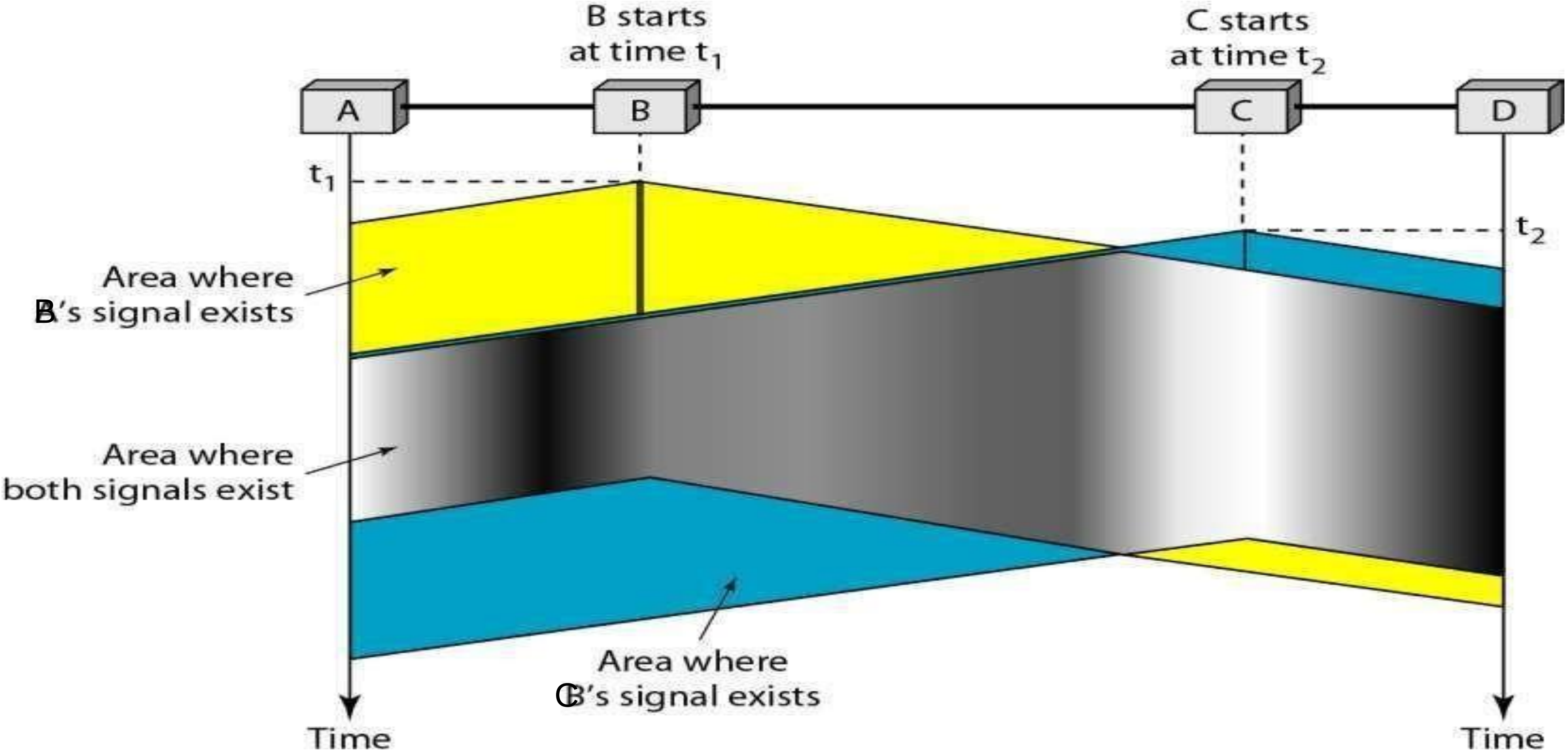
Vulnerable time for slotted ALOHA protocol



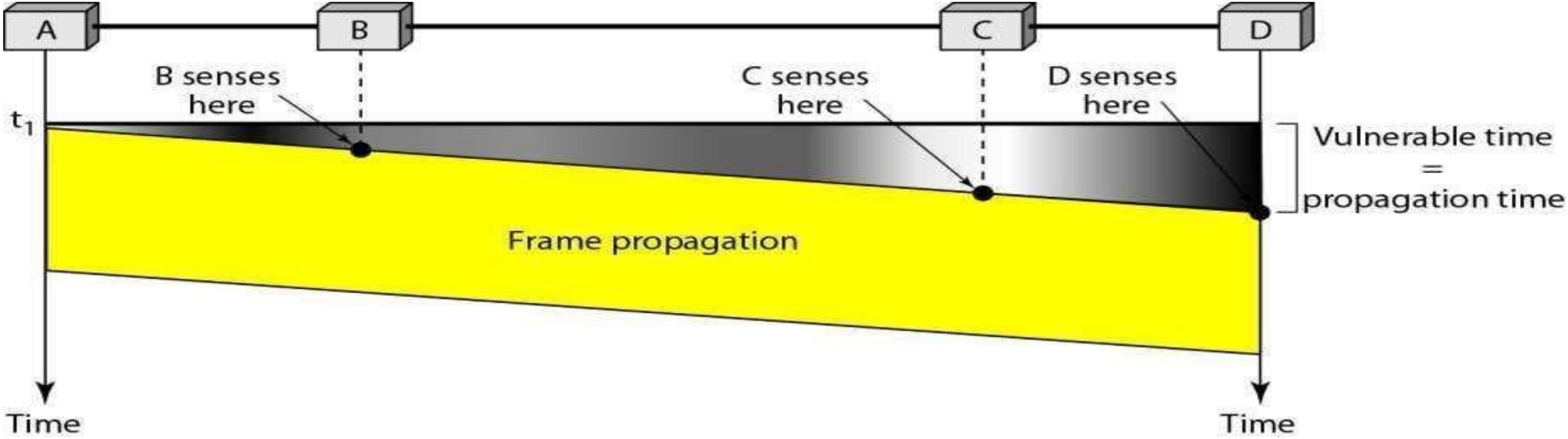
CSMA/CD

Carrier Sense Multiple Access with Collision Detection

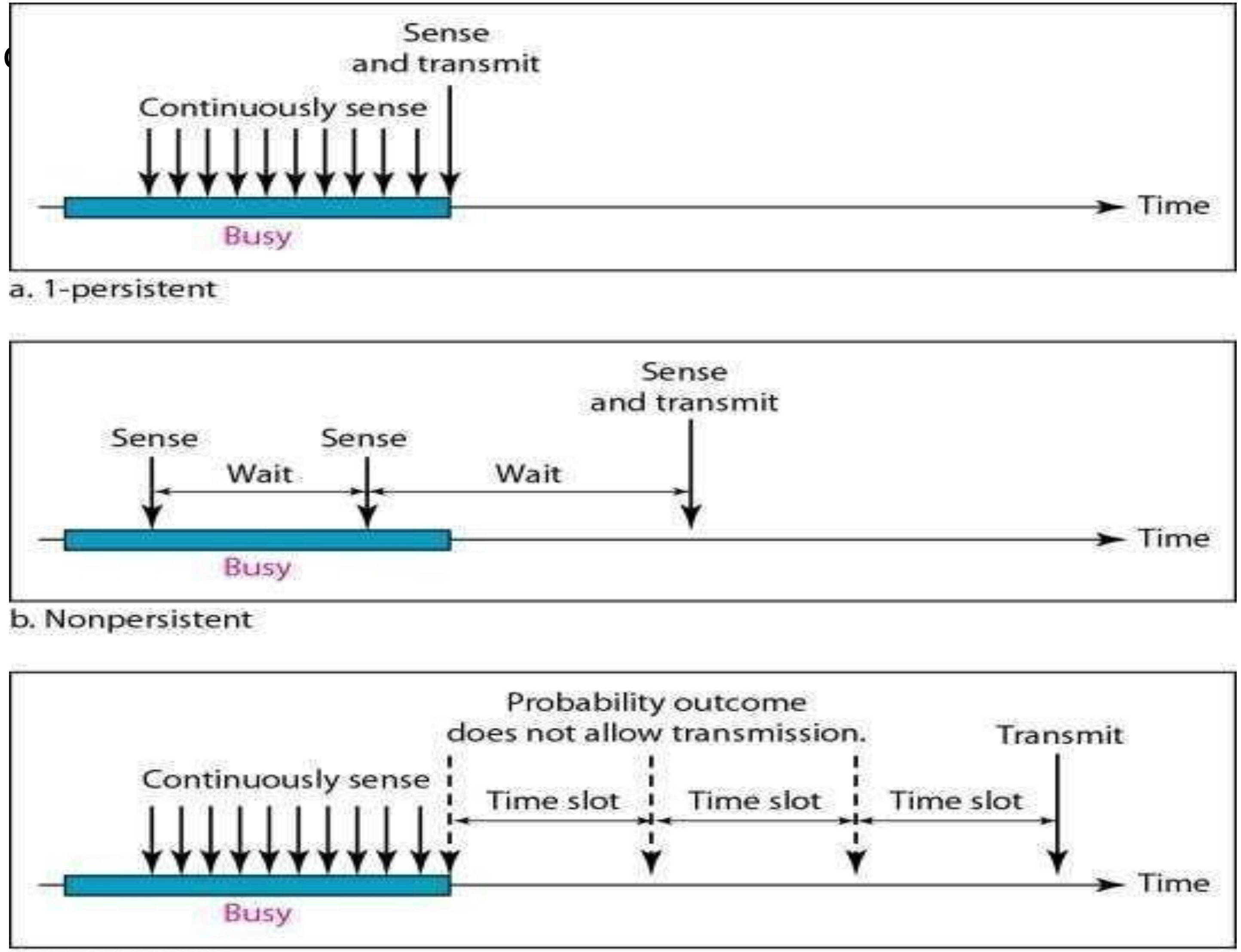
Space/time model of the collision in CSMA



Vulnerable time in CSMA



Behavior

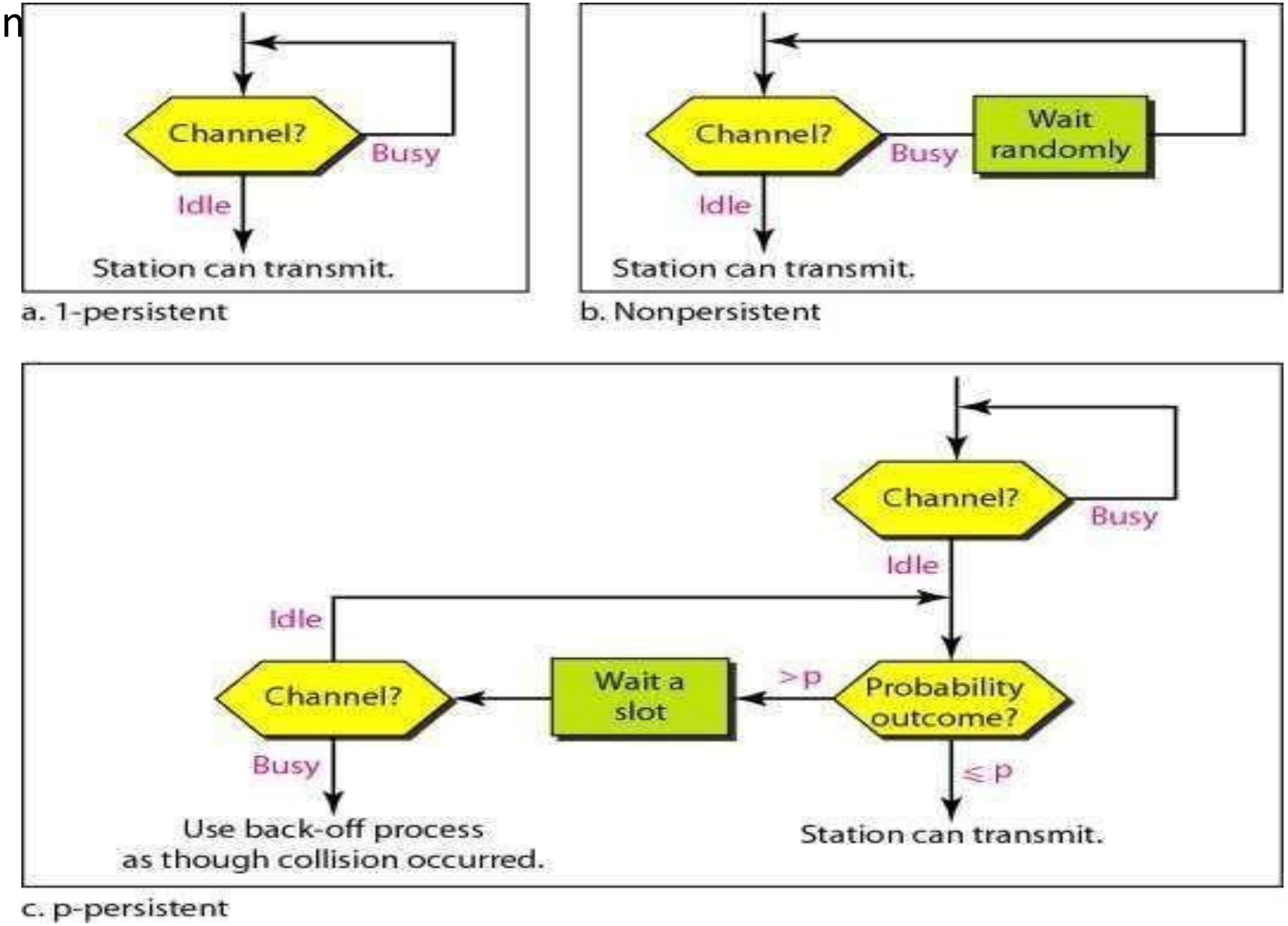


a. 1-persistent

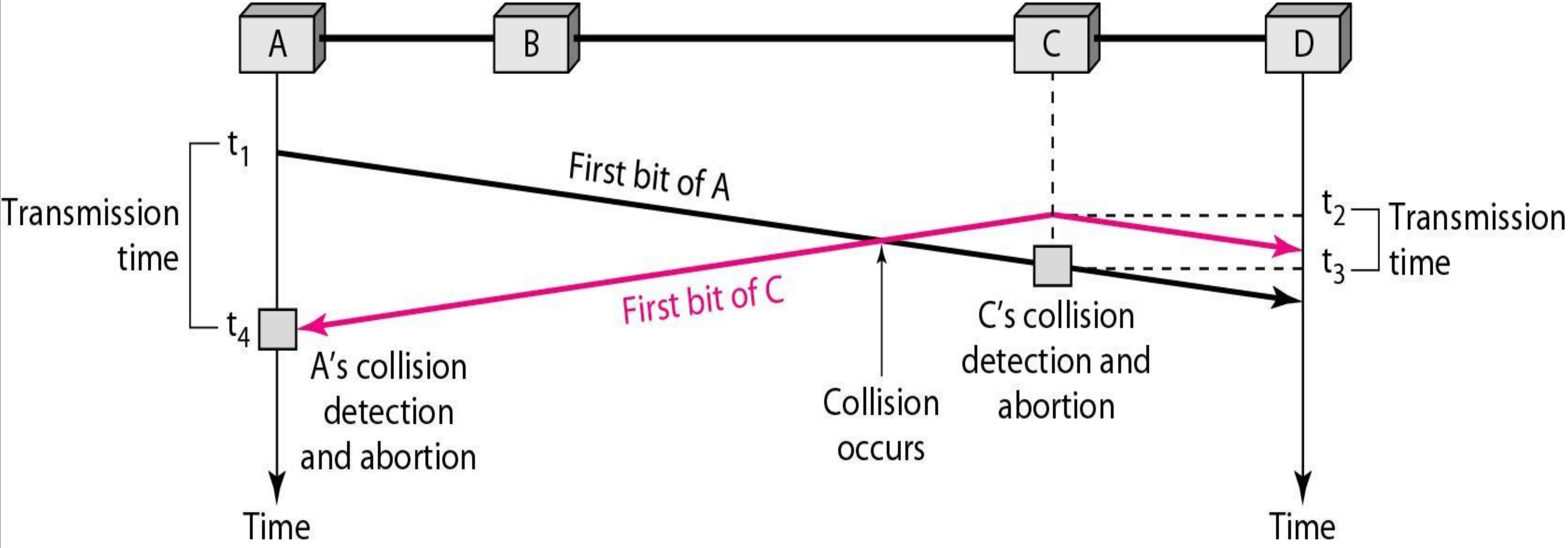
b. Nonpersistent

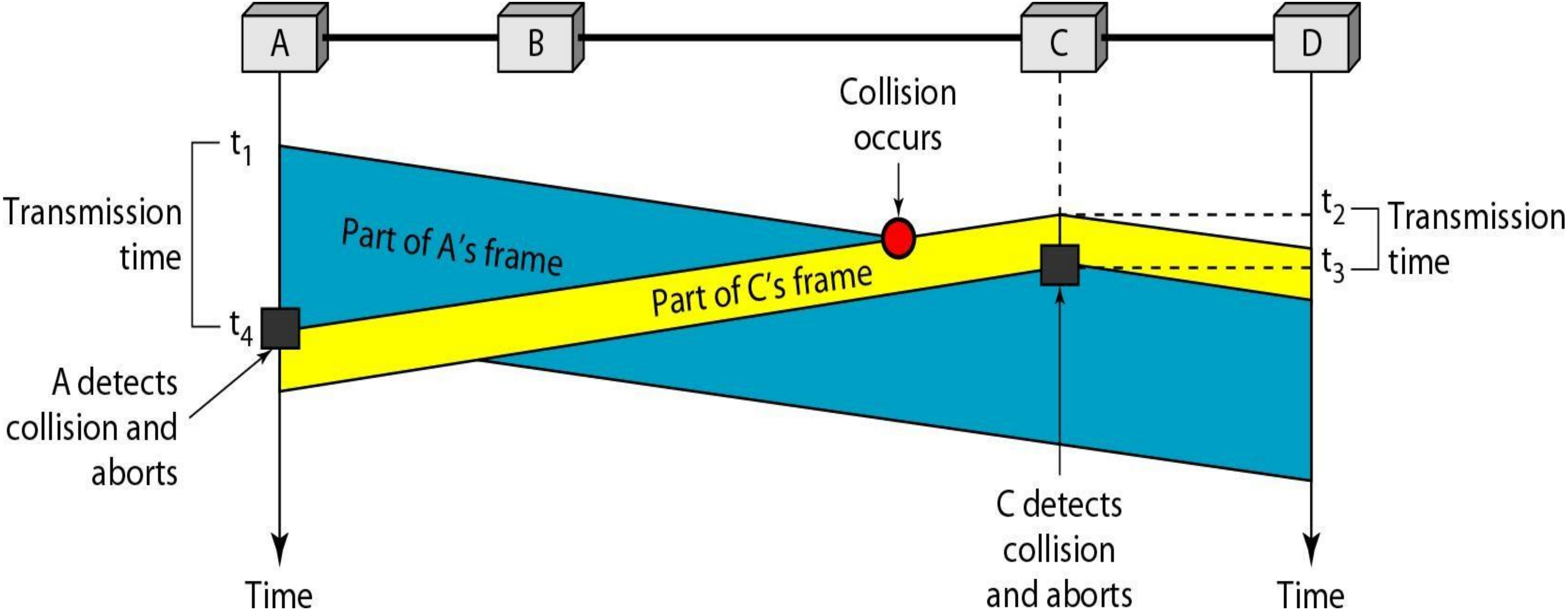
c. p-persistent

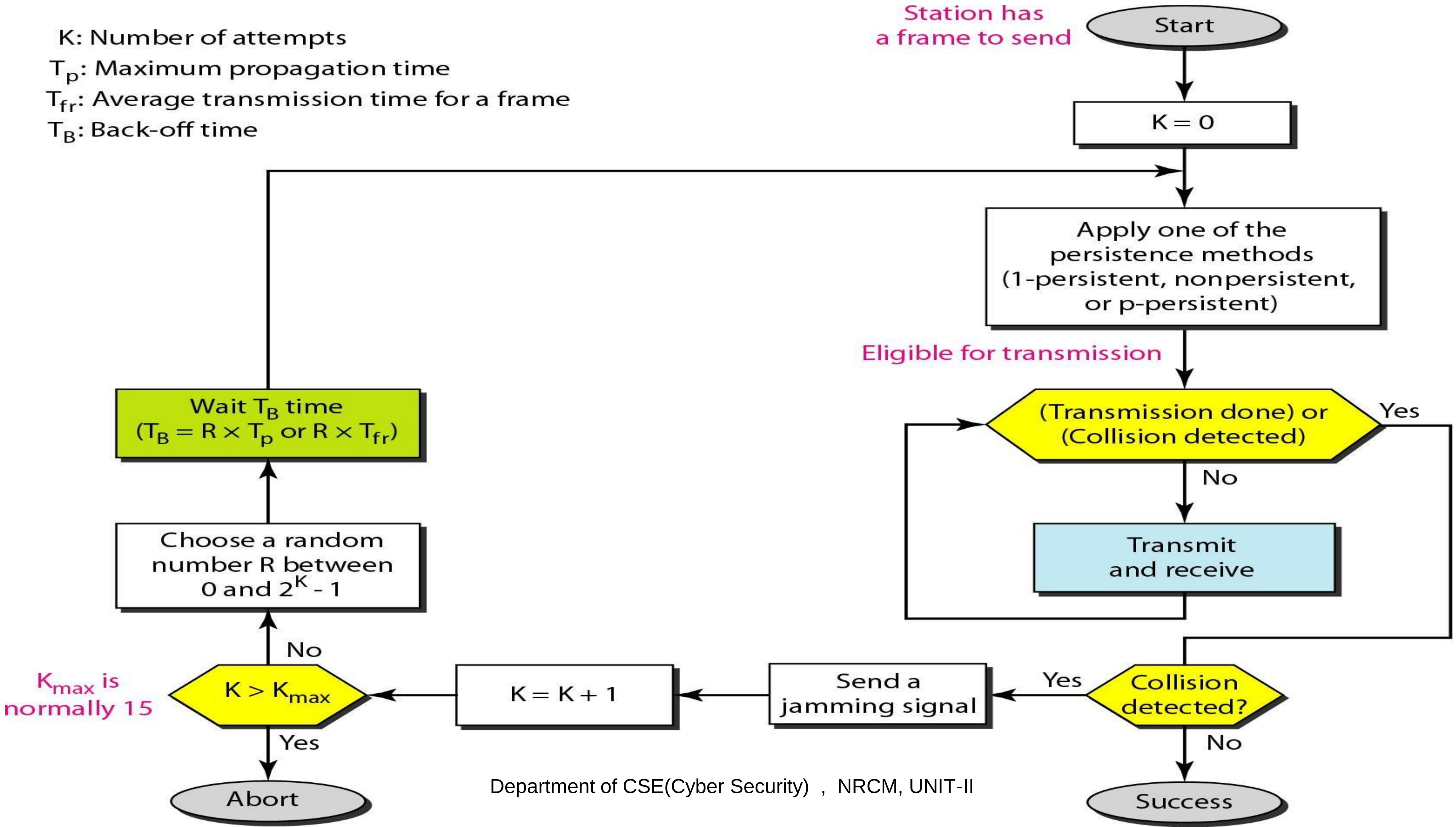
Flow diagran



Collision of the first bit in CSMA/CD




A
B
C
D
Collision occurs
t_1
Transmission time
t_4
Part of A's frame
Part of C's frame
t_2
t_3
Transmission time
A detects collision and aborts
C detects collision and aborts
Time
Time



K: Number of attempts

T_p: Maximum propagation time

T_{fr}: Average transmission time for a frame

T_B: Back-off time

Station has a frame to send

Start

$K = 0$

Apply one of the persistence methods (1-persistent, nonpersistent, or p-persistent)

Eligible for transmission

(Transmission done) or (Collision detected)

Yes

No

Transmit and receive

Collision detected?

Yes

No

Success

Send a jamming signal

$K = K + 1$

$K > K_{max}$

K_{max} is normally 15

No

Yes

Abort

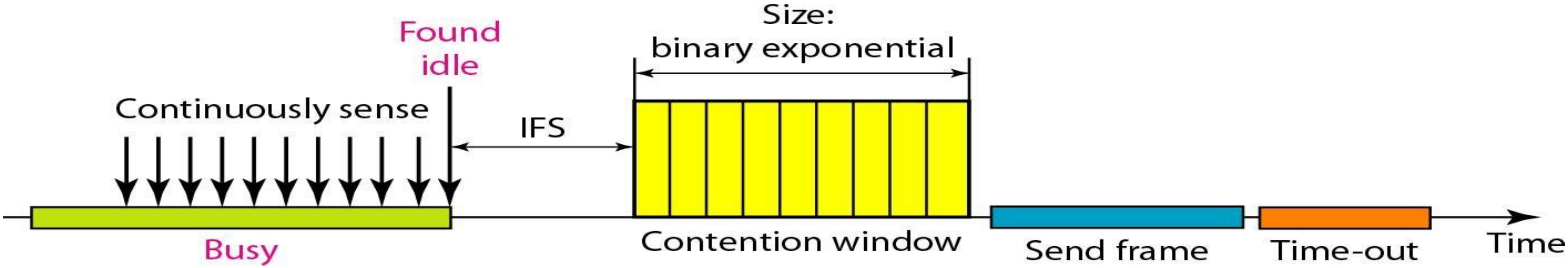
Choose a random number R between 0 and $2^K - 1$

Wait T_B time ($T_B = R \times T_p$ or $R \times T_{fr}$)

CSMA/CA

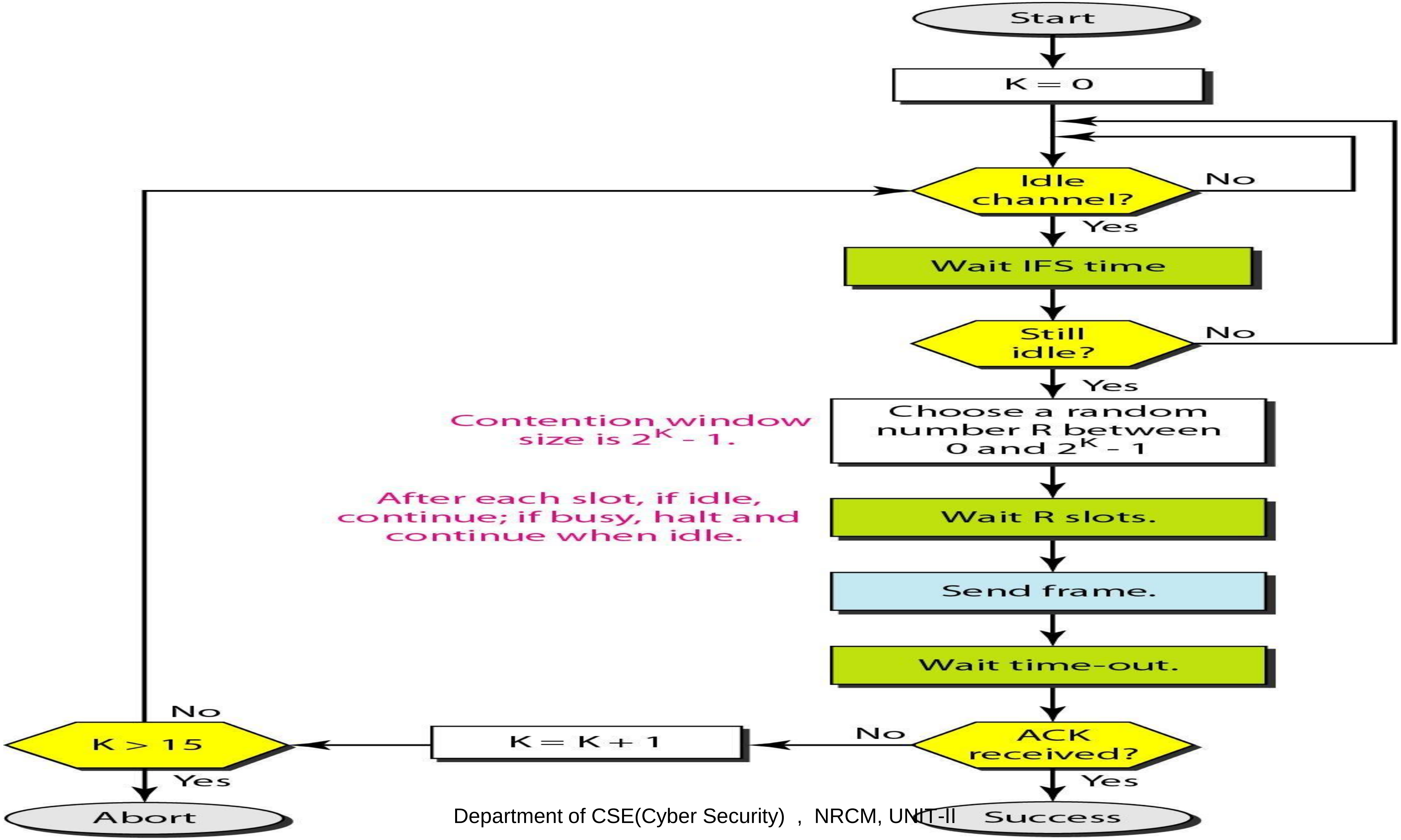
Carrier Sense Multiple Access with Collision Avoidance

Timing in CSMA/CA



Note

In CSMA/CA, the IFS can also be used to define the priority of a station or a frame.



Start

K = 0

Idle channel?

No

Yes

Wait IFS time

Still idle?

No

Yes

Choose a random number R between 0 and 2^K - 1

Contention window size is 2^K - 1.

After each slot, if idle, continue; if busy, halt and continue when idle.

Wait R slots.

Send frame.

Wait time-out.

ACK received?

No

Yes

K = K + 1

K > 15

No

Yes

Abort

Success

CONTROLLED ACCESS

CONTROLLED ACCESS

In controlled access, the stations consult one another to find which station has the right to send. A station cannot send unless it has been authorized by other stations.

In controlled access, the stations consult one another to find which station has the right to send. A station cannot send unless it has been authorized by other stations.

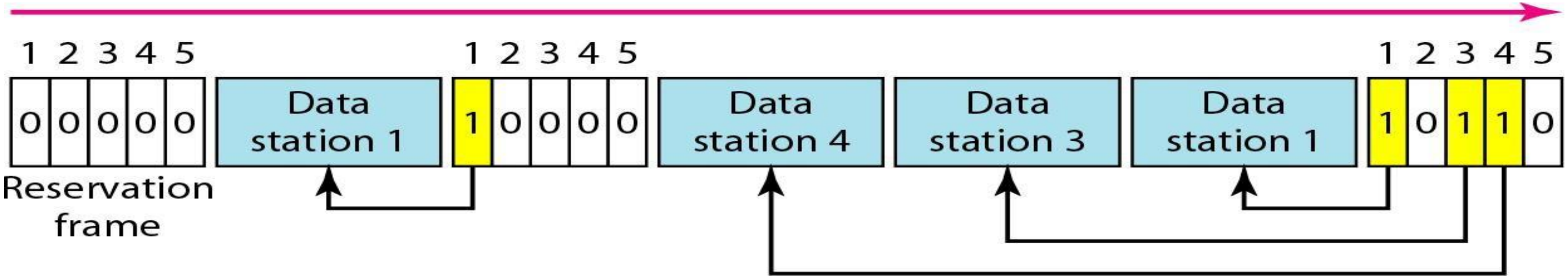
Topics discussed in this section:

Topics discussed in this section:

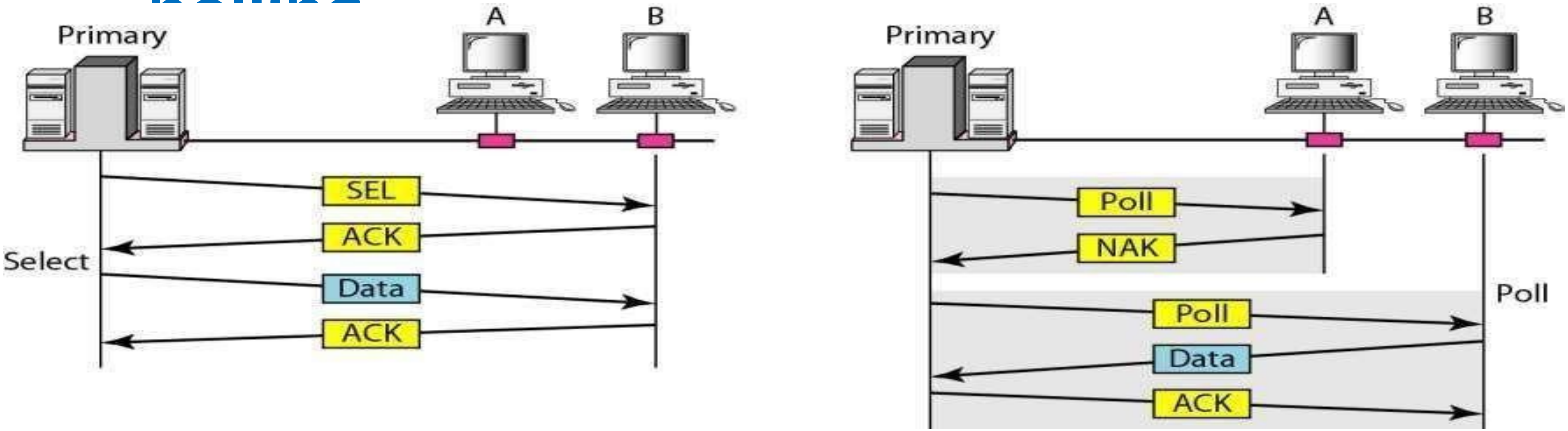
Reservation Polling

Token Passing

Reservation access method



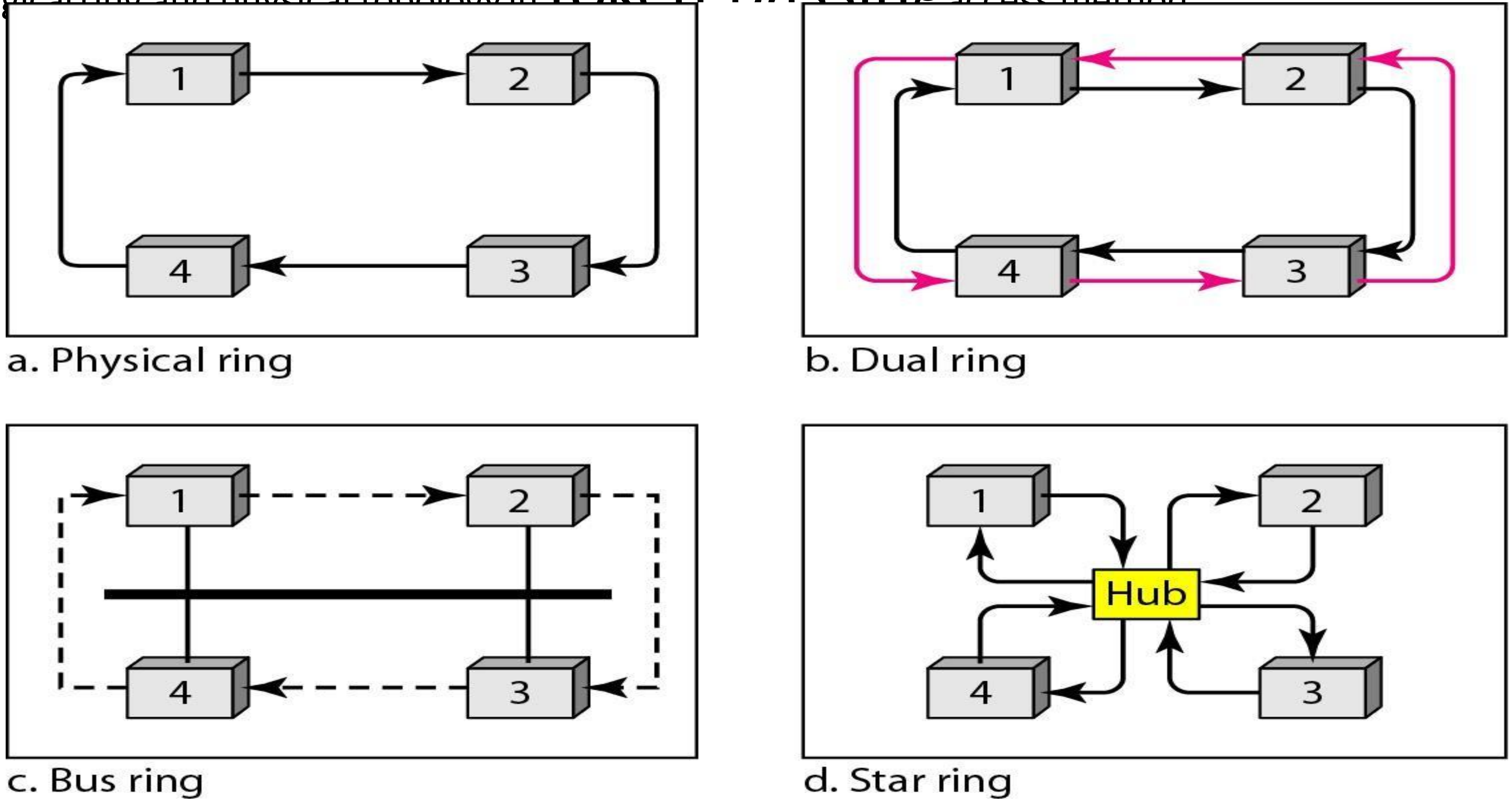
polling



Polling – one device as primary station and the other device as secondary station

Select – primary device wants to send data to secondary device, secondary device gets ready to receive
Poll – primary device solicits (ask) transmissions from secondary devices

Logical ring and physical topology in token-passing access method



a. Physical ring

b. Dual ring

c. Bus ring

d. Star ring

CHANNELIZATION

CHANNELIZATION

Channelization is a multiple-access method in which the available bandwidth of a link is shared in time, frequency, or through code, between different stations.

Channelization is a multiple-access method in which the available bandwidth of a link is shared in time, frequency, or through code, between different stations.

Topics discussed in this section:

Topics discussed in this section:

Frequency-Division Multiple Access (FDMA) Time-Division Multiple Access (TDMA) Code-Division Multiple Access (CDMA)

CHANNELIZATION - FDMA

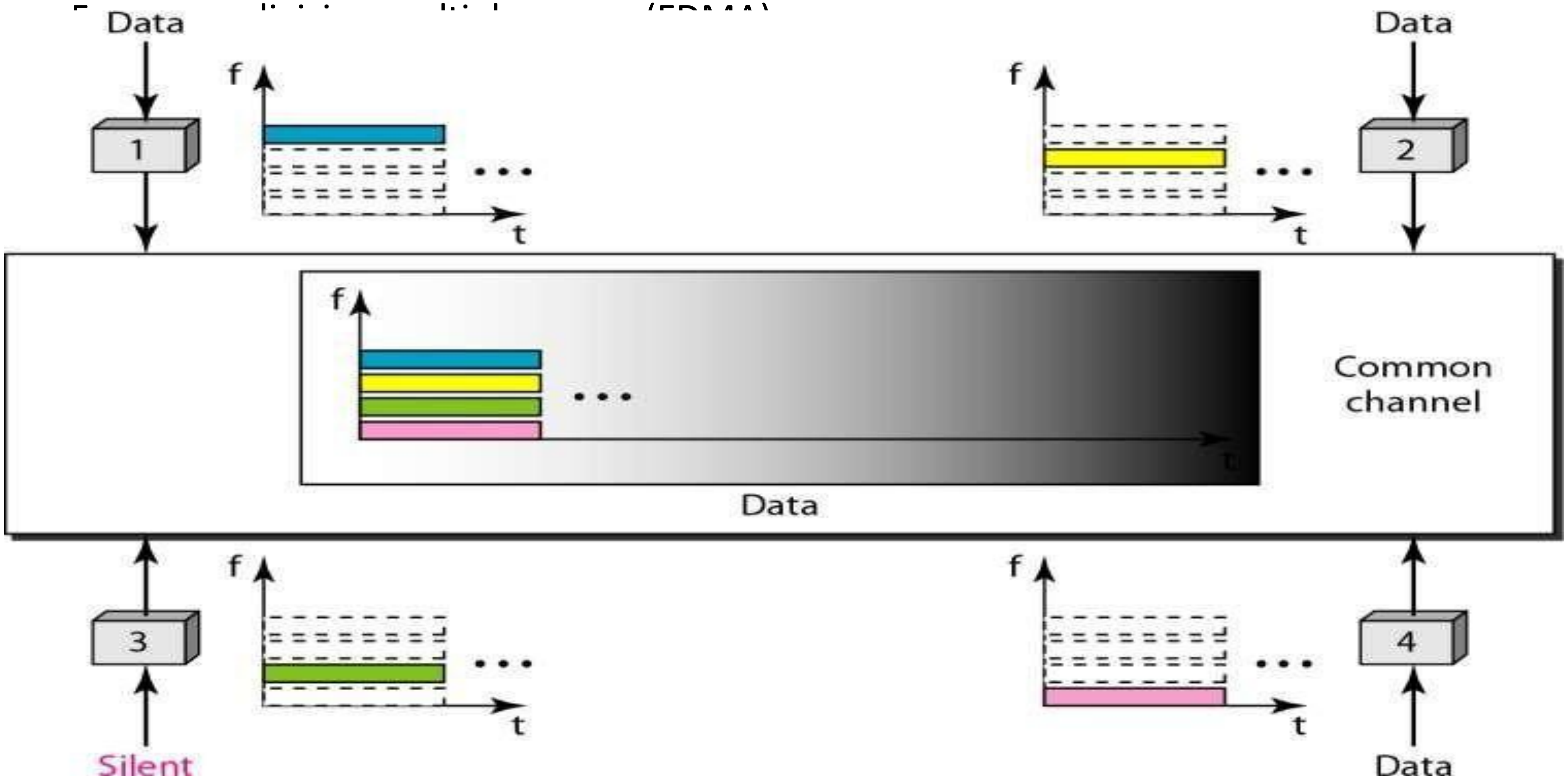
CHANNELIZATION - FDMA

- **FDMA:** Frequency Division Multiple Access:
  - Transmission medium is divided into **M** separate frequency bands
  - Each station transmits **continuously** on the assigned band at an average rate of **R/M**
  - A node is **limited** to an average rate equal **R/M** (where M is number of nodes) even when it is **the only node with frame** to be sent

Note

In FDMA, the available bandwidth of the common channel is divided into bands that are separated by guard bands.

Frequency-division multiple access (FDMA)



CHANNELIZATION - TDMA

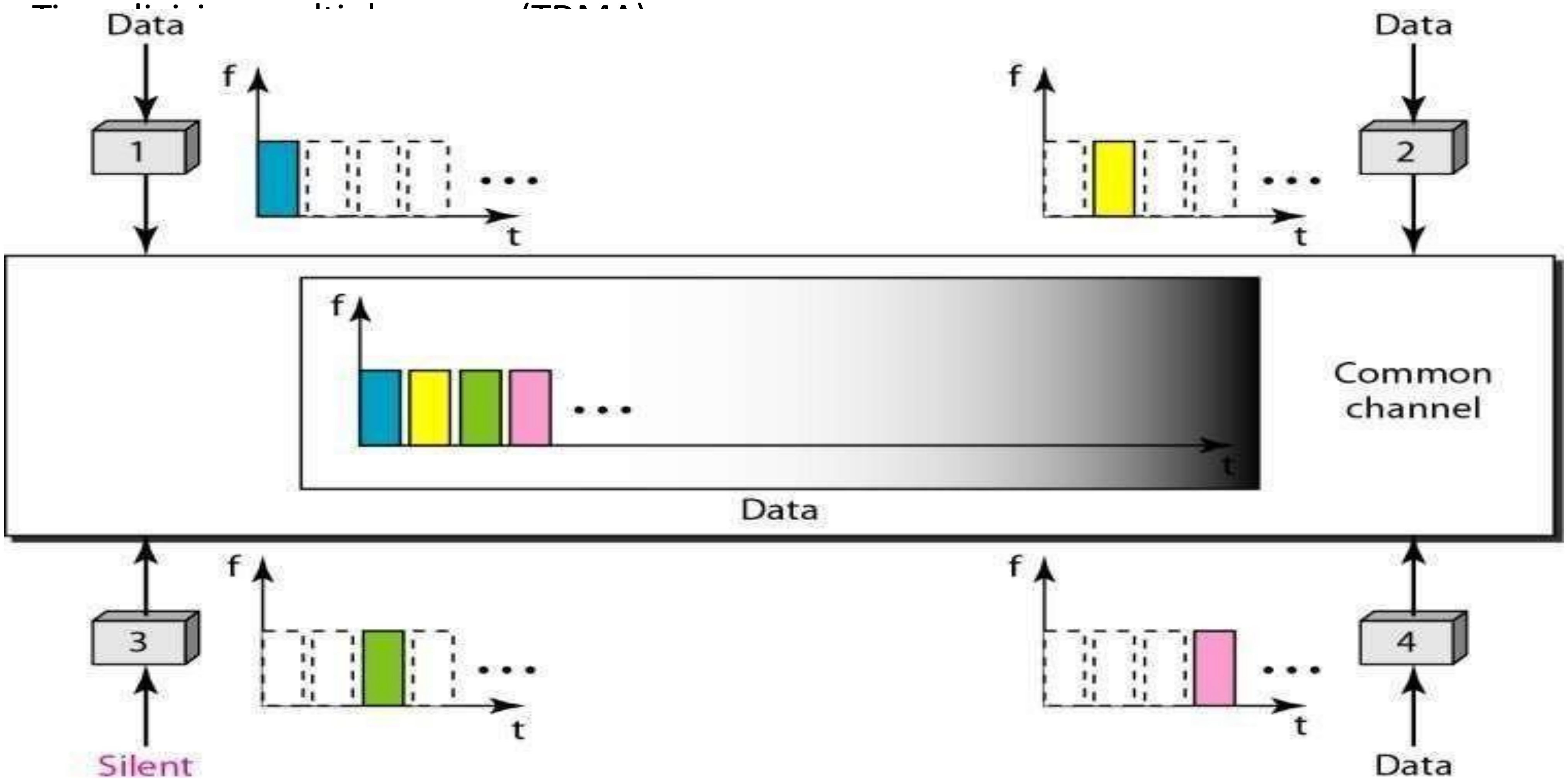
CHANNELIZATION - TDMA

- **TDMA: Time Division Multiple Access**
  - The entire bandwidth capacity is a **single channel** with its capacity shared **in time** between **M** stations
  - A node must **always wait for its turn** until its slot time arrives even when it is the **only node** with frames to send
  - A node is limited to an average rate equal **R/M** (where M is number of nodes) even when it is the only node with frame to be sent

Note

In TDMA, the bandwidth is just one channel that is timeshared between different stations.

Time-division multiple access (TDMA)



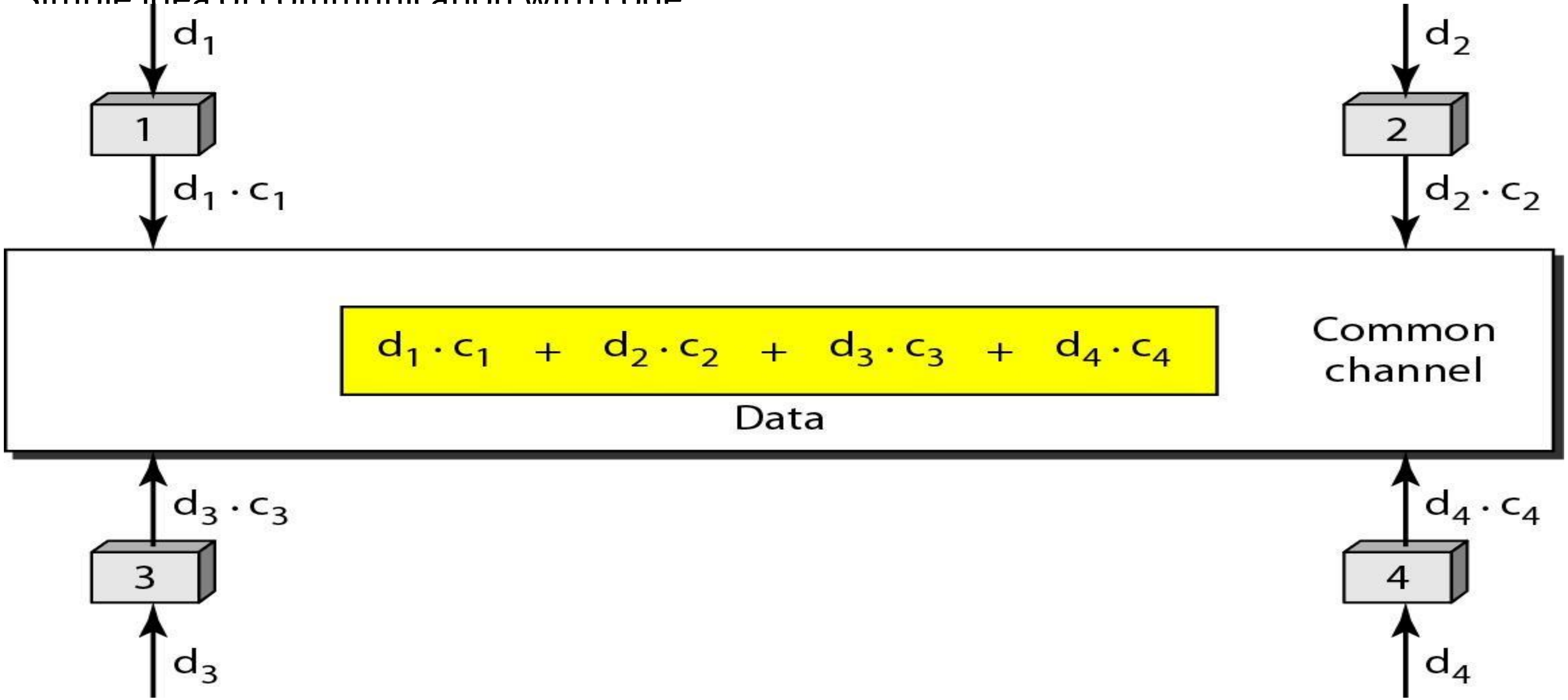
CHANNELIZATION - CDMA

- **CDMA: Code Division Multiple Access**
  - In CDMA, **<u>one channel</u>** carries all transmissions **simultaneously**
  - Each station codes its data signal by a specific codes before transmission
  - The stations receivers use these codes to recover the data for the desired station

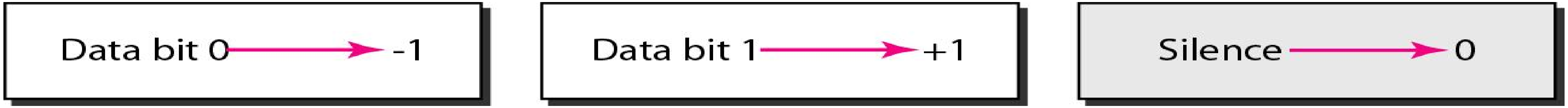
Note

In CDMA, one channel carries all transmissions simultaneously.

Simple idea of communication with code



Data representation in CDMA

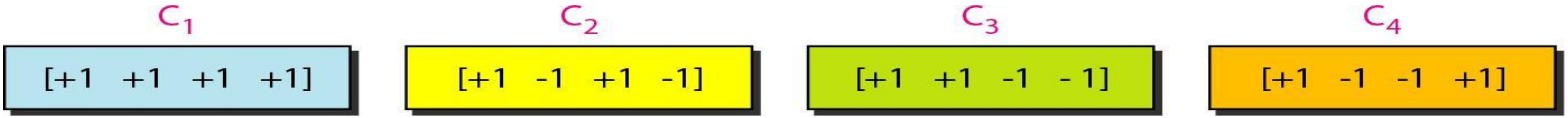


Data bit 0 → -1

Data bit 1 → +1

Silence → 0

Chip sequences



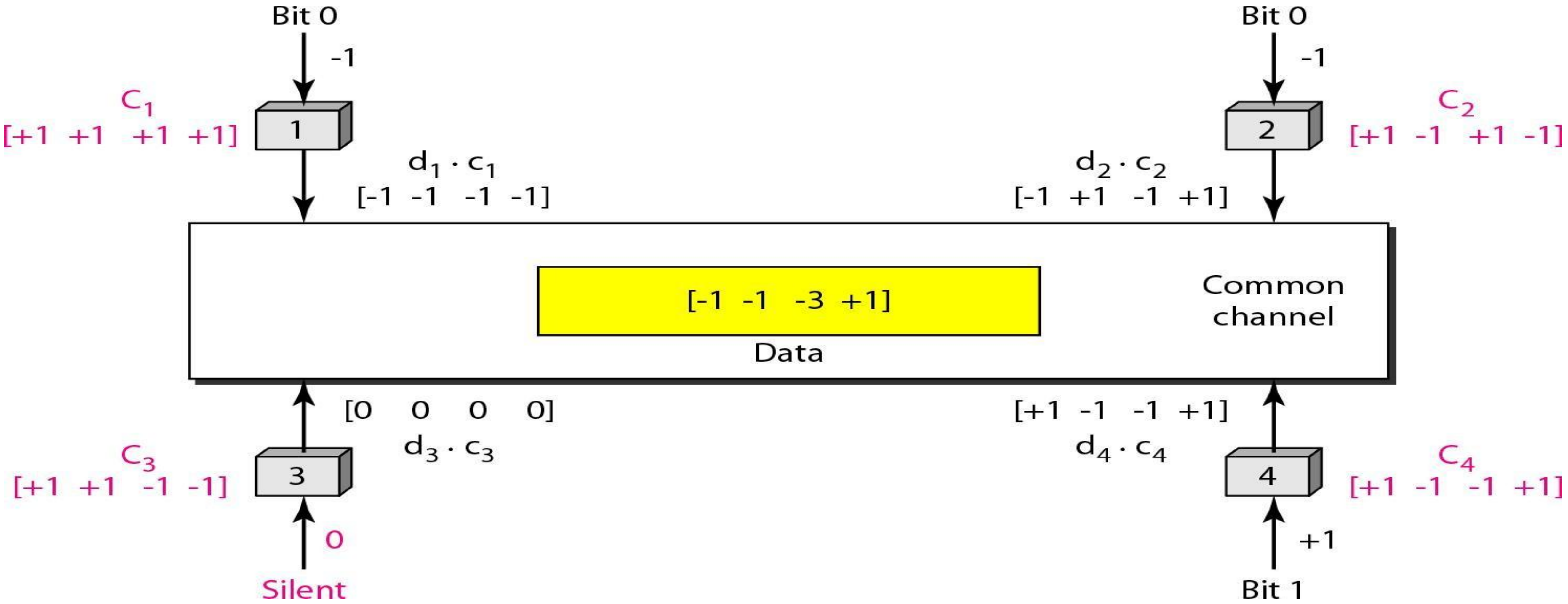
C_1 [+1 +1 +1 +1]

C_2 [+1 -1 +1 -1]

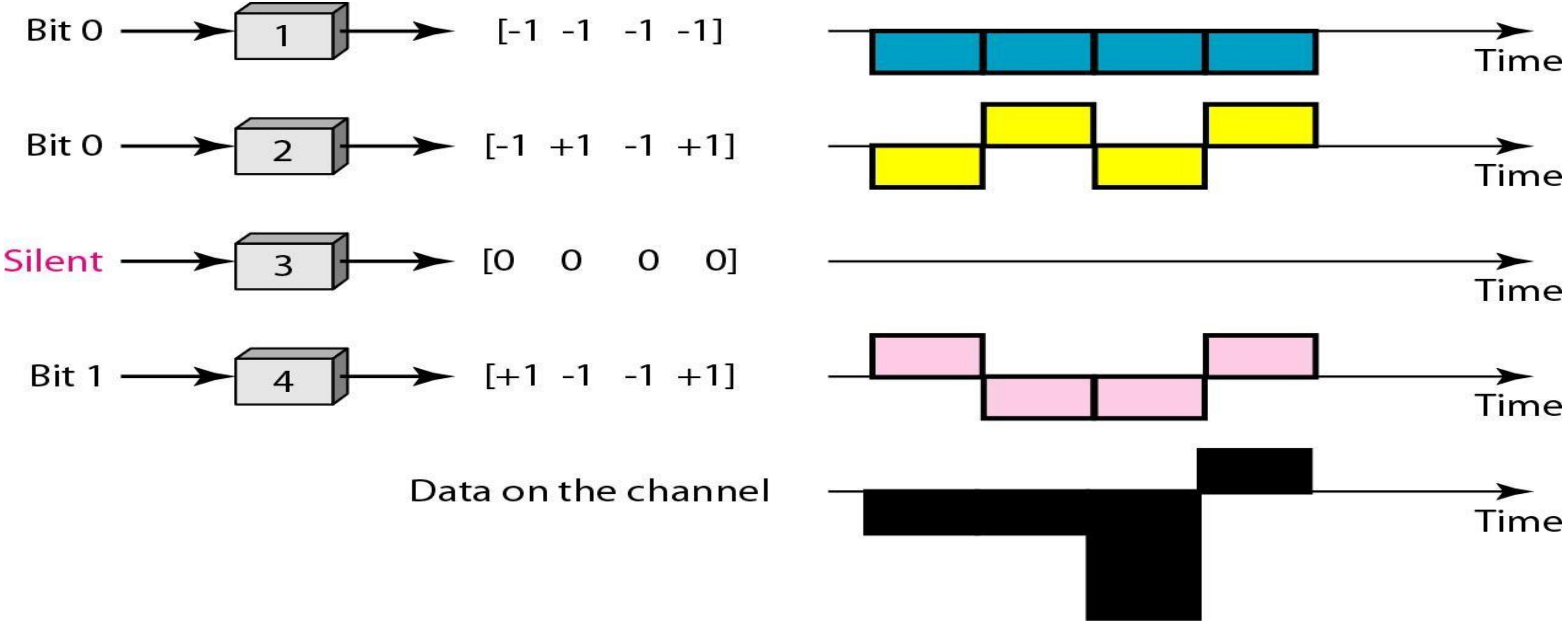
C_3 [+1 +1 -1 - 1]

C_4 [+1 -1 -1 +1]

Sharing channel in CDMA

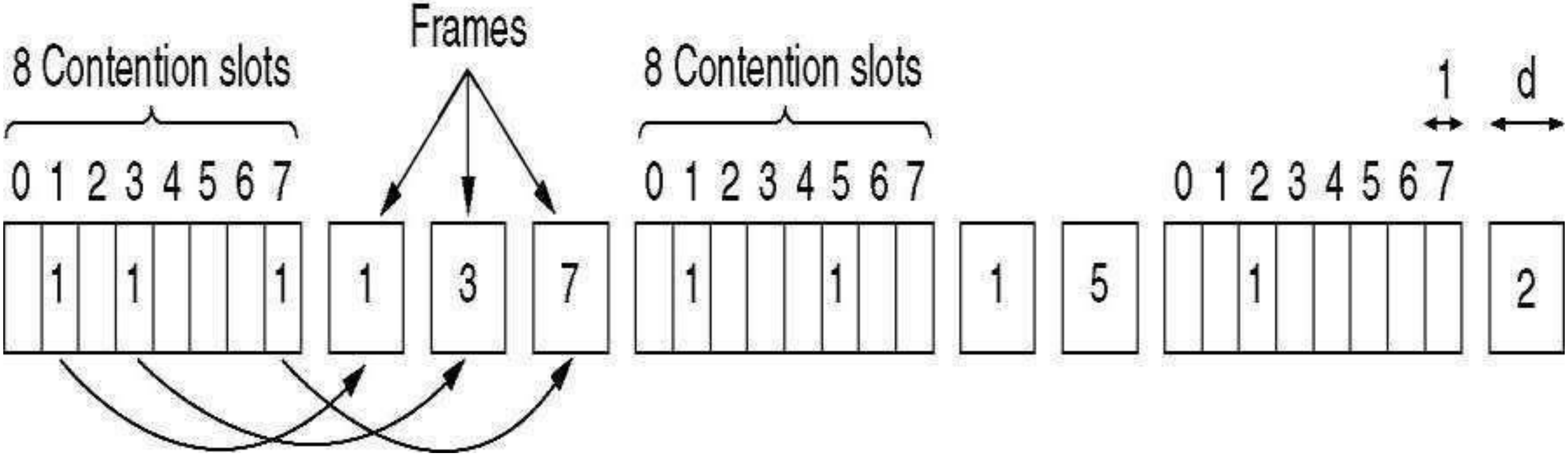


Digital signal created by four stations in CDMA

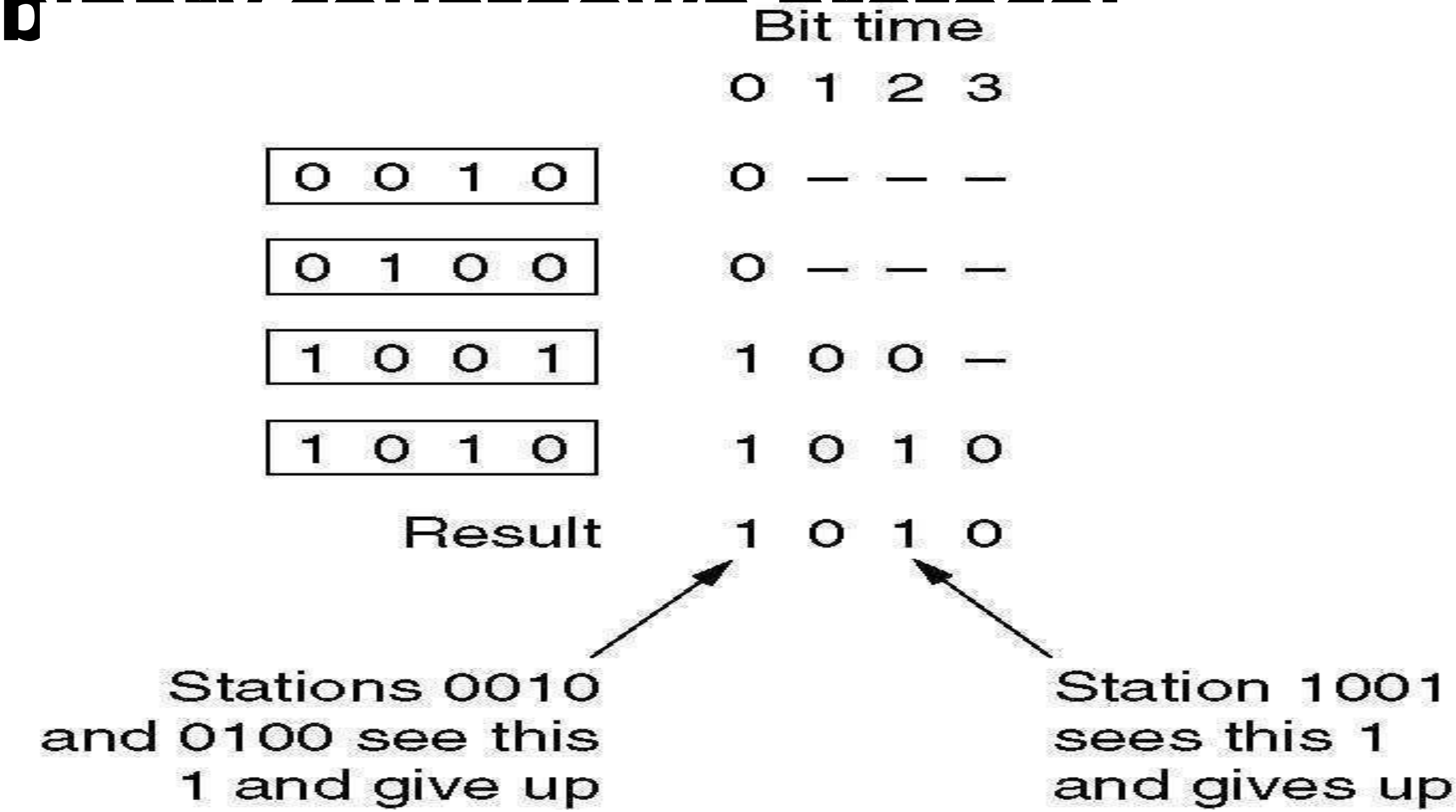


Collision-Free Protocols

A Bit-Map Protocol



binary countdown protocol



	Bit time 0	1	2	3
0 0 1 0	0	—	—	—
0 1 0 0	0	—	—	—
1 0 0 1	1	0	0	—
1 0 1 0	1	0	1	0
Result	1	0	1	0

Stations 0010 and 0100 see this 1 and give up

Station 1001 sees this 1 and gives up

Chapter 13 Wired

LANs: Ethernet

IEEE STANDARDS

IEEE 802.1 INTERNET WORKING

IEEE 802.3 Ethernet IEEE 802.4

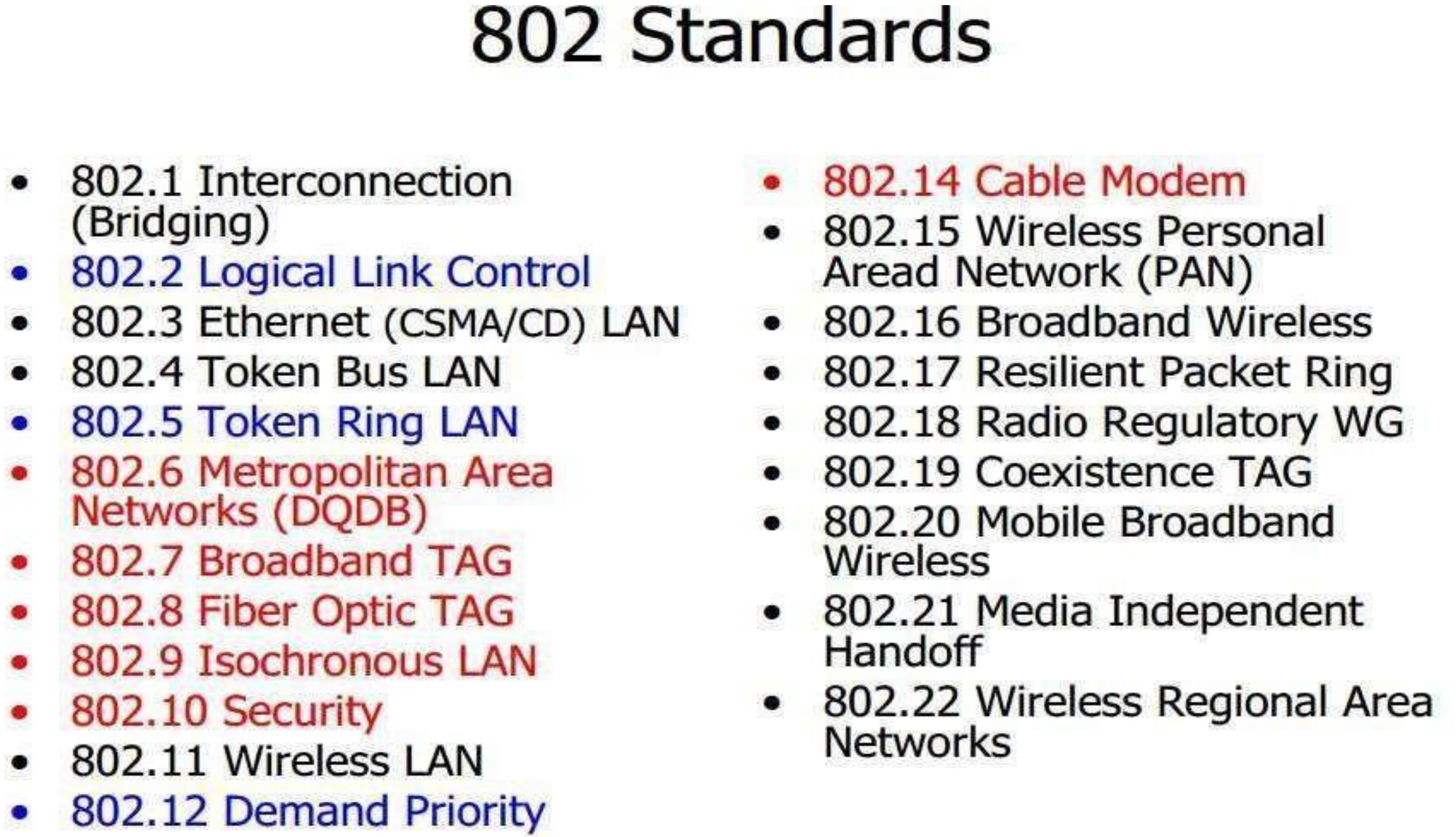
Token bus IEEE 802.5 Token Ring

IEEE 802.6 Metropolitan Area Networks

IEEE 802.11 Wireless LAN IEEE 802.15

Wireless PAN

IEEE 802.15.1 (Bluetooth certification)



802 Standards

- 802.1 Interconnection (Bridging)
- 802.2 Logical Link Control
- 802.3 Ethernet (CSMA/CD) LAN
- 802.4 Token Bus LAN
- 802.5 Token Ring LAN
- 802.6 Metropolitan Area Networks (DQDB)
- 802.7 Broadband TAG
- 802.8 Fiber Optic TAG
- 802.9 Isochronous LAN
- 802.10 Security
- 802.11 Wireless LAN
- 802.12 Demand Priority
- 802.14 Cable Modem
- 802.15 Wireless Personal Aread Network (PAN)
- 802.16 Broadband Wireless
- 802.17 Resilient Packet Ring
- 802.18 Radio Regulatory WG
- 802.19 Coexistence TAG
- 802.20 Mobile Broadband Wireless
- 802.21 Media Independent Handoff
- 802.22 Wireless Regional Area Networks

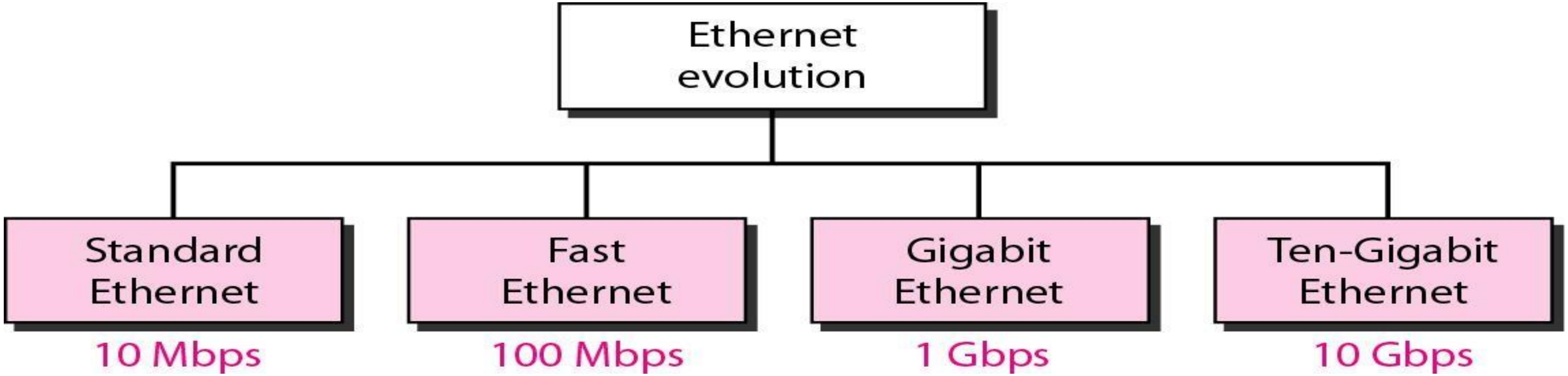
STANDARD ETHERNET(802.3)

The original Ethernet was created in 1976 at Xerox's Palo Alto Research Center (PARC). Since then, it has gone through four generations. We briefly discuss the Standard (or traditional) Ethernet in this section.

Topics discussed in this section:

MAC Sublayer Physical Layer

Figure 13.3 *Ethernet evolution through four generations*



- **STANDARD ETHERNET**
- **FAST ETHERNET**
- **GIGABIT ETHERNET**
- **TEN-GIGABIT ETHERNET**

MAC Sub layer

Figure 13.4 *802.3 MAC frame*

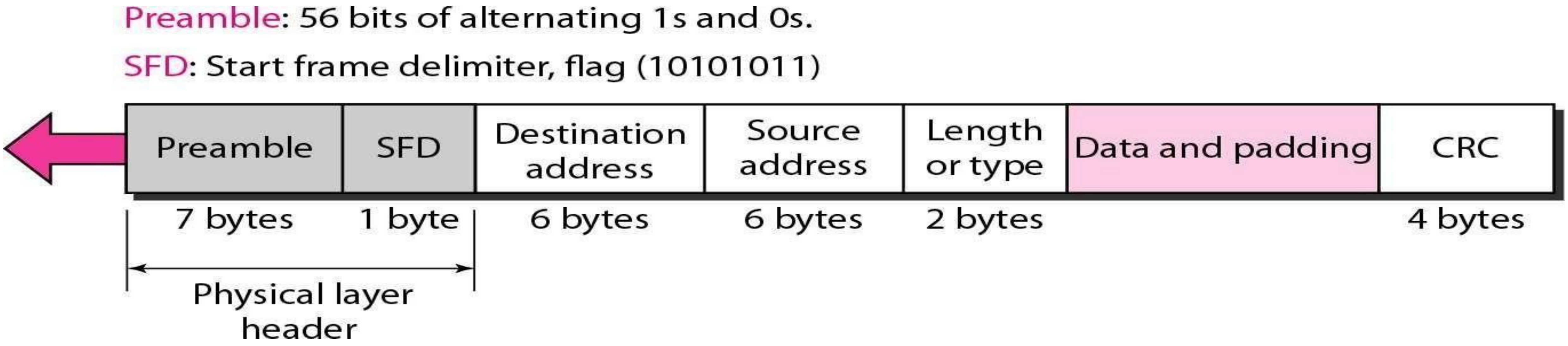


Figure 13.5 *Minimum and maximum lengths*

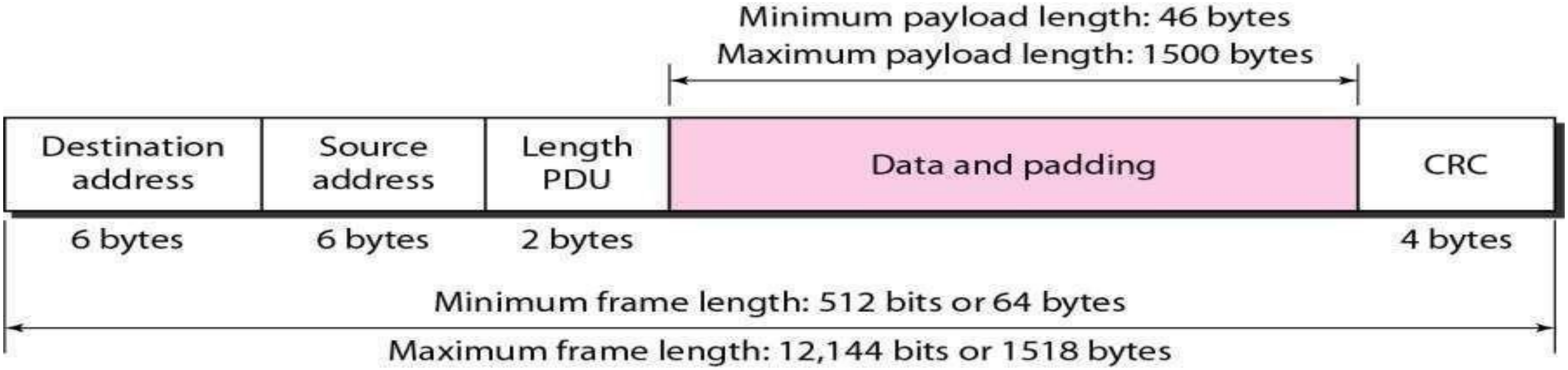
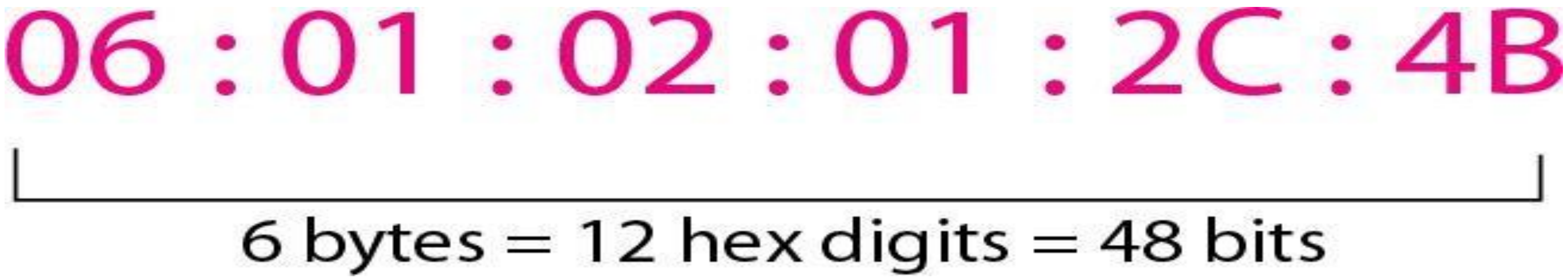

Minimum payload length: 46 bytes
Maximum payload length: 1500 bytes
Destination address
Source address
Length PDU
Data and padding
CRC
6 bytes
6 bytes
2 bytes
4 bytes
Minimum frame length: 512 bits or 64 bytes
Maximum frame length: 12,144 bits or 1518 bytes

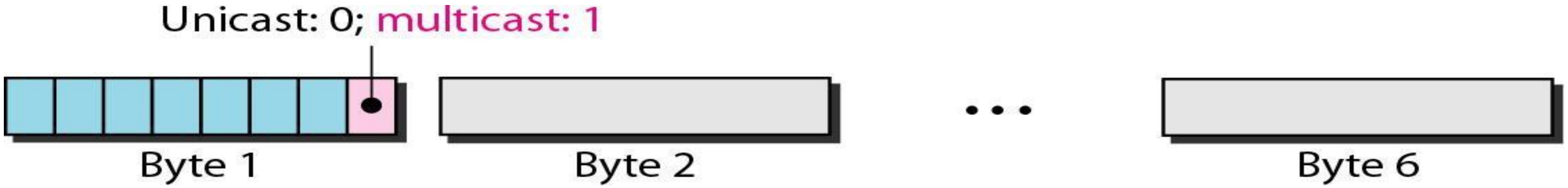
Figure 13.6 *Example of an Ethernet address in hexadecimal notation*



06 : 01 : 02 : 01 : 2C : 4B

6 bytes = 12 hex digits = 48 bits

Figure 13.7 *Unicast and multicast addresses*



Physical Layer

Categories of Standard Ethernet

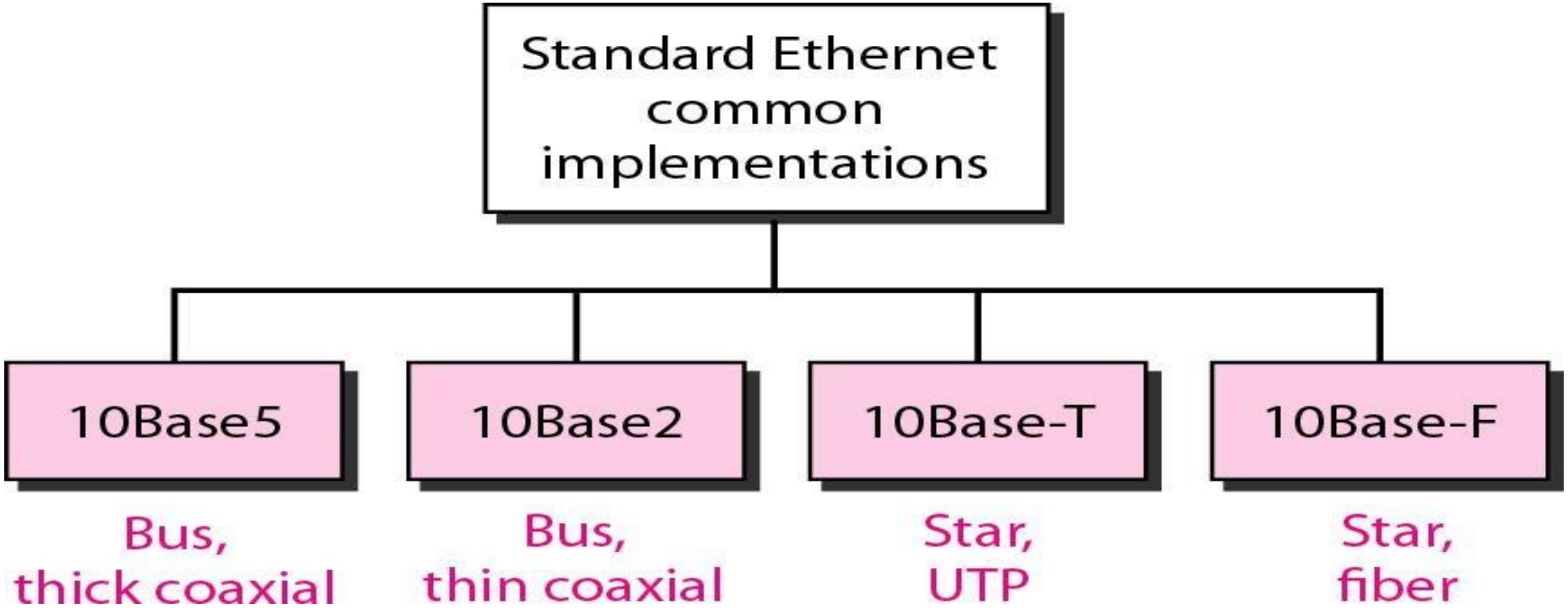
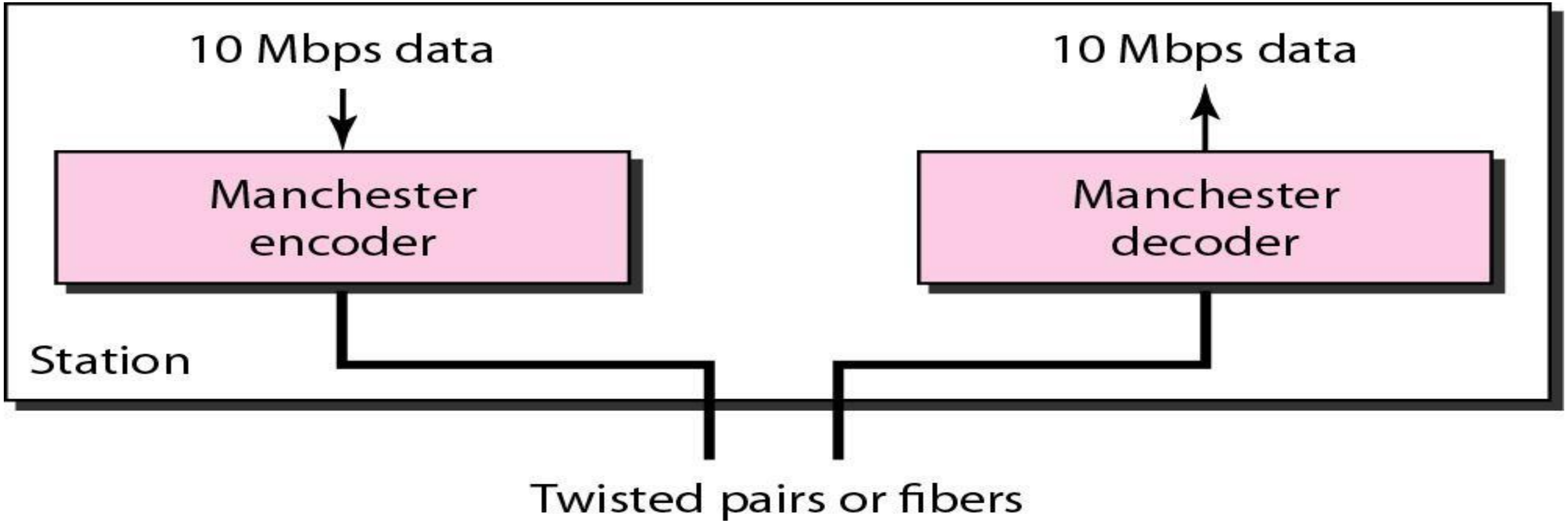
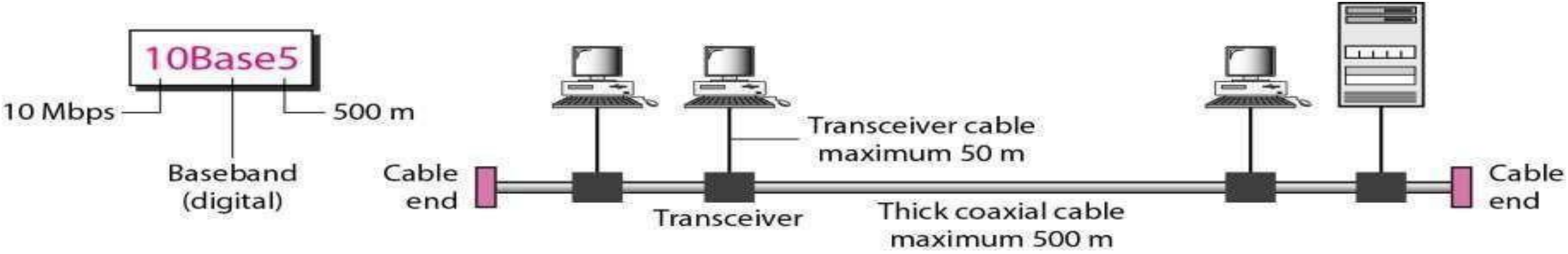


Figure 13.9 ***Encoding in a Standard Ethernet implementation***



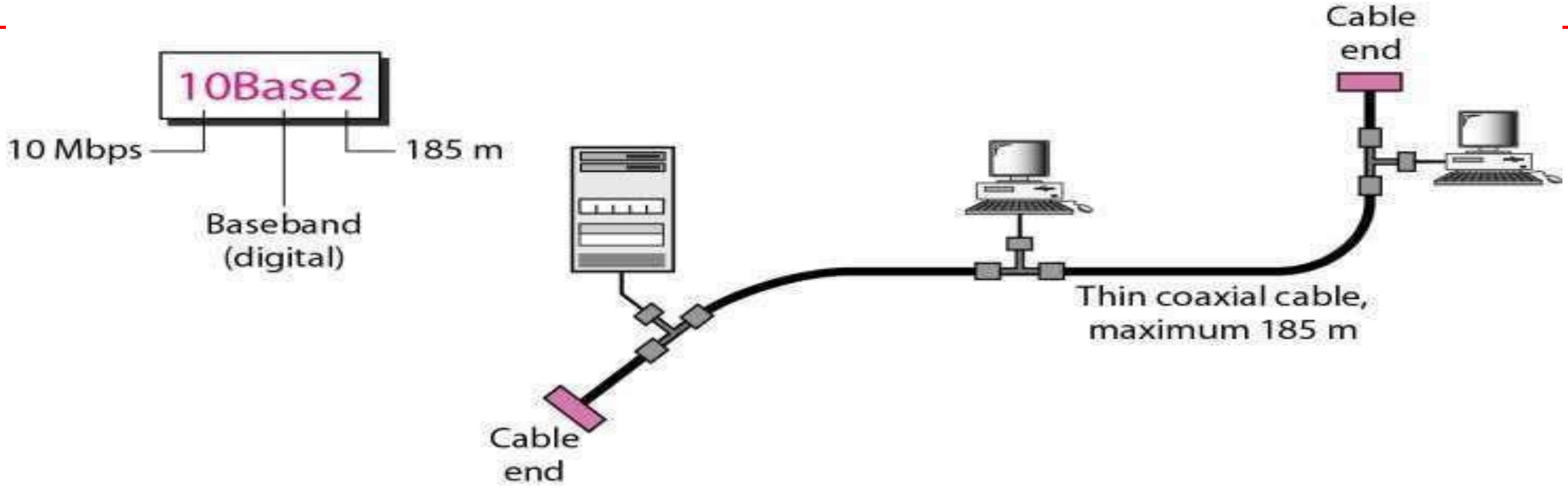
Use digital signaling (baseband) at 10Mbps

Figure 13.10 *10Base5 implementation*



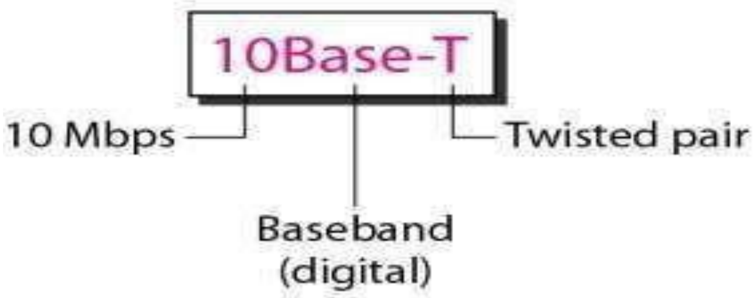
1. **Known as Thicknet**
2. **Thick coaxial cable**
3. **Uses bus topology with external transceiver**
4. **Max length of each segment 500m**

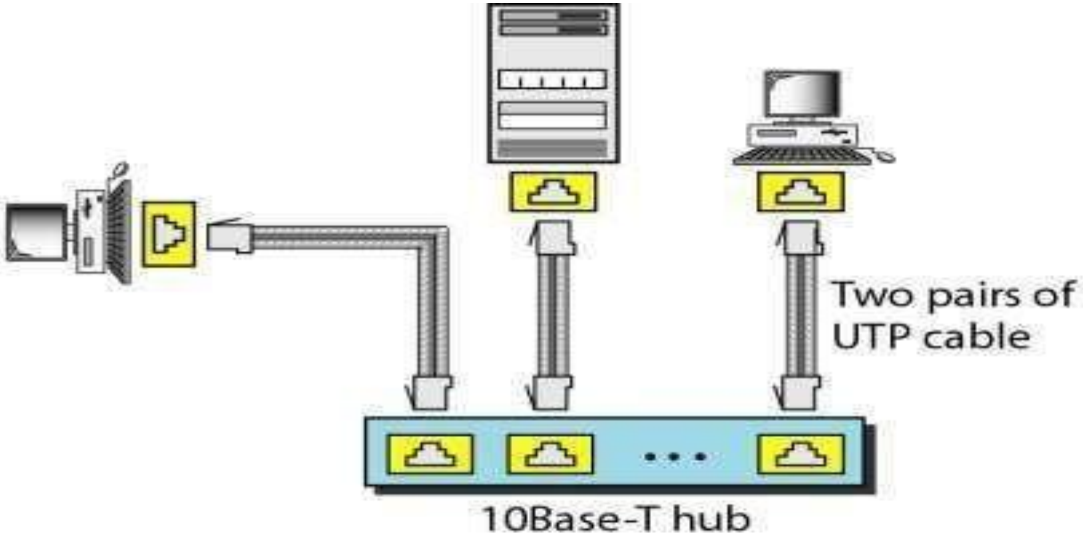
Figure 13.11 *10Base2 implementation*



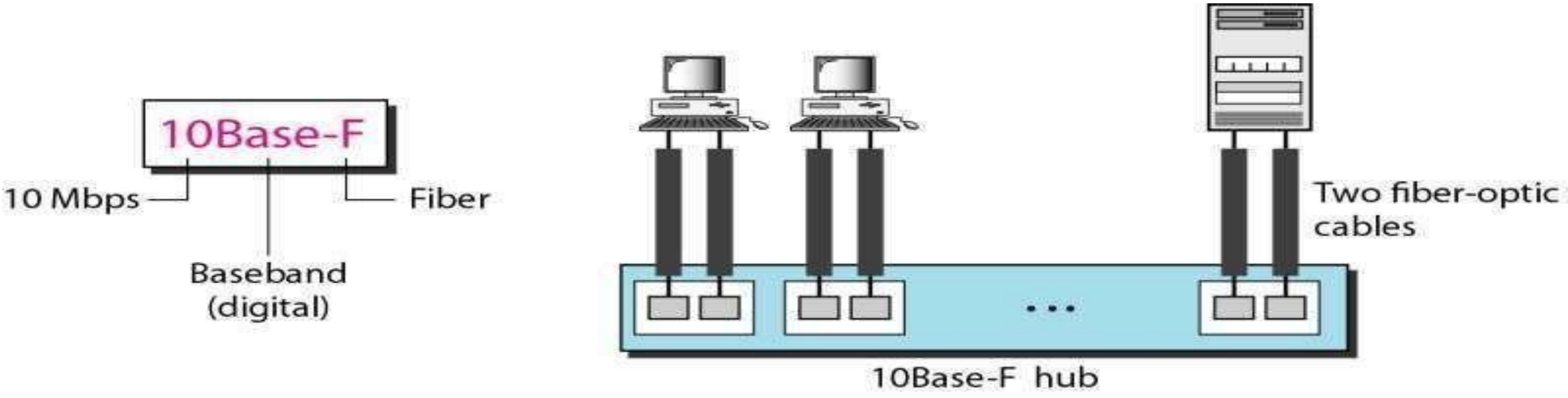
1. **Knows as Thin Ethernet**
2. **Uses bus topology with thin and flexible cable**
3. **Transceiver – part of NIC**
4. **Max length of each segment 185m**

10Base-T implementation



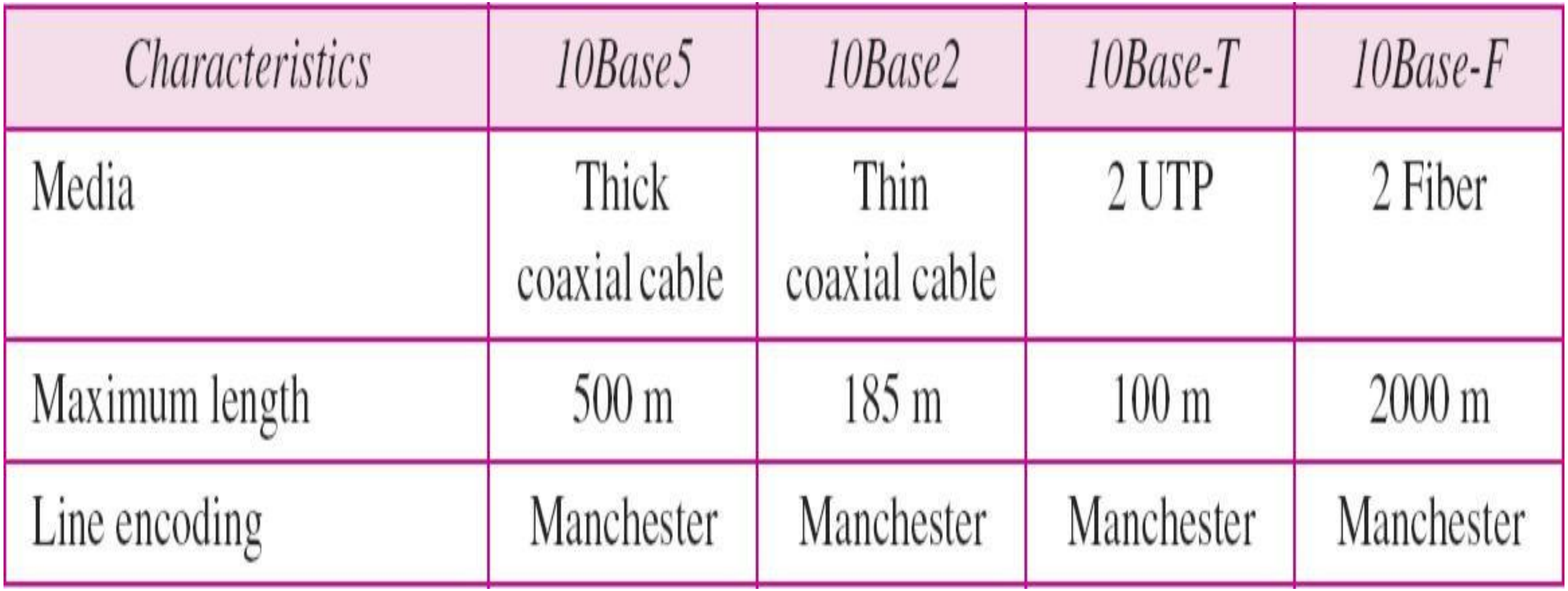


1. **Knows as twisted pair Ethernet**
2. **Uses physical star topology**
3. **Stations connected to hub**
4. **Max length 100m**



1. **Uses star topology**
2. **Stations connected to hub**

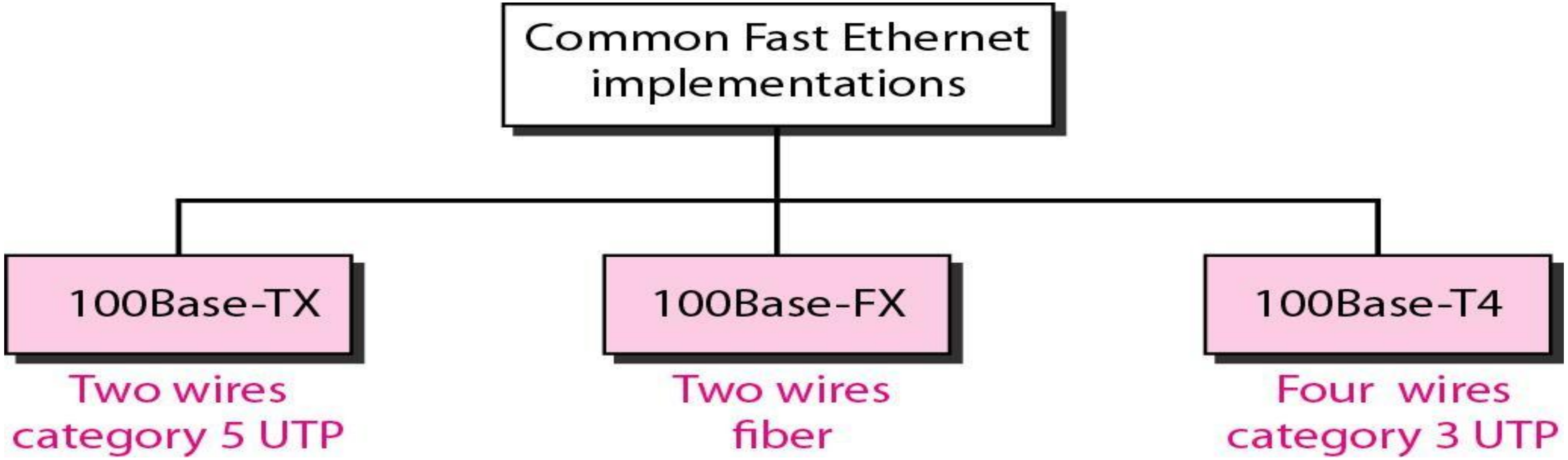
Table 13.1 ***Summary of Standard Ethernet implementations***

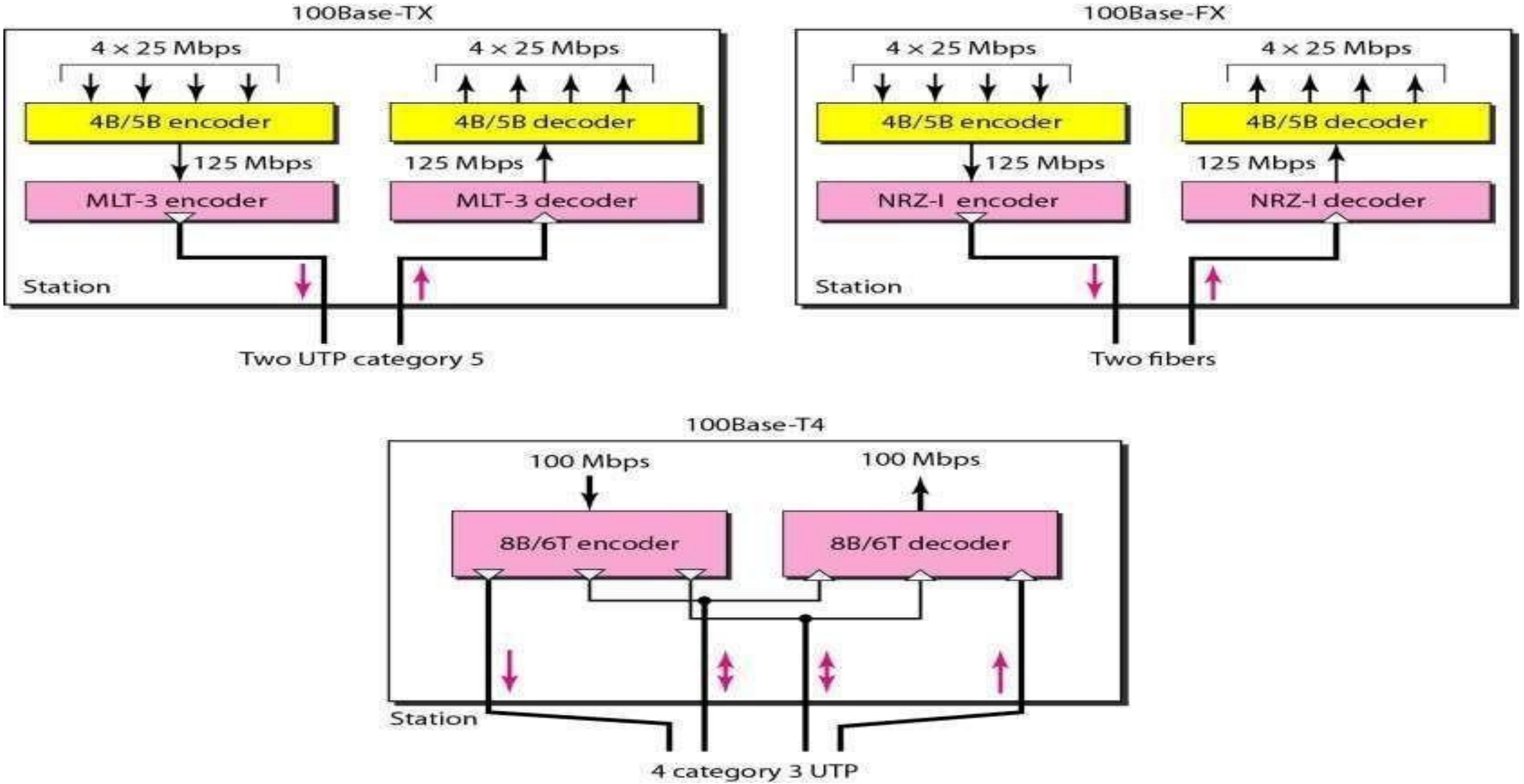


Characteristics	*10Base5*	*10Base2*	*10Base-T*	*10Base-F*
Media	Thick coaxial cable	Thin coaxial cable	2 UTP	2 Fiber
Maximum length	500 m	185 m	100 m	2000 m
Line encoding	Manchester	Manchester	Manchester	Manchester

FAST ETHERNET

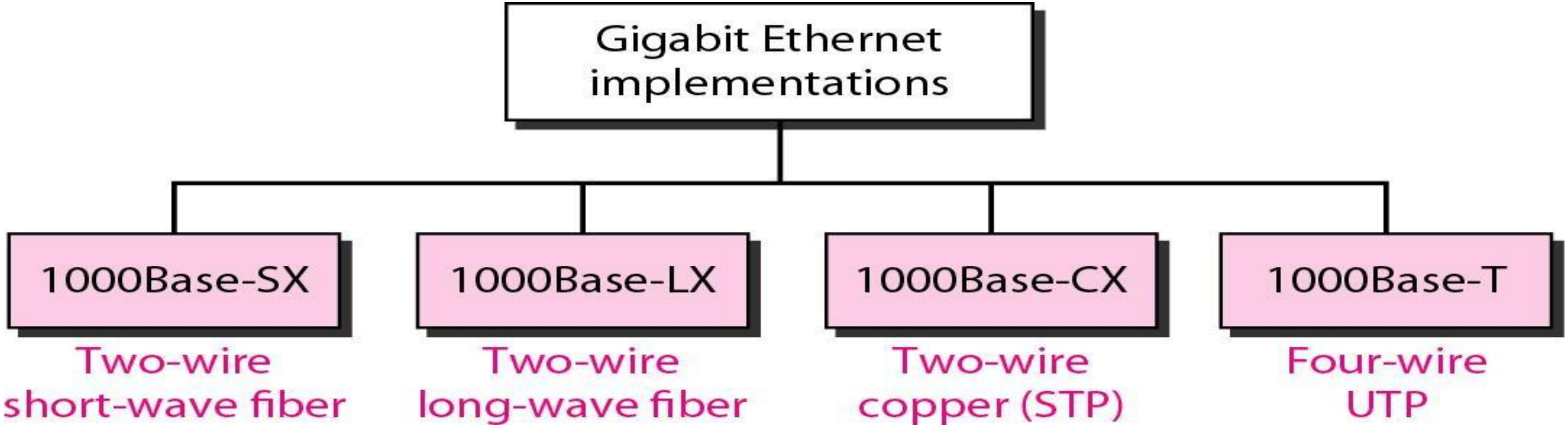
Figure 13.20 ***Fast Ethernet implementations***




100Base-TX
4 × 25 Mbps
4B/5B encoder
125 Mbps
MLT-3 encoder
4 × 25 Mbps
4B/5B decoder
125 Mbps
MLT-3 decoder
Station
Two UTP category 5
100Base-FX
4 × 25 Mbps
4B/5B encoder
125 Mbps
NRZ-I encoder
4 × 25 Mbps
4B/5B decoder
125 Mbps
NRZ-I decoder
Station
Two fibers
100Base-T4
100 Mbps
8B/6T encoder
100 Mbps
8B/6T decoder
Station
4 category 3 UTP

GIGABIT ETHERNET

Figure 13.23 *Gigabit Ethernet implementations*



Gigabit Ethernet implementations

1000Base-SX

Two-wire short-wave fiber

1000Base-LX

Two-wire long-wave fiber

1000Base-CX

Two-wire copper (STP)

1000Base-T

Four-wire UTP

Figure 13.24 *Encoding in Gigabit Ethernet implementations*

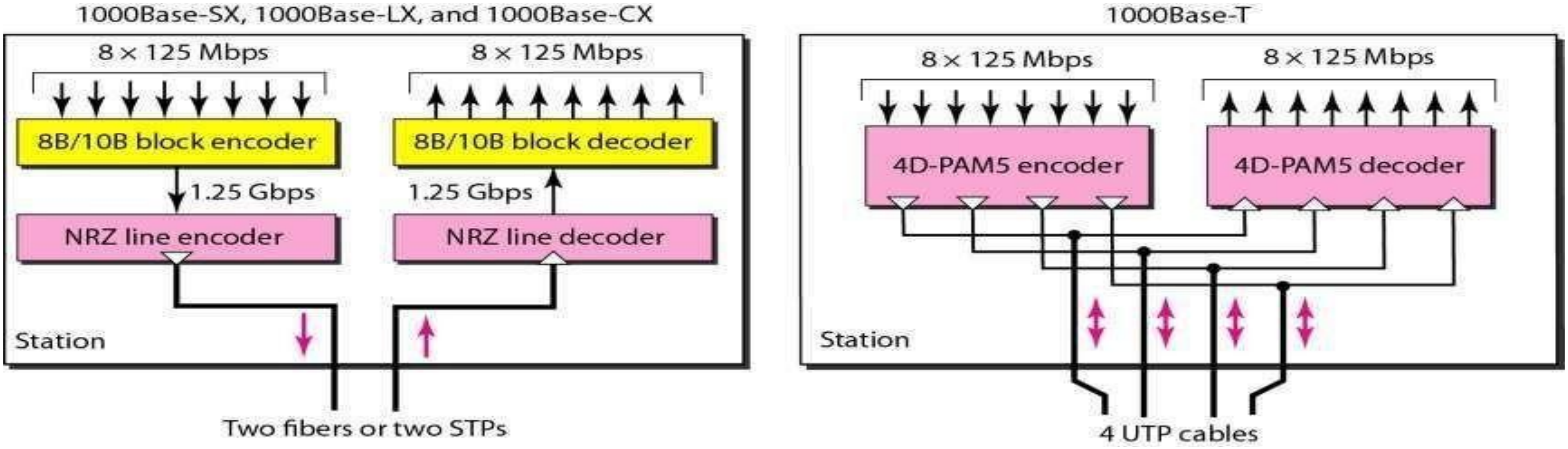
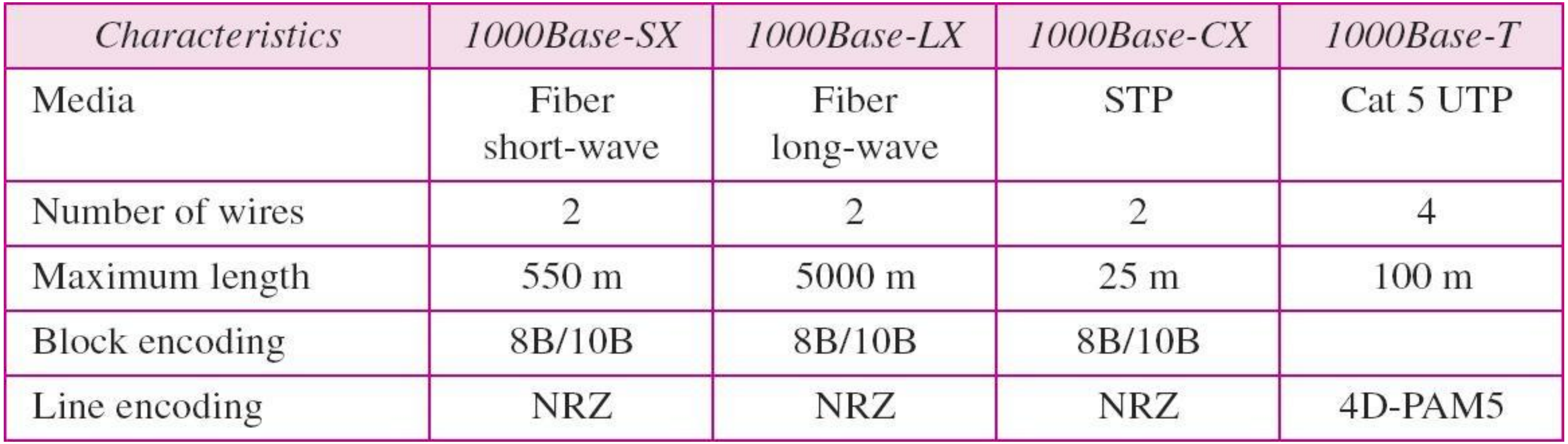


Table 13.3 *Summary of Gigabit Ethernet implementations*



Characteristics	*1000Base-SX*	*1000Base-LX*	*1000Base-CX*	*1000Base-T*
Media	Fiber short-wave	Fiber long-wave	STP	Cat 5 UTP
Number of wires	2	2	2	4
Maximum length	550 m	5000 m	25 m	100 m
Block encoding	8B/10B	8B/10B	8B/10B	
Line encoding	NRZ	NRZ	NRZ	4D-PAM5

Five categories of **Connecting devices**

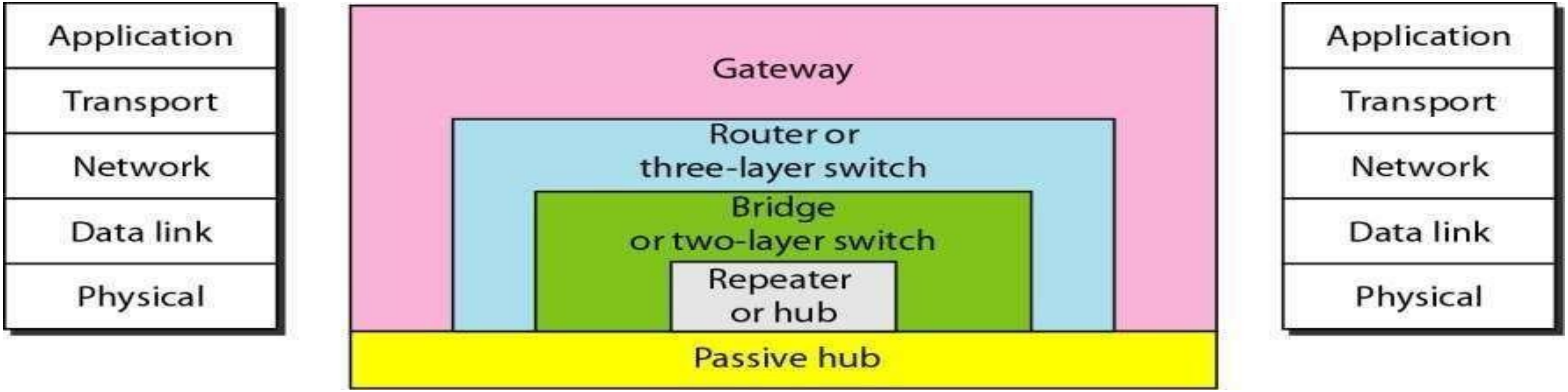
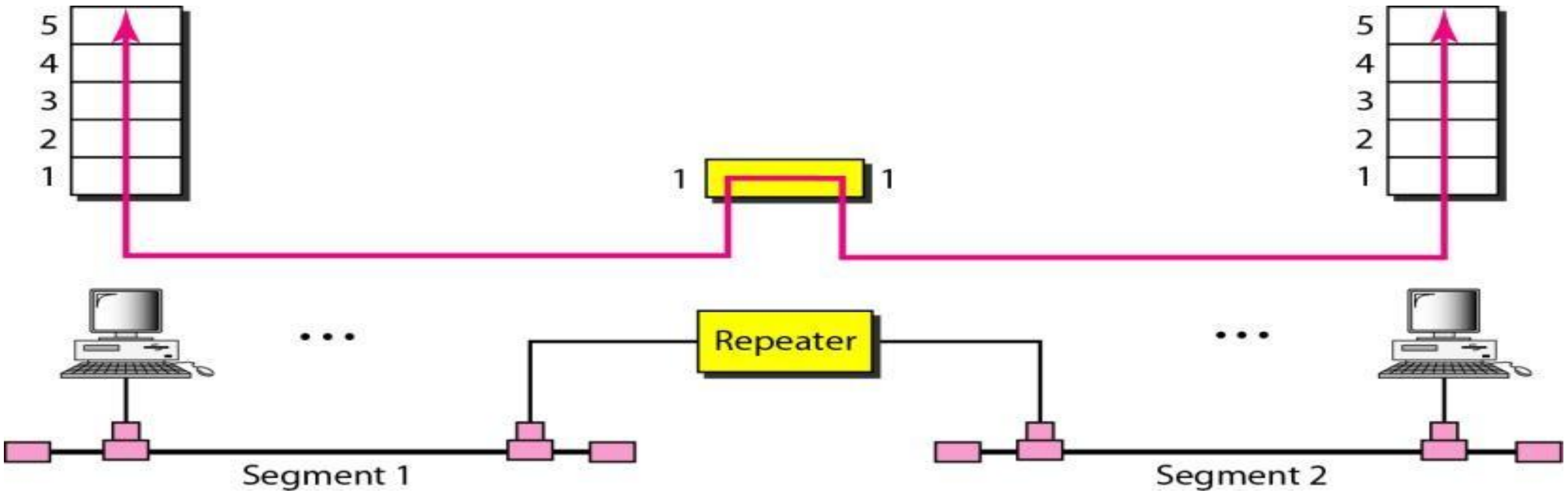


Figure 15.2 *Arepeater connecting two segments of a LAN*



Note

A repeater connects segments of a LAN.

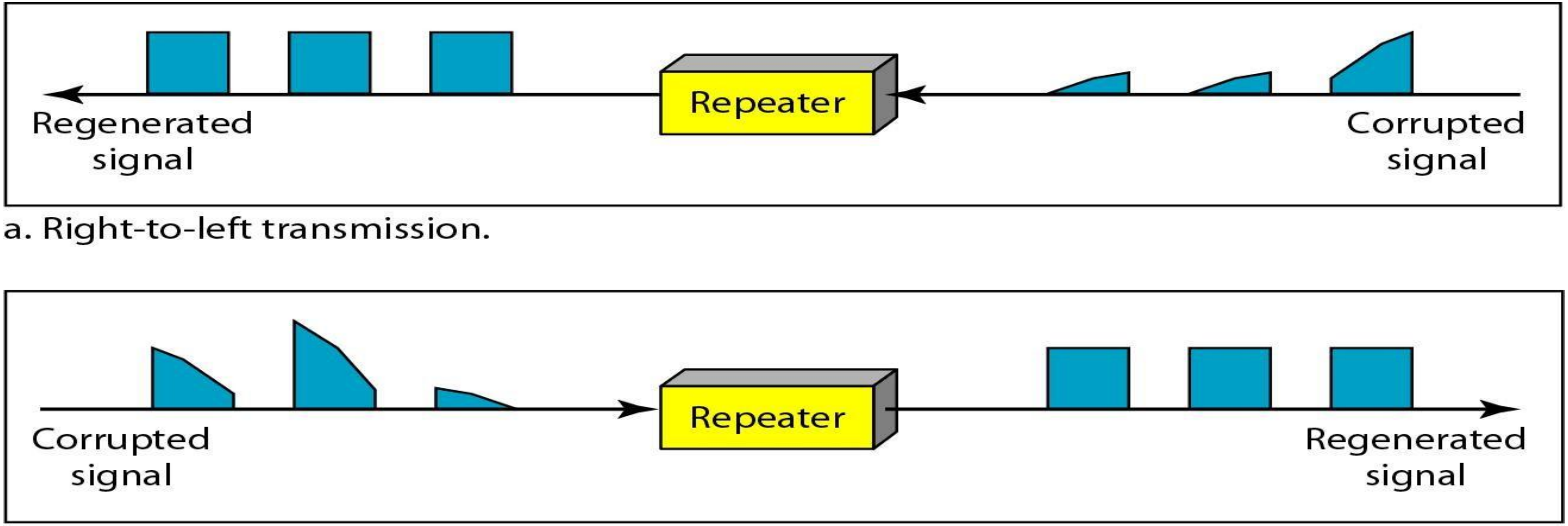
Note

A repeater forwards every frame; it has no filtering capability.

Note

A repeater is a regenerator, not an amplifier.

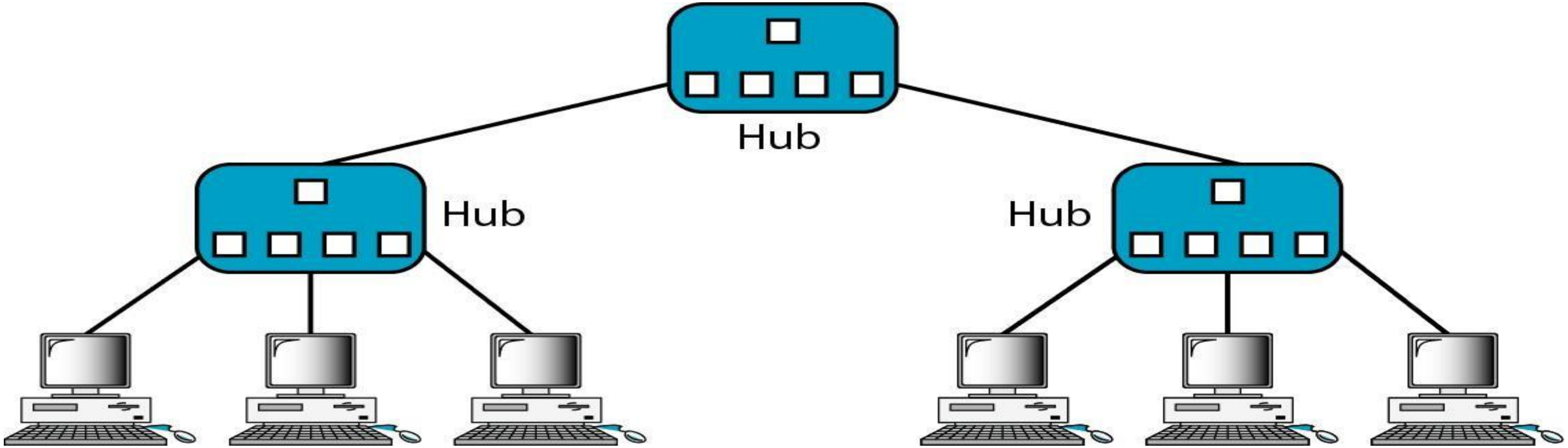
Figure 15.3 *Function of a repeater*



a. Right-to-left transmission.

b. Left-to-right transmission.

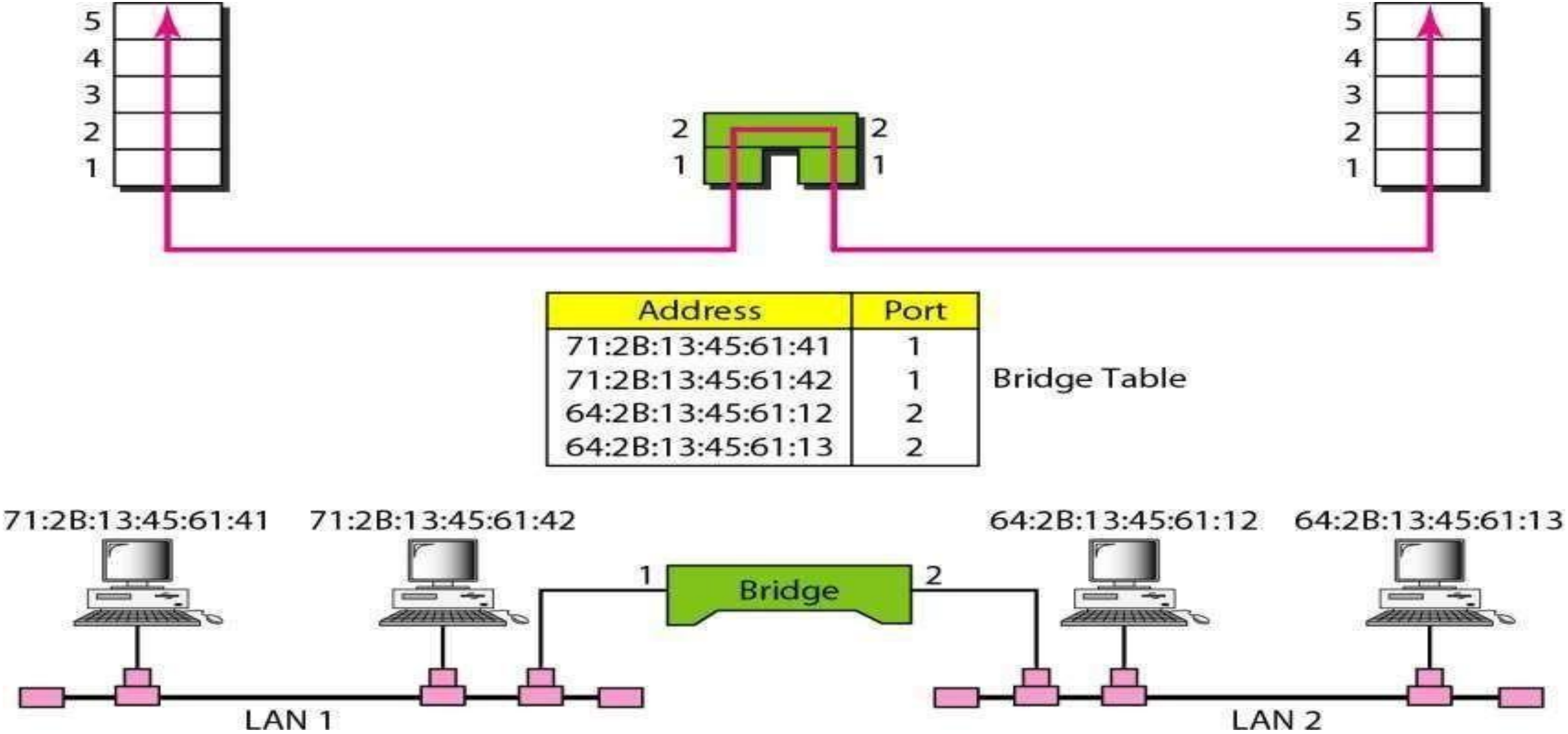
Figure 15.4 *Ahierarchy of hubs*



Note

A bridge has a table used in filtering decisions.

Figure 15.5 *Abridge connecting two LANs*

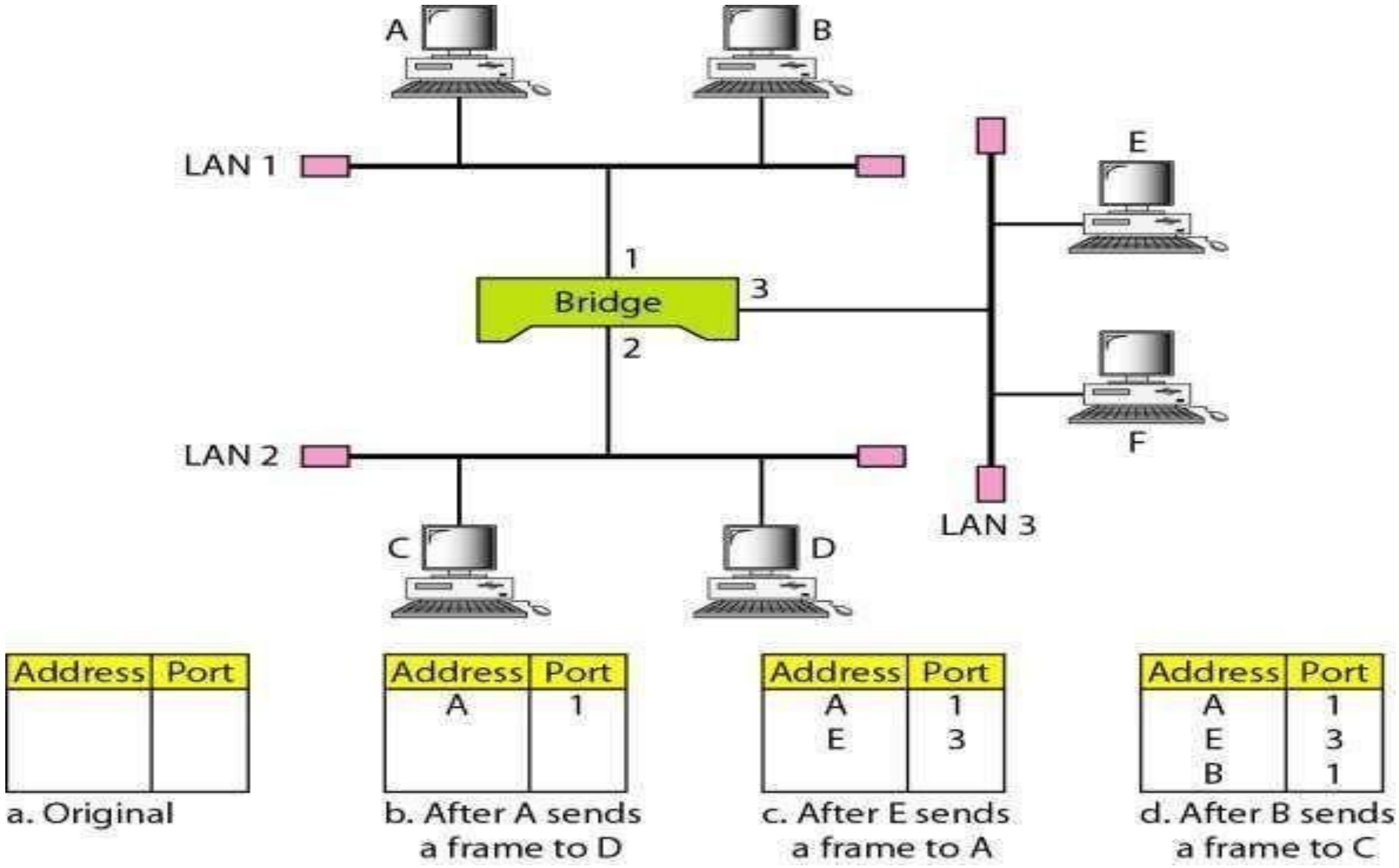


Address	Port
71:2B:13:45:61:41	1
71:2B:13:45:61:42	1
64:2B:13:45:61:12	2
64:2B:13:45:61:13	2

Note

A bridge does not change the physical (MAC) addresses in a frame.

Figure 15.6 *Alearning bridge and the process of* ***learning***



Address	Port

a. Original

Address	Port
A	1

b. After A sends a frame to D

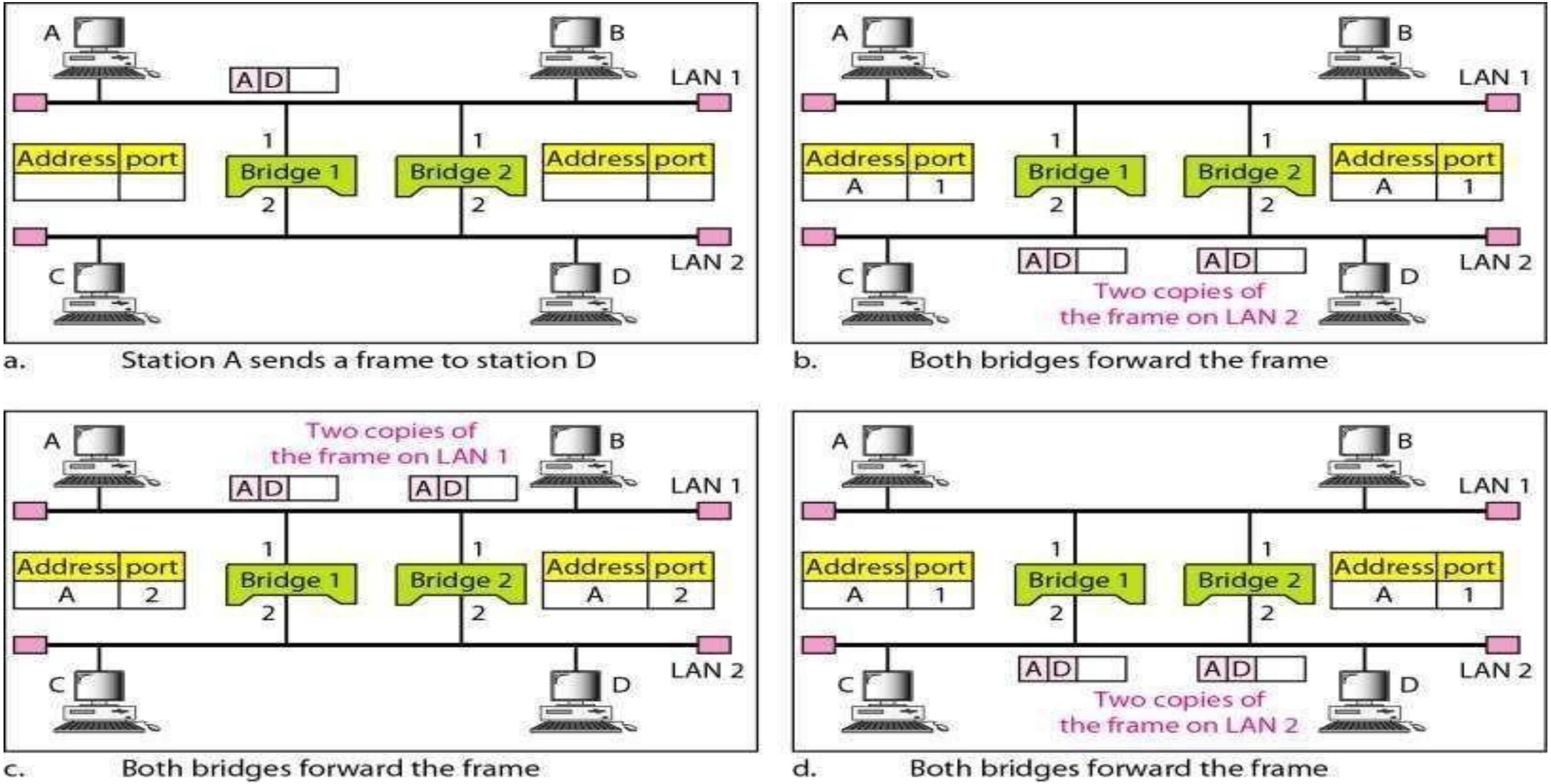
Address	Port
A	1
E	3

c. After E sends a frame to A

Address	Port
A	1
E	3
B	1

d. After B sends a frame to C

Figure 15.7 *Loop problem in a learning bridge*


A
B
C
D
LAN 1
LAN 2
A D
Bridge 1
Bridge 2
1
2
Address port
A 1
A 2
Two copies of the frame on LAN 2
Two copies of the frame on LAN 1

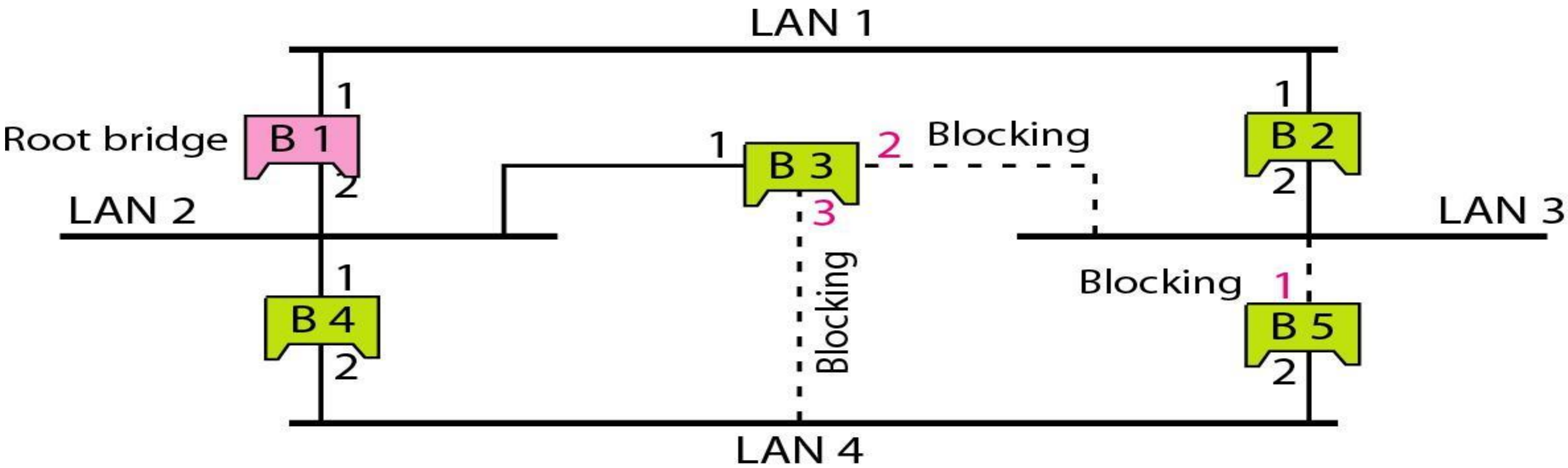
a. Station A sends a frame to station D

b. Both bridges forward the frame

c. Both bridges forward the frame

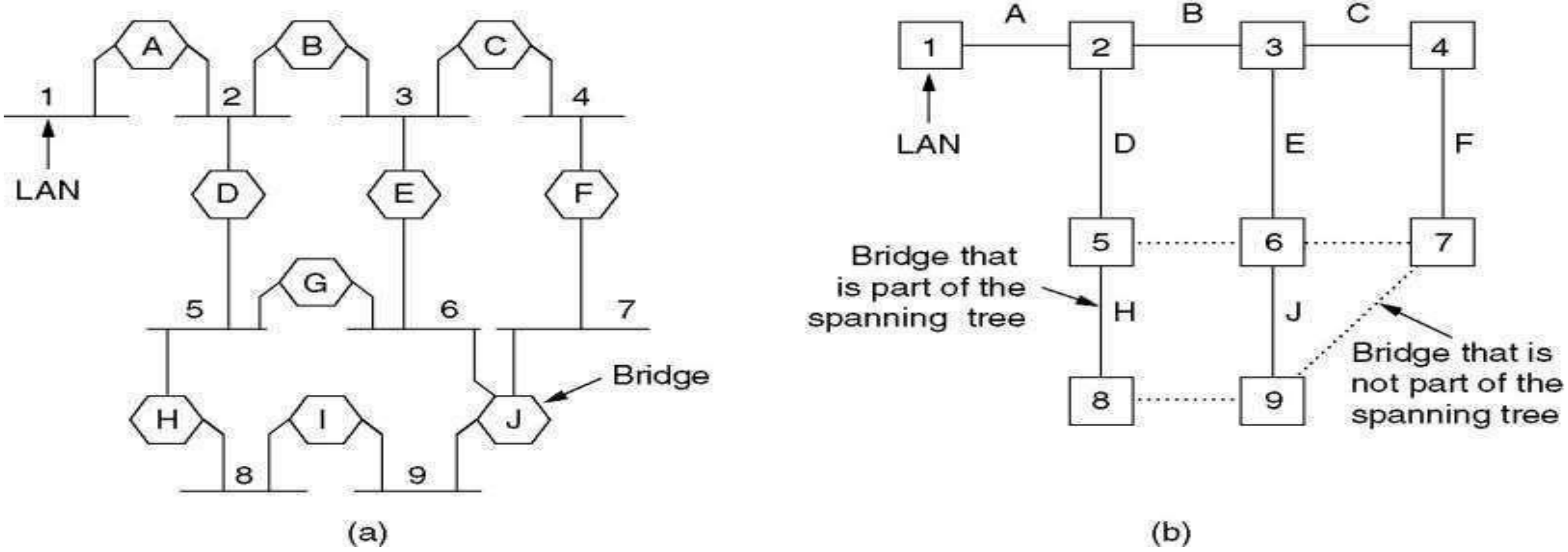
d. Both bridges forward the frame

Forwarding and blocking ports after using ***spanning tree*** *algorithm*



Ports 2 and 3 of bridge B3 are blocking ports (no frame is sent out of these ports).
Port 1 of bridge B5 is also a blocking port (no frame is sent out of this port).

Spanning Tree Bridges


A
B
C
1
2
3
4
LAN
D
E
F
G
5
6
7
Bridge
H
I
J
8
9
(a)
A
B
C
1
2
3
4
LAN
D
E
F
5
6
7
Bridge that is part of the spanning tree
H
J
Bridge that is not part of the spanning tree
8
9
(b)

(a) Interconnected LANs. (b) A spanning tree covering the LANs. The dotted lines are not part of the spanning tree.

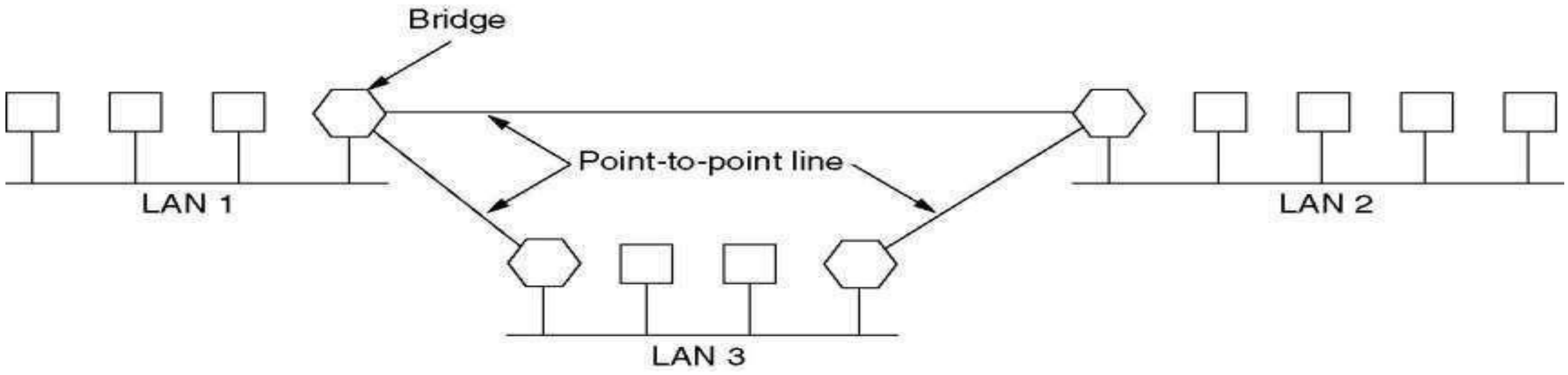
Bridge types

1. Source routing bridges

   * Routing parameters will be take care by source only

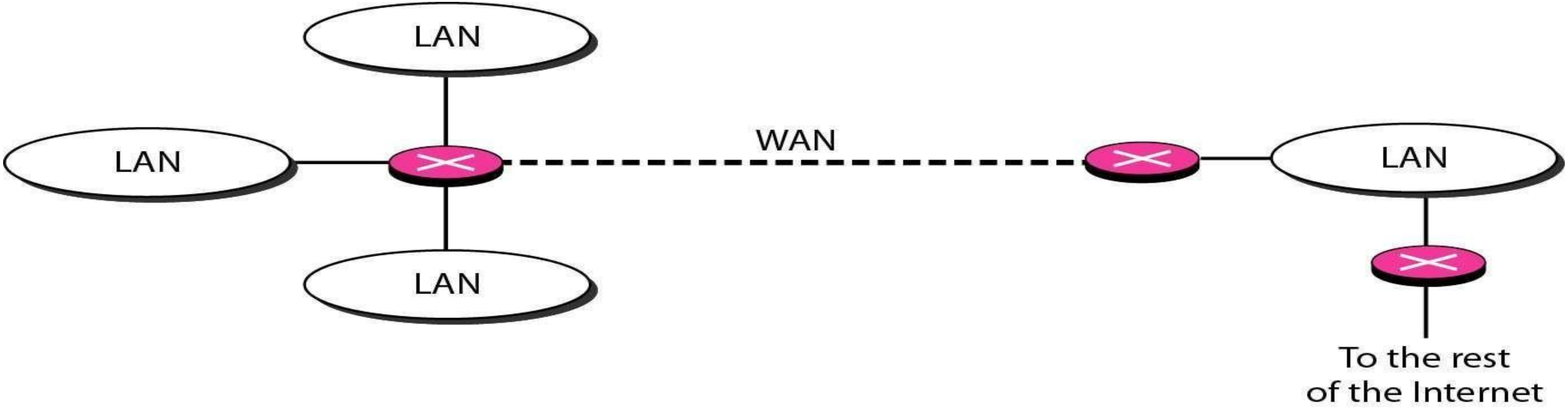
1. Transparent bridges
2. Remote bridges

Remote Bridges



Remote bridges can be used to interconnect distant LANs.

Figure 15.11 *Routers connecting independent LANs and WANs*



Gateways

- These connect two computers that use different connection- oriented transport protocols. For example, suppose a computer using the connection-oriented TCP/IP protocol needs to talk to a computer using the connection-oriented ATM transport protocol.

- The transport gateway can copy the packets from one connection to the other, reformatting them as need be.

- Finally, application gateways understand the format and contents of the data and translate messages from one format to another. An e-mail gateway could translate Internet messages into SMS messages for mobile phones, for example.

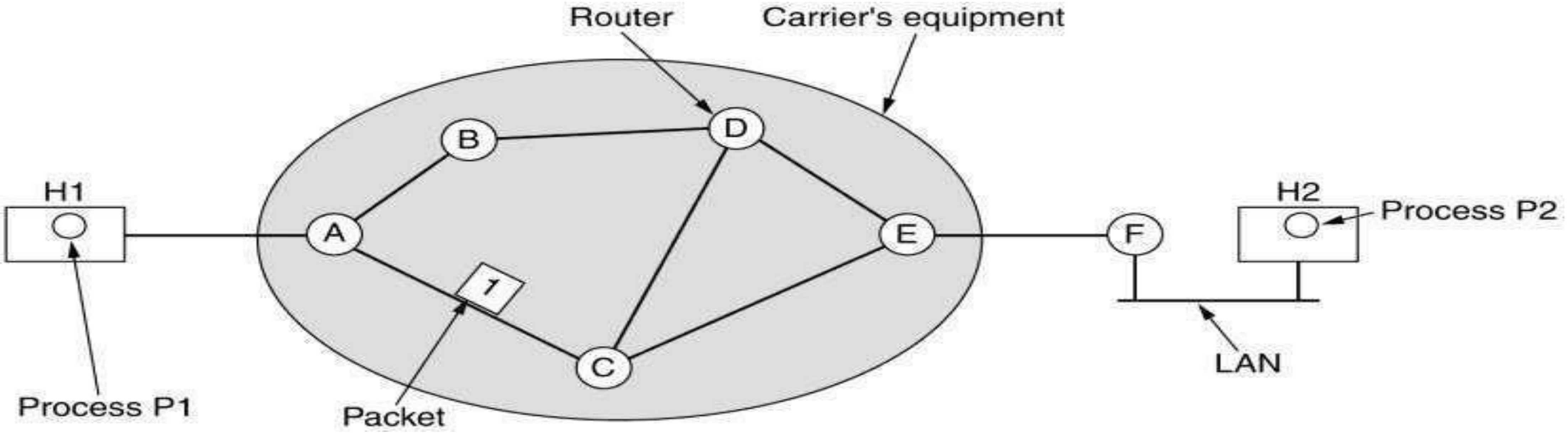
III UNIT

The Network Layer

Network Layer Design Isues

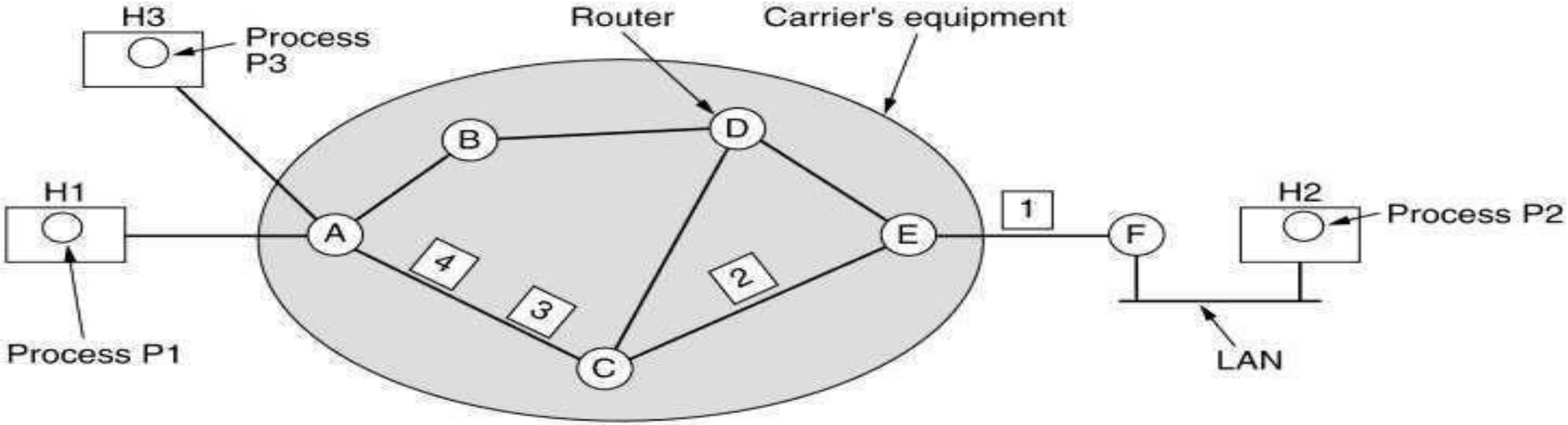
- Store-and-Forward Packet Switching
- Services Provided to the Transport Layer
- Implementation of Connectionless Service
- Implementation of Connection-Oriented Service
- Comparison of Virtual-Circuit and Datagram Subnets

Store-and-Forward Packet Switching



The environment of the network layer protocols.

Implementation of Connection-Oriented Service



A's table

In		Out	
H1	1	C	1
H3	1	C	2

C's table

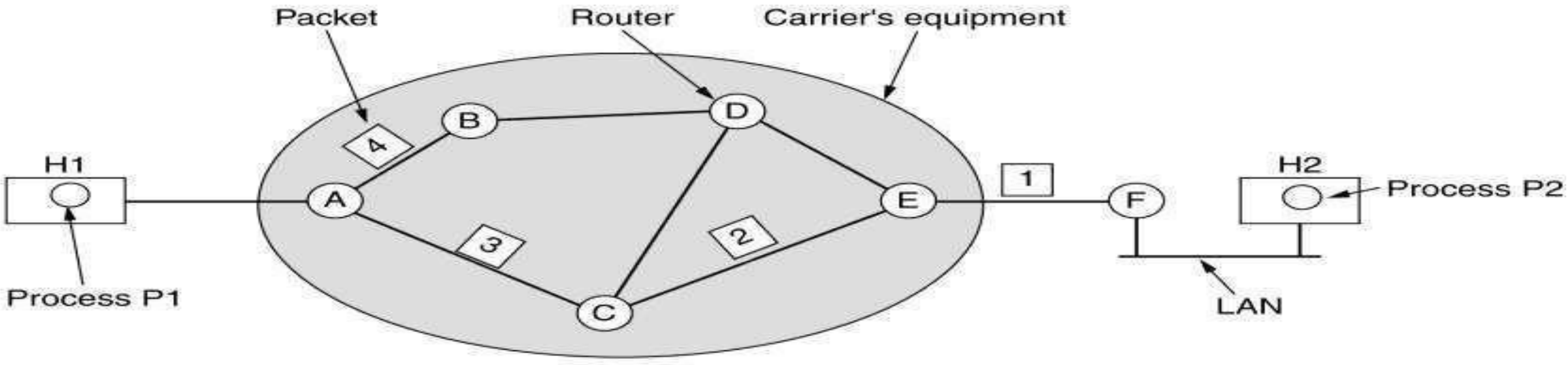
A	1	E	1
A	2	E	2

E's table

C	1	F	1
C	2	F	2

Routing within a virtual-circuit subnet.

Implementation of Connectionless Service



A's table

initially

Dest.	Line
A	–
B	B
C	C
D	B
E	C
F	C

later

Dest.	Line
A	–
B	B
C	C
D	B
E	B
F	B

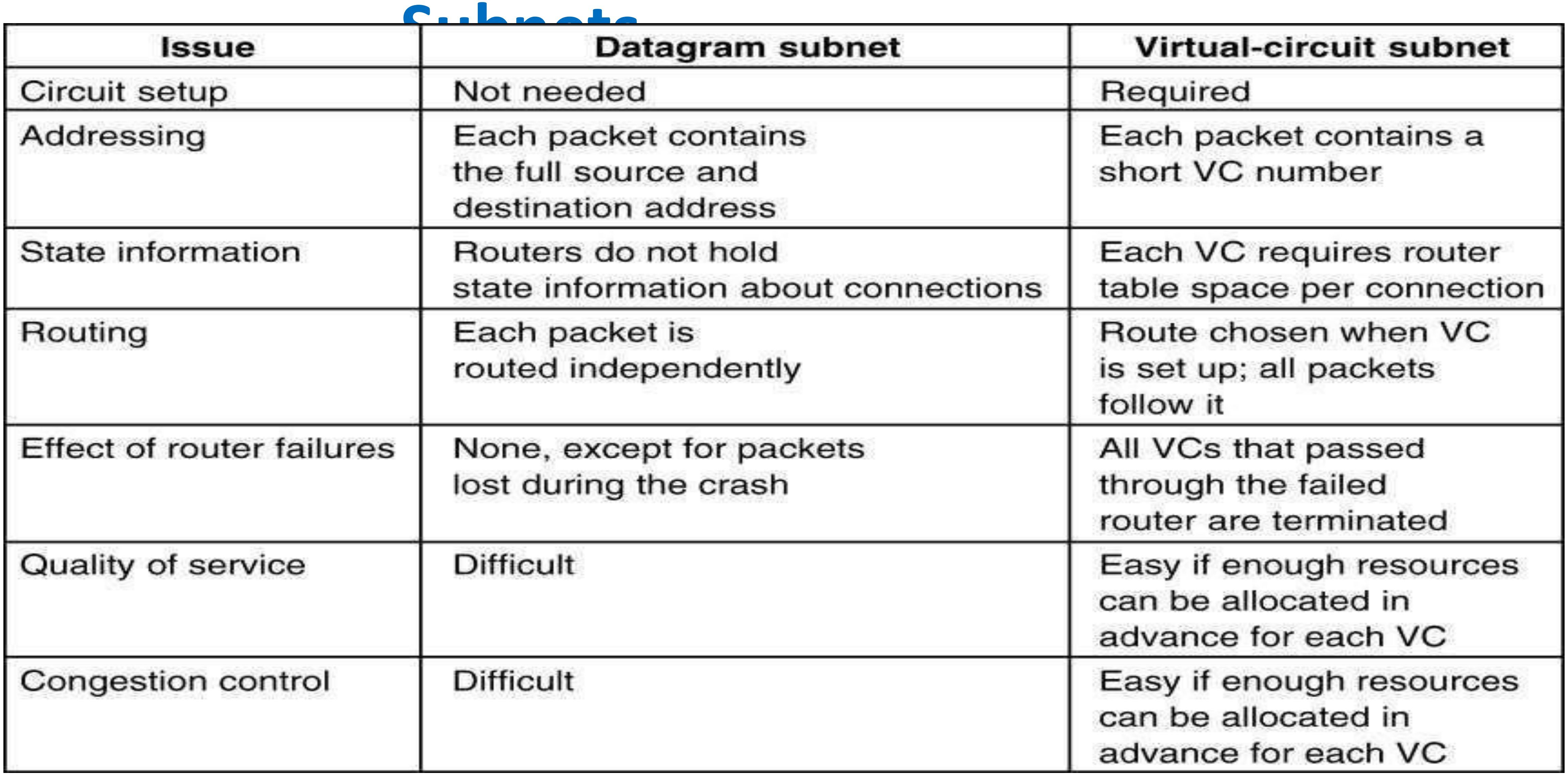
C's table

Dest.	Line
A	A
B	A
C	–
D	D
E	E
F	E

E's table

Dest.	Line
A	C
B	D
C	C
D	D
E	–
F	F

Comparison of Virtual-Circuit and Datagram Subnets

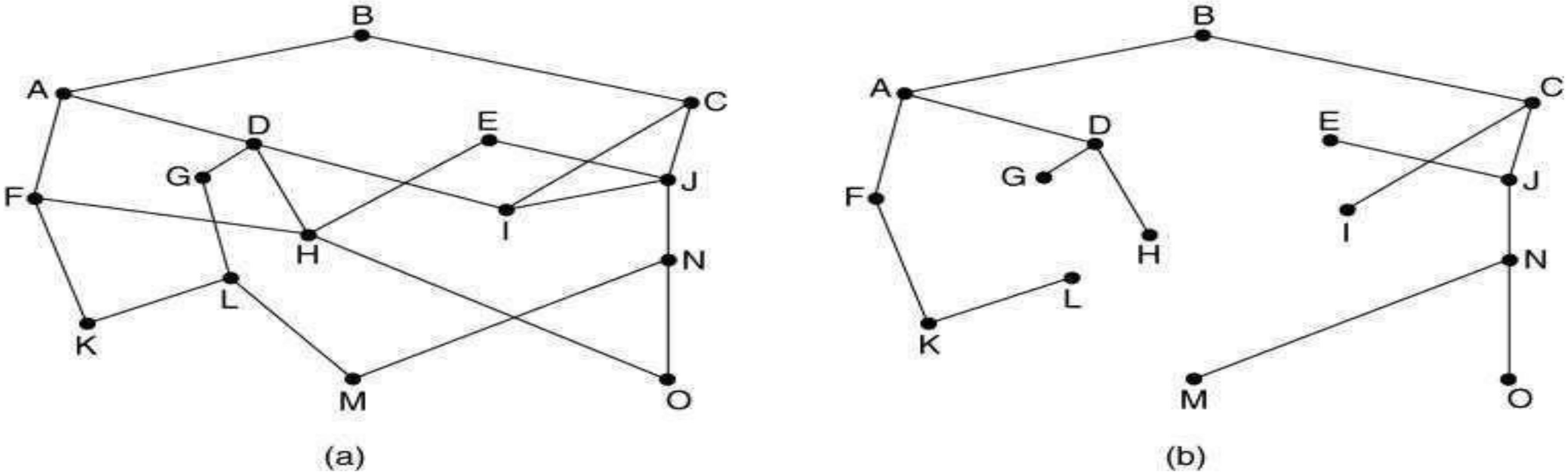


Issue	Datagram subnet	Virtual-circuit subnet
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

Routing Algorithms

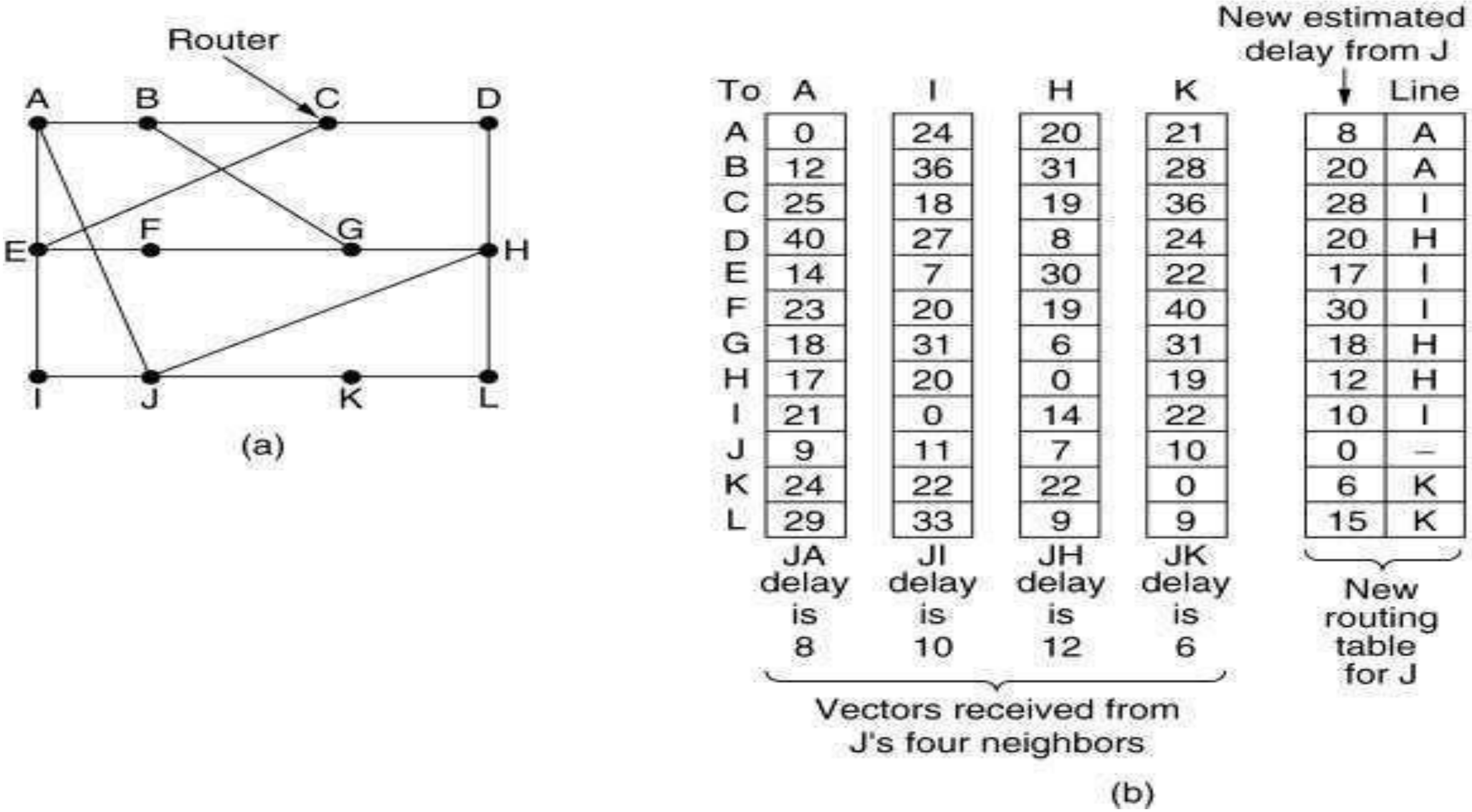
- The Optimality Principle
- Shortest Path Routing
- Flooding
- Distance Vector Routing
- Link State Routing
- Hierarchical Routing

The Optimality Principle



(a) A subnet. (b) A sink tree for router B.

Distance Vector Routing



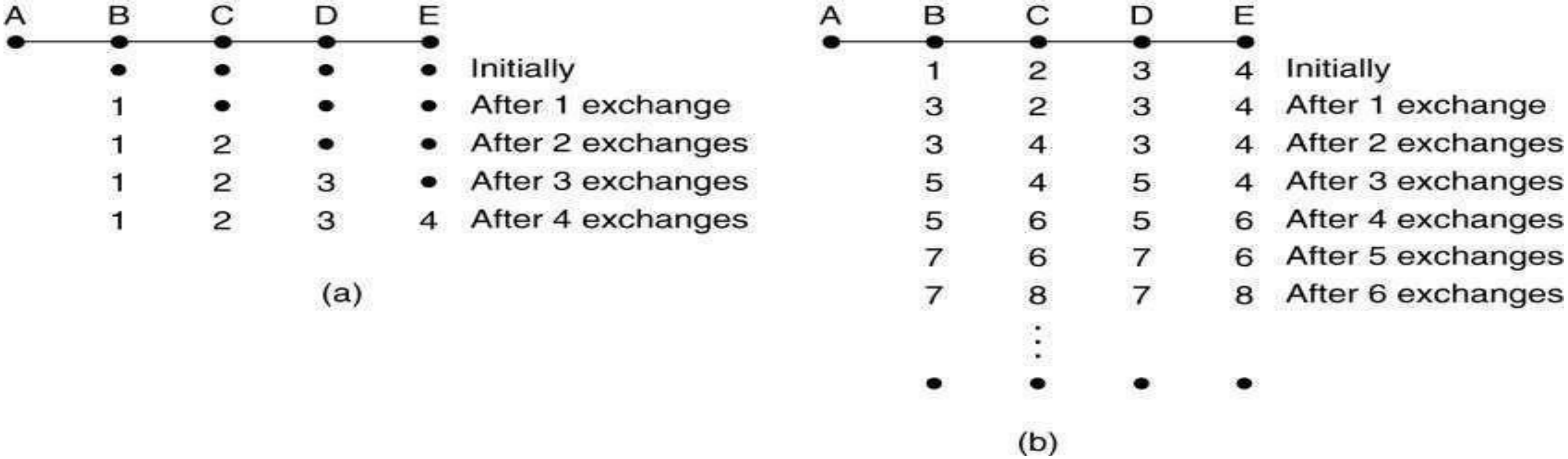
To	A	I	H	K	New estimated delay from J	Line
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	–
K	24	22	22	0	6	K
L	29	33	9	9	15	K
	JA delay is 8	JI delay is 10	JH delay is 12	JK delay is 6	New routing table for J	

Vectors received from J's four neighbors

(b)

(a) A subnet. (b) Input from A, I, H, K, and the new routing table for J.

Distance Vector Routing (2)



A	B	C	D	E	
	●	●	●	●	Initially
	1	●	●	●	After 1 exchange
	1	2	●	●	After 2 exchanges
	1	2	3	●	After 3 exchanges
	1	2	3	4	After 4 exchanges

(a)

A	B	C	D	E	
	1	2	3	4	Initially
	3	2	3	4	After 1 exchange
	3	4	3	4	After 2 exchanges
	5	4	5	4	After 3 exchanges
	5	6	5	6	After 4 exchanges
	7	6	7	6	After 5 exchanges
	7	8	7	8	After 6 exchanges
		⋮			
	●	●	●	●	

(b)

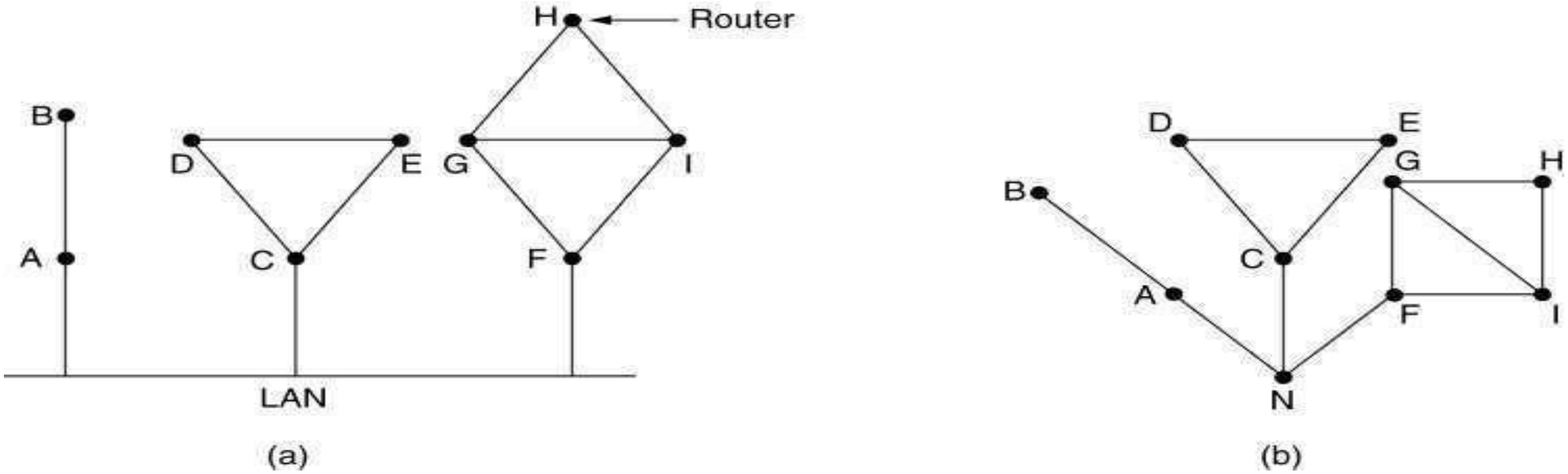
The count-to-infinity problem.

Link State Routing

Each router must do the following:

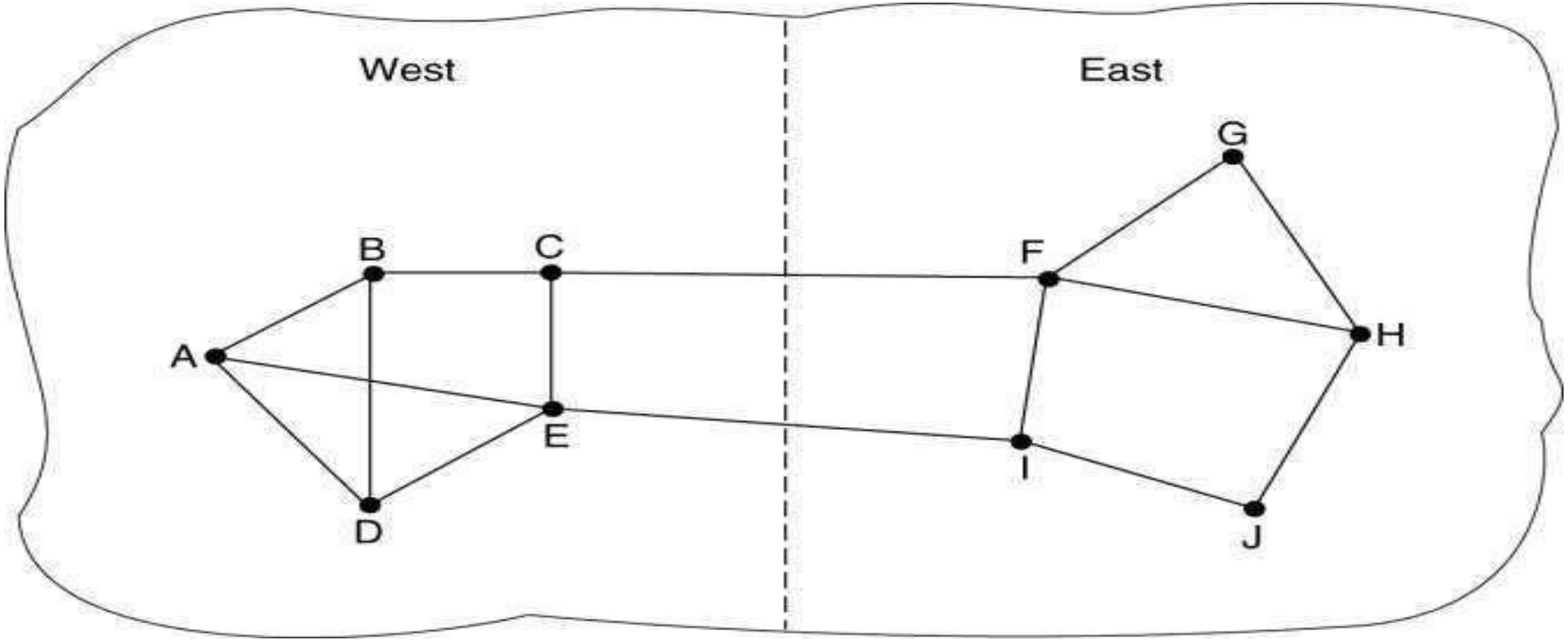
1. Discover its neighbors, learn their network address.
2. Measure the delay or cost to each of its neighbors.
3. Construct a packet telling all it has just learned.
4. Send this packet to all other routers.
5. Compute the shortest path to every other router.

Learning about the Neighbors



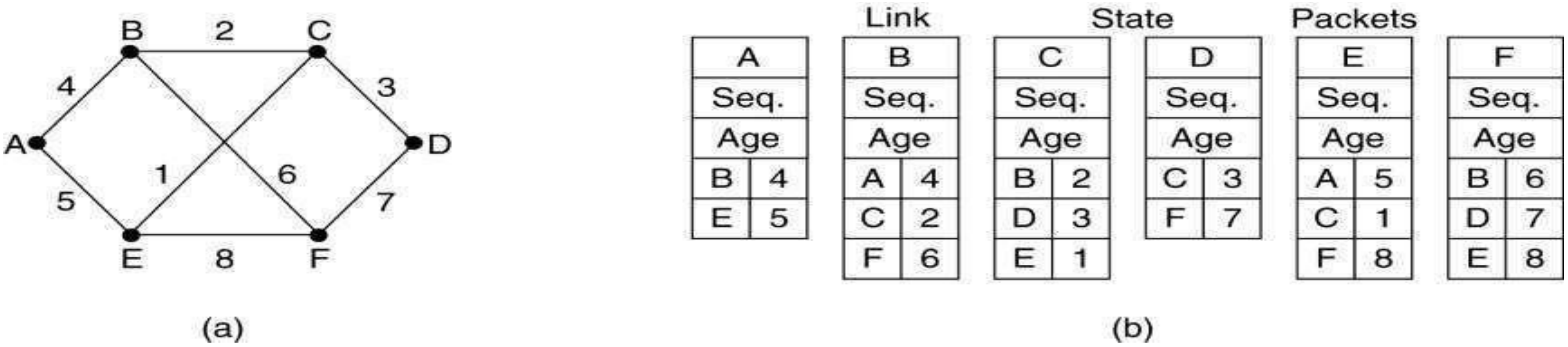
(a) Nine routers and a LAN. (b) A graph model of (a).

Measuring Line Cost



A subnet in which the East and West parts are connected by two lines.

Building Link State Packets



Link State Packets

A	
Seq.	
Age	
B	4
E	5

B	
Seq.	
Age	
A	4
C	2
F	6

C	
Seq.	
Age	
B	2
D	3
E	1

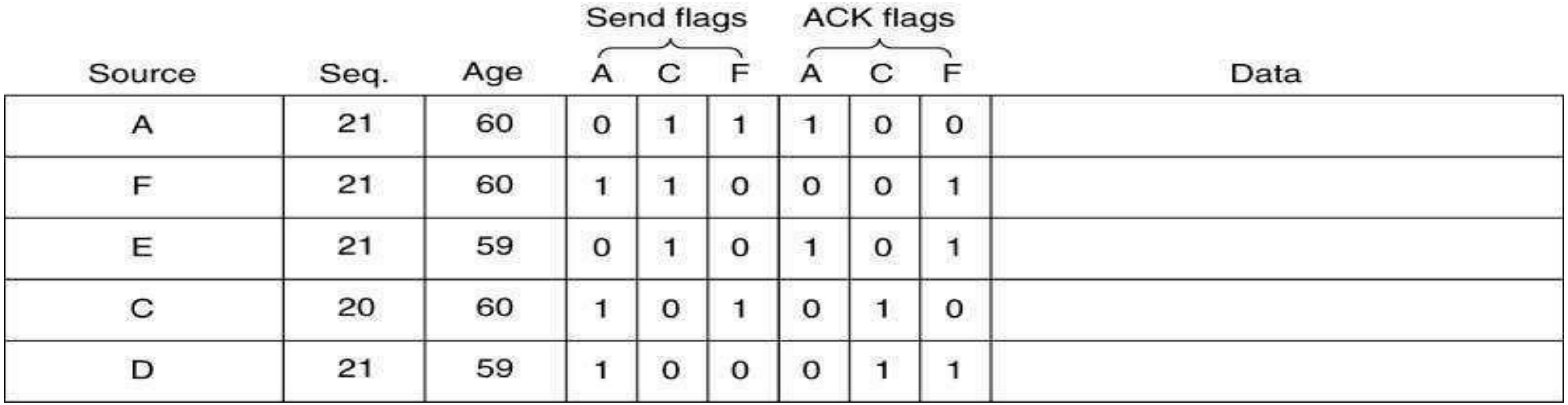
D	
Seq.	
Age	
C	3
F	7

E	
Seq.	
Age	
A	5
C	1
F	8

F	
Seq.	
Age	
B	6
D	7
E	8

(a) A subnet. (b) The link state packets for this subnet.

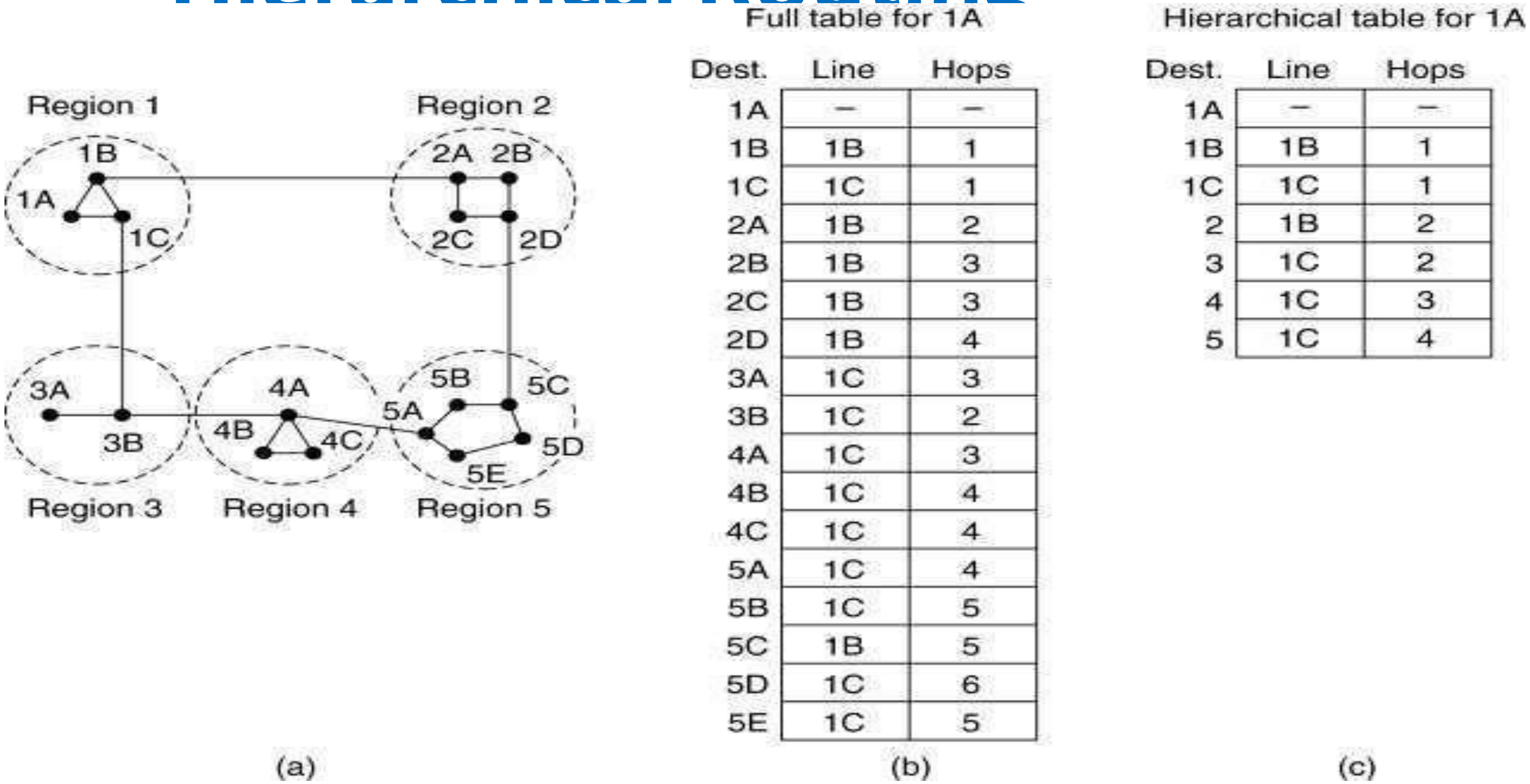
Distributing the Link State Packets



			Send flags			ACK flags			
Source	Seq.	Age	A	C	F	A	C	F	Data
A	21	60	0	1	1	1	0	0	
F	21	60	1	1	0	0	0	1	
E	21	59	0	1	0	1	0	1	
C	20	60	1	0	1	0	1	0	
D	21	59	1	0	0	0	1	1	

The packet buffer for router B in the previous slide (Fig. 5-13).

Hierarchical Routing



Full table for 1A

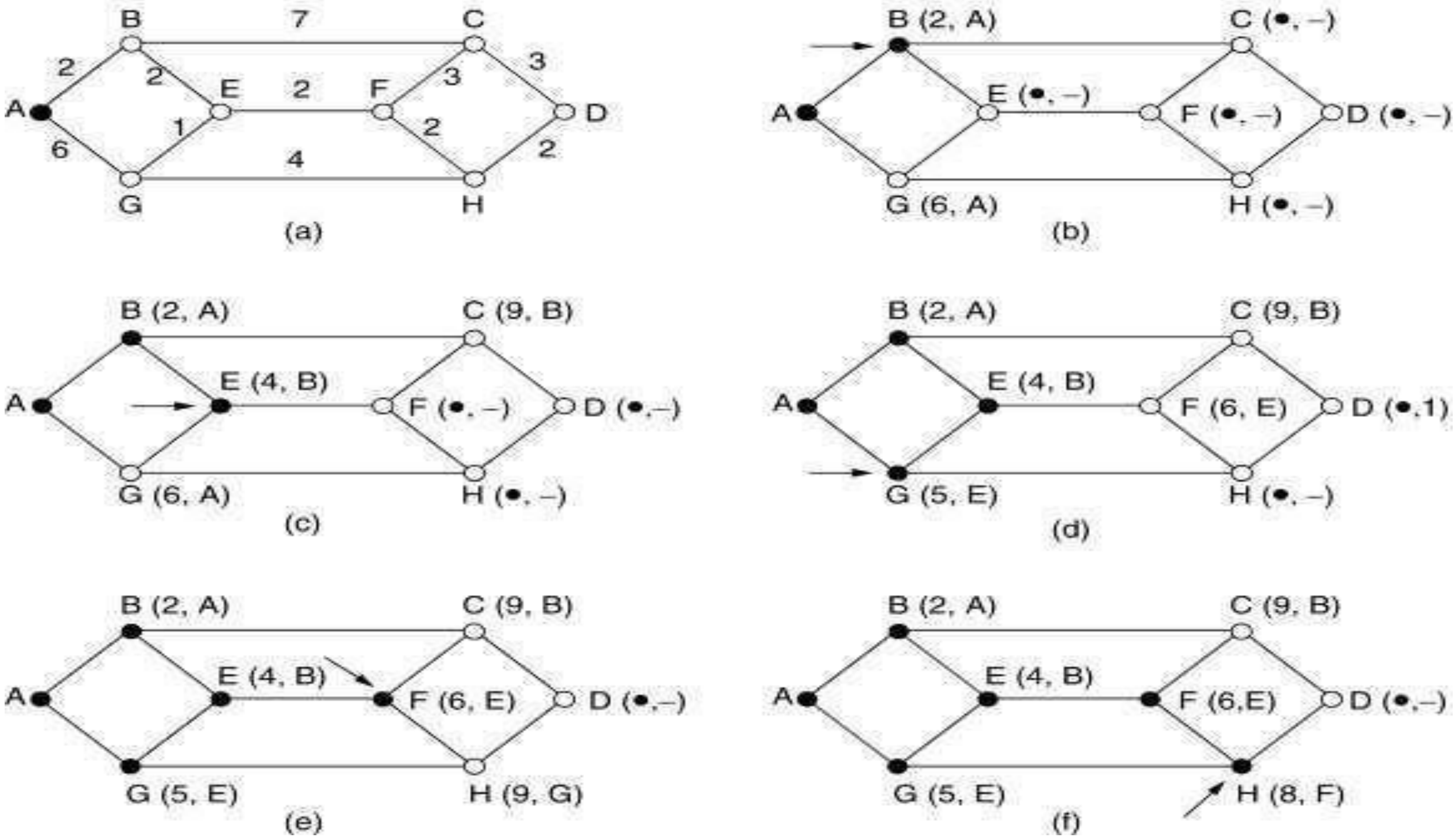
Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

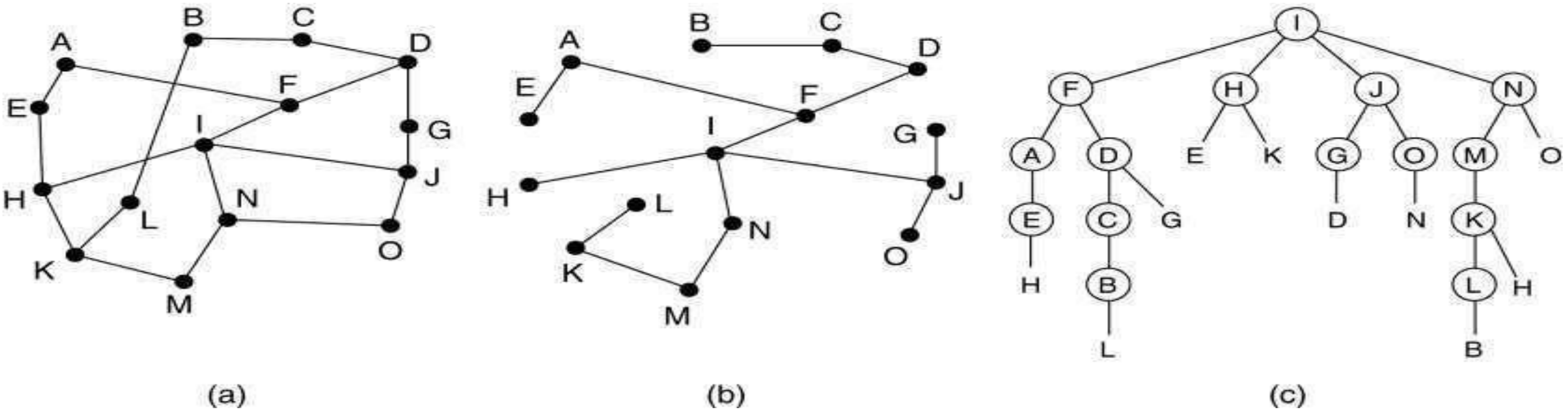
Hierarchical routing.

Shortest Path Routing



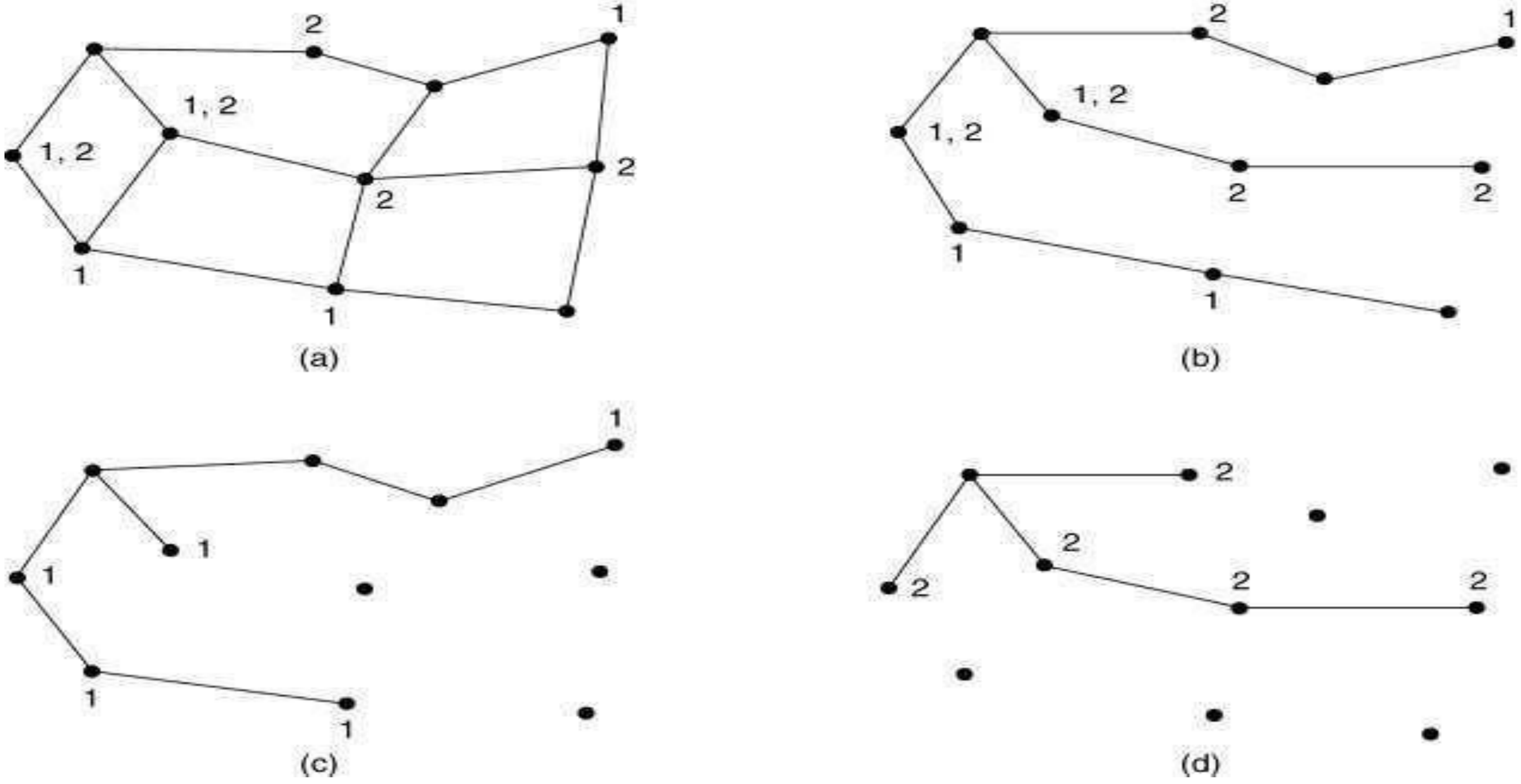
The first 5 steps used in computing the shortest path from A to D. The arrows indicate the working node.

Broadcast Routing



Reverse path forwarding. (a) A subnet. (b) a Sink tree. (c) The tree built by reverse path forwarding.

Multicast Routing

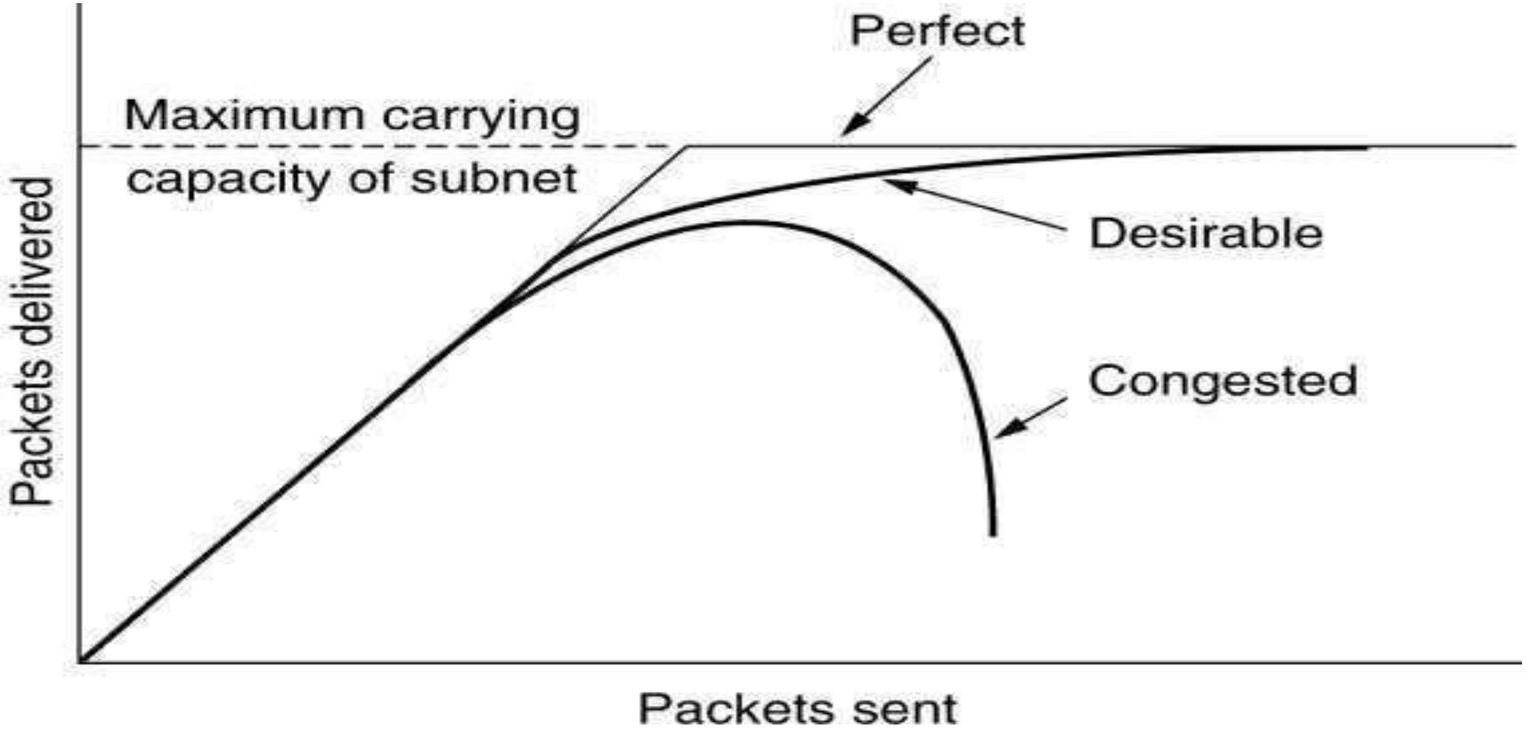


(a) A network. (b) A spanning tree for the leftmost router.
(c) A multicast tree for group 1. (d) Amulticast tree for group 2.

Congestion Control Algorithms

- General Principles of Congestion Control
- Congestion Prevention Policies
- Congestion Control in Virtual-Circuit Subnets
- Jitter Control

Congestion

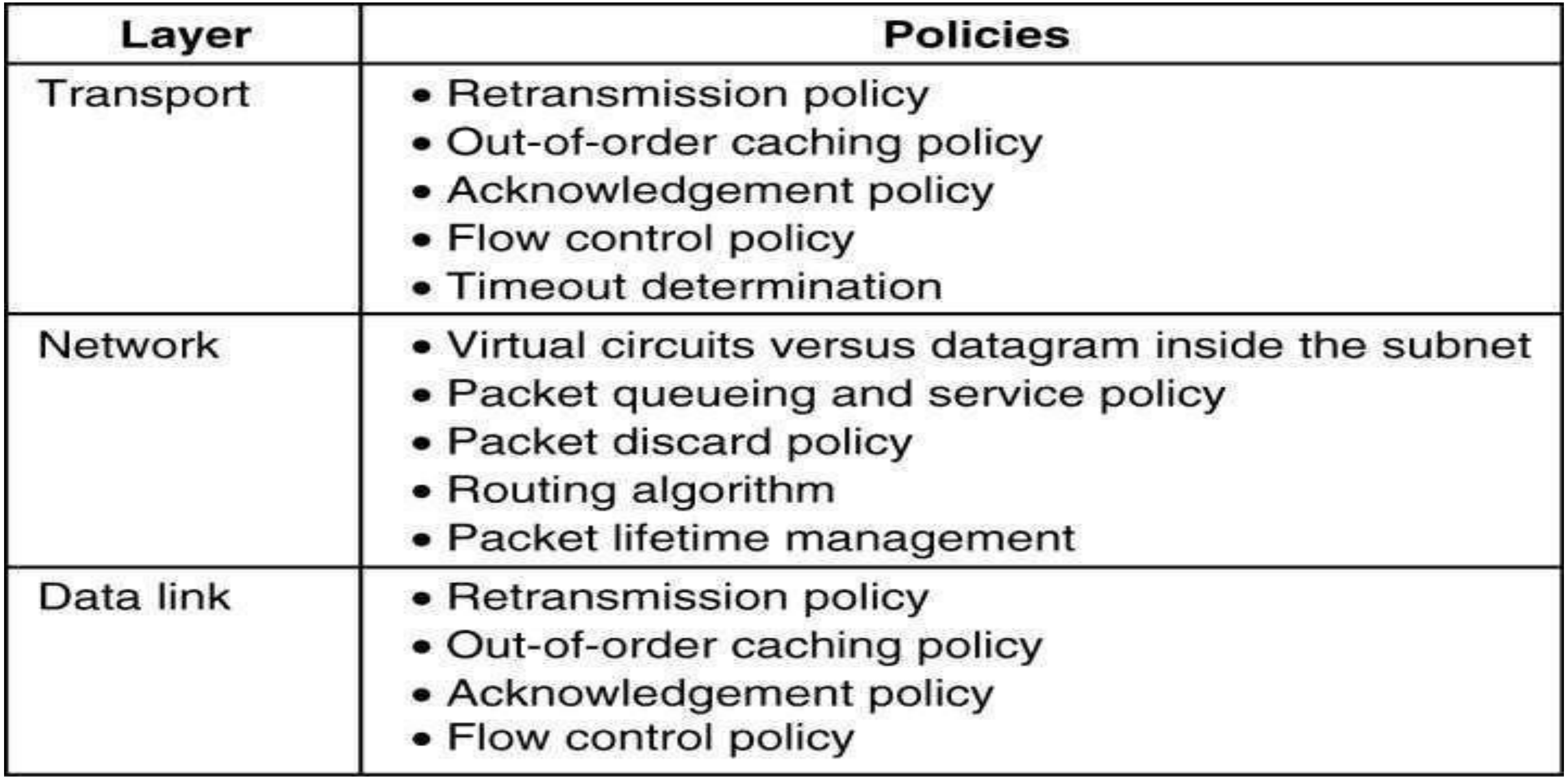


When too much traffic is offered, congestion sets in and performance degrades sharply.

General Principles of Congestion Control

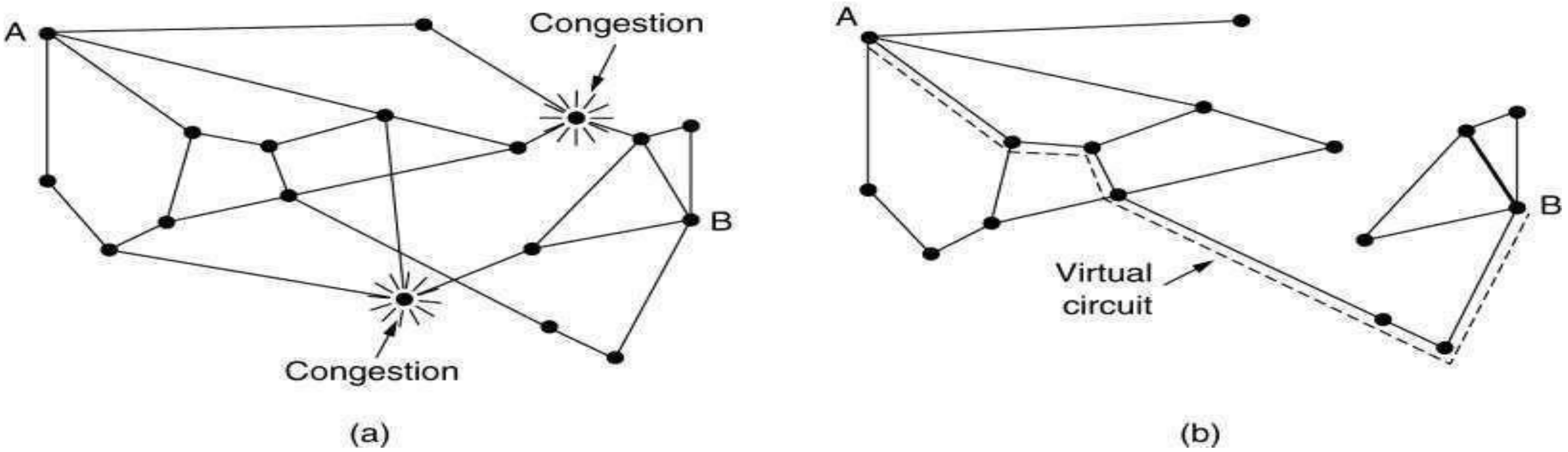
1. Monitor the system .
   - detect when and where congestion occurs.
2. Pass information to where action can be taken.
3. Adjust system operation to correct the problem.

Congestion Prevention Policies



Layer	Policies
Transport	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy • Timeout determination
Network	• Virtual circuits versus datagram inside the subnet • Packet queueing and service policy • Packet discard policy • Routing algorithm • Packet lifetime management
Data link	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy

Congestion Control in Virtual-Circuit Subnets

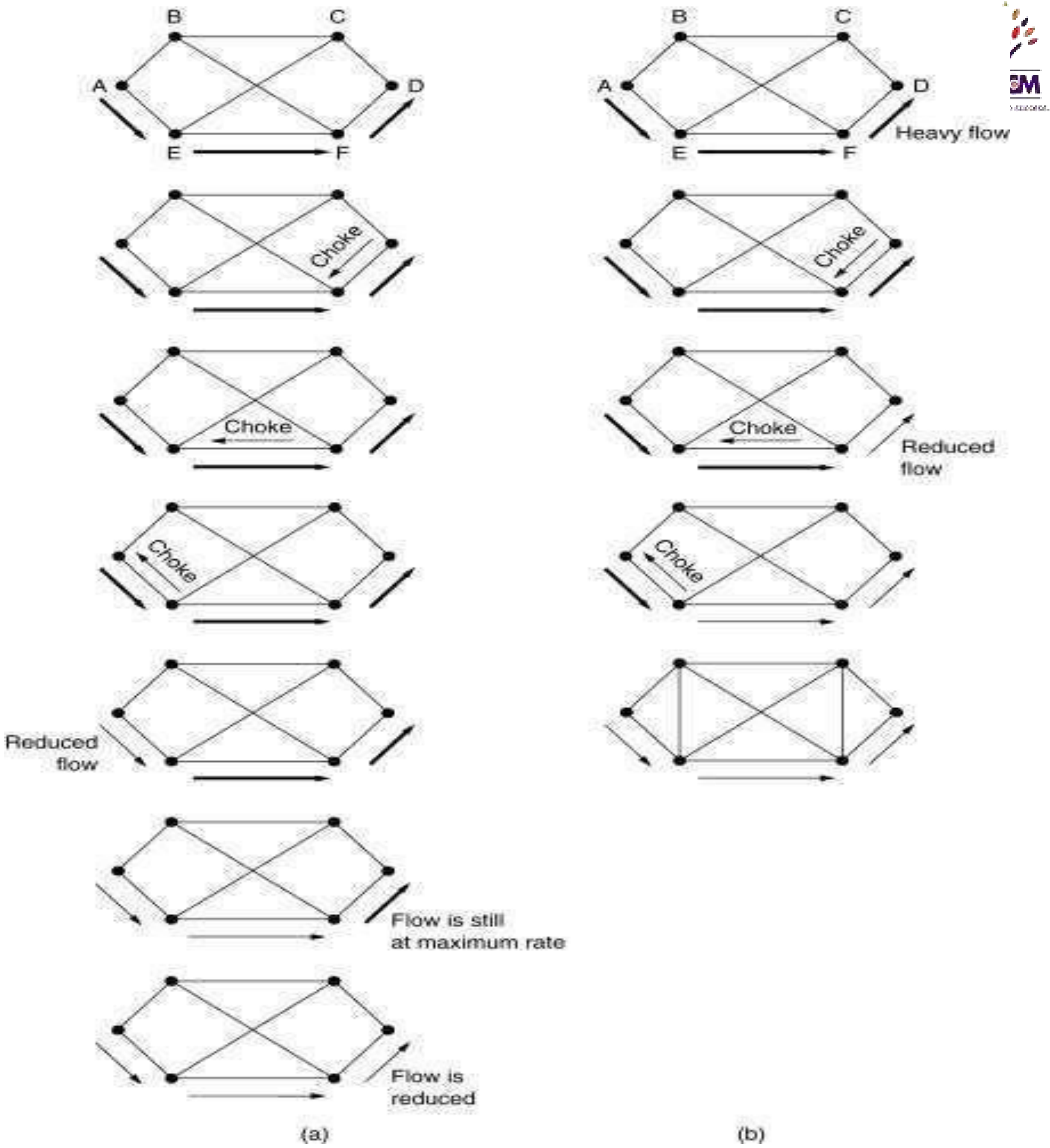


(a) A congested subnet. (b) A redrawn subnet, eliminates congestion and a virtual circuit from A to B.

Congestion Control in Datagram Subnets

- **The warning bit**
- **Choke packets**
- **Load shedding**
- **Random Early Detection**

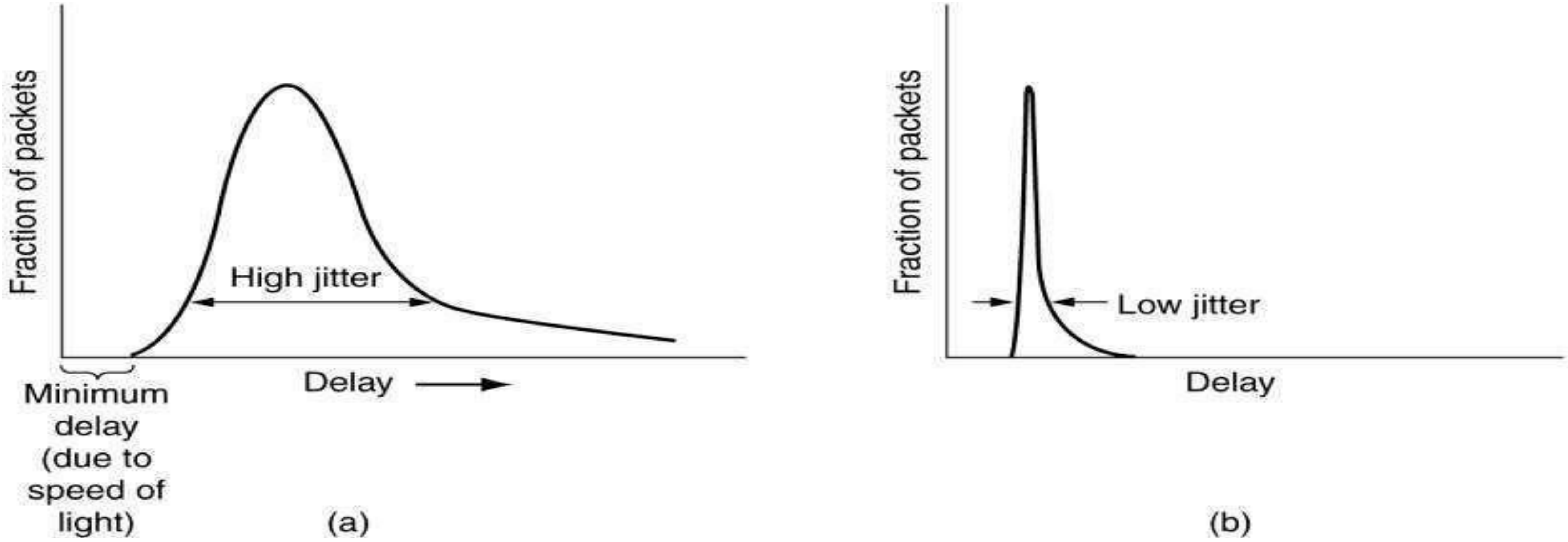
Hop-by-Hop Choke Packets



(a) A choke packet that affects only the source.

(b) A choke packet that affects each hop it passes through.

Jitter Control

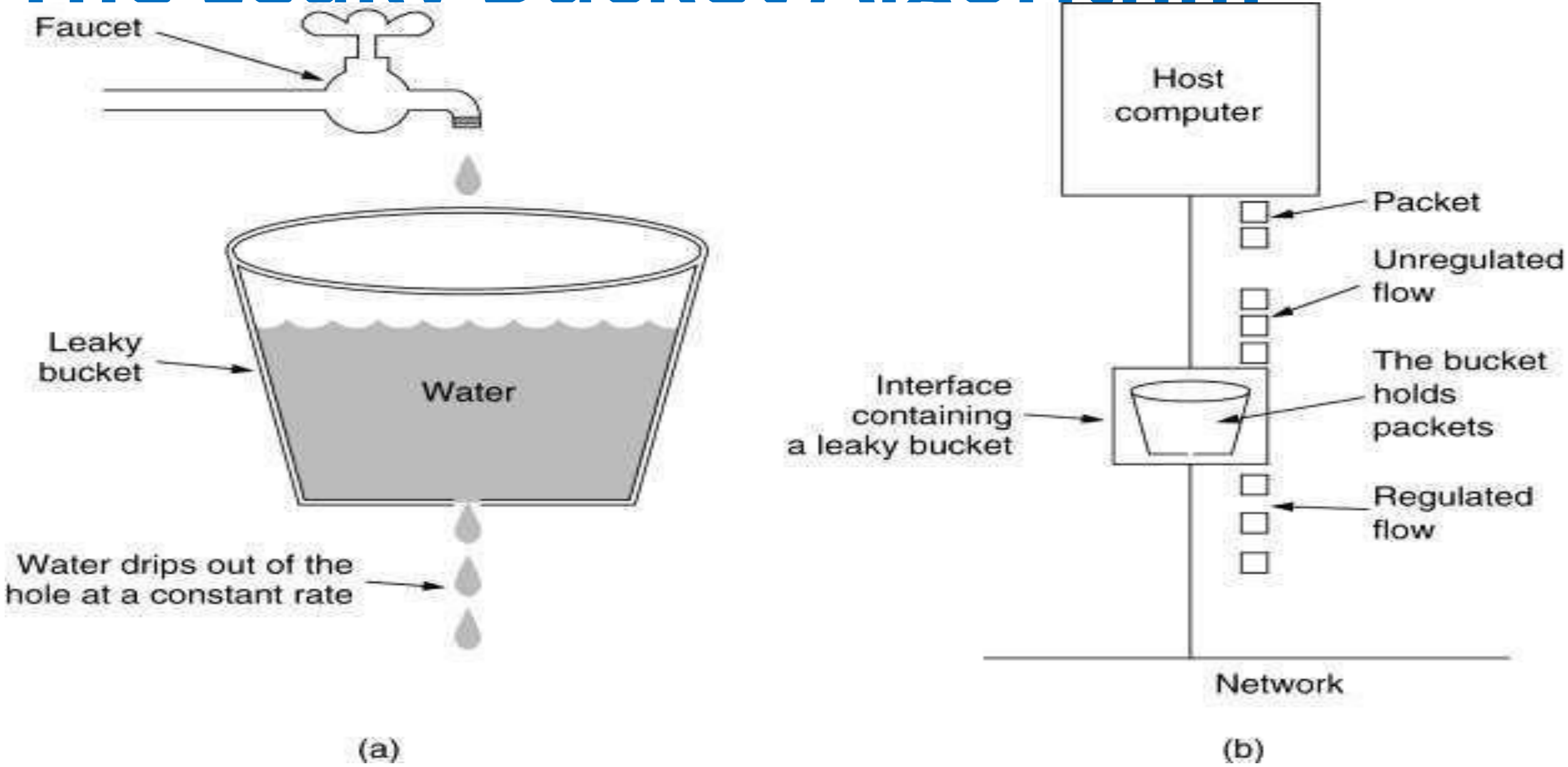


(a) High jitter. (b) Low jitter.

Congestion control Algorithms

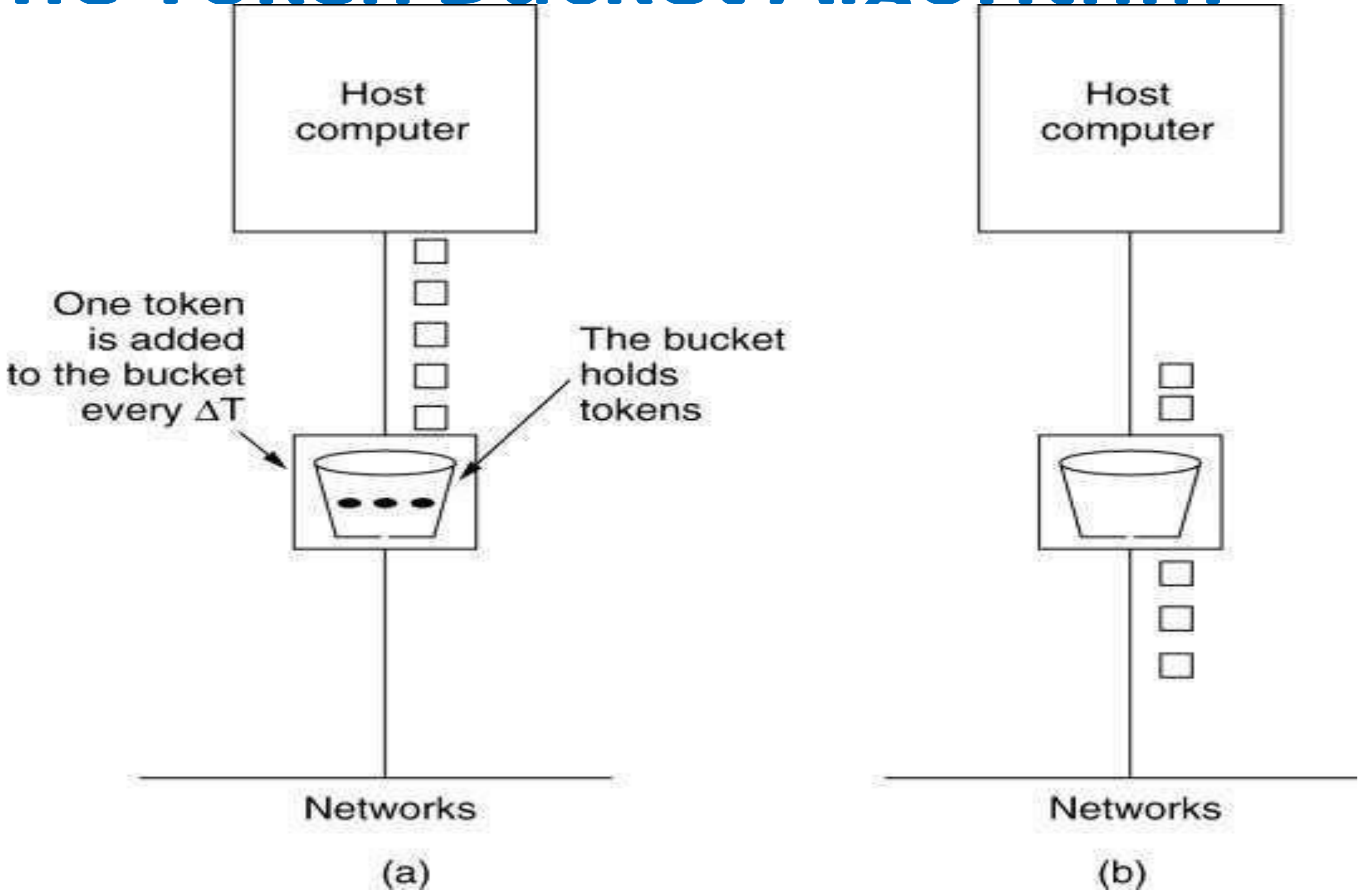
- The Leaky Bucket Algorithm
- Token Bucket Algorithm

The Leaky Bucket Algorithm



(a) A leaky bucket with water. (b) a leaky bucket with packets.

The Token Bucket Algorithm



(a) Before. (b) After.

IV-UNIT

Chapter 19

Network Layer: Logical Addressing

IPv4 ADDRESSES

IPv4 ADDRESSES

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

Topics discussed in this section:

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

Note

An IPv4 address is 32 bits long.

Note

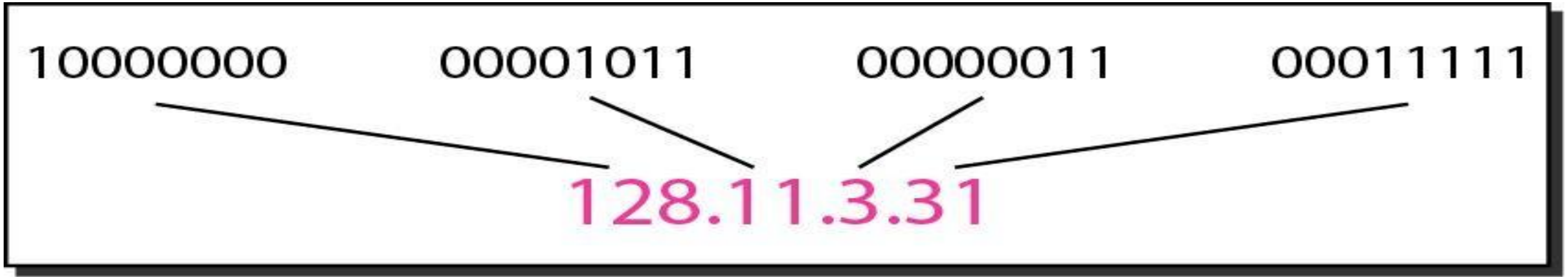
The IPv4 addresses are unique and universal.

Note

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

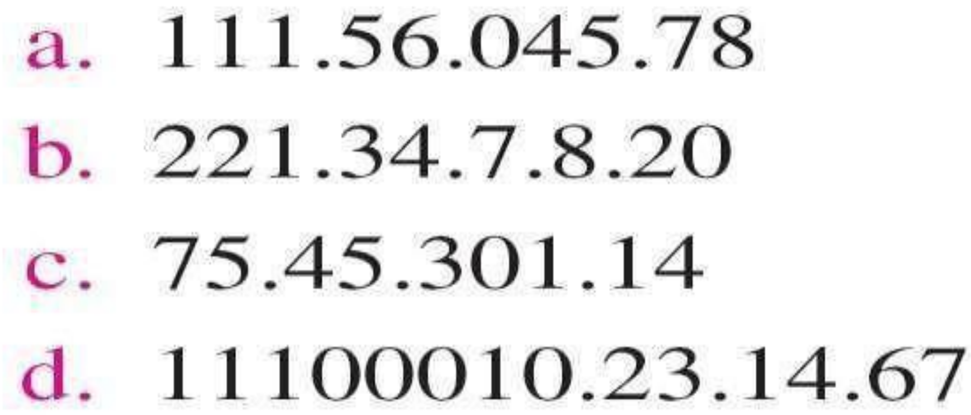
Address space: is the total no of addresses used by the protocol

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



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.

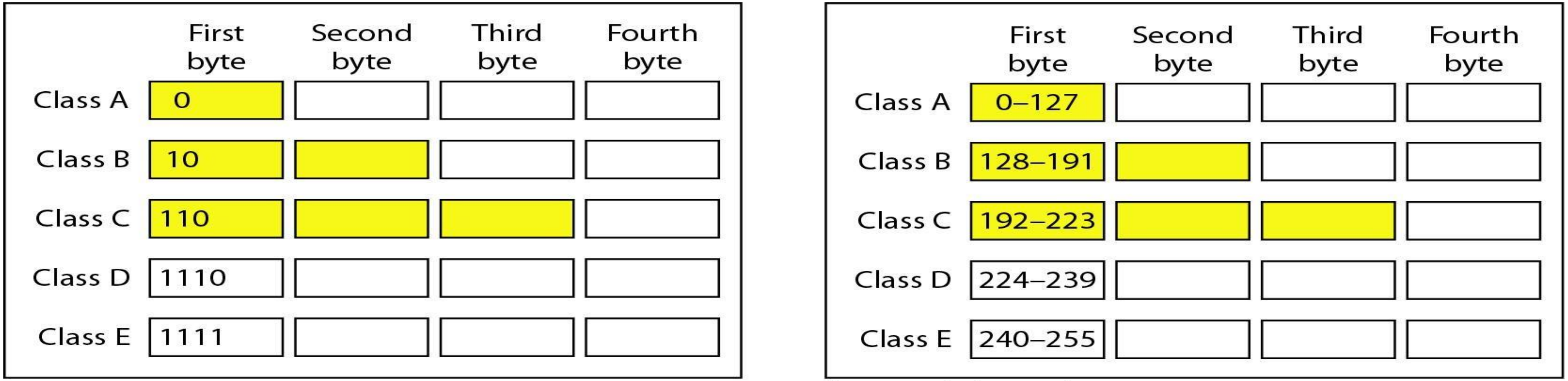
c. Each number needs to be less than or equal to 255.

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

Note

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

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



	First byte	Second byte	Third byte	Fourth byte
Class A	0			
Class B	10	🟨		
Class C	110	🟨	🟨	
Class D	1110			
Class E	1111			

a. Binary notation

	First byte	Second byte	Third byte	Fourth byte
Class A	0–127			
Class B	128–191	🟨		
Class C	192–223	🟨	🟨	
Class D	224–239			
Class E	240–255			

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

Solution

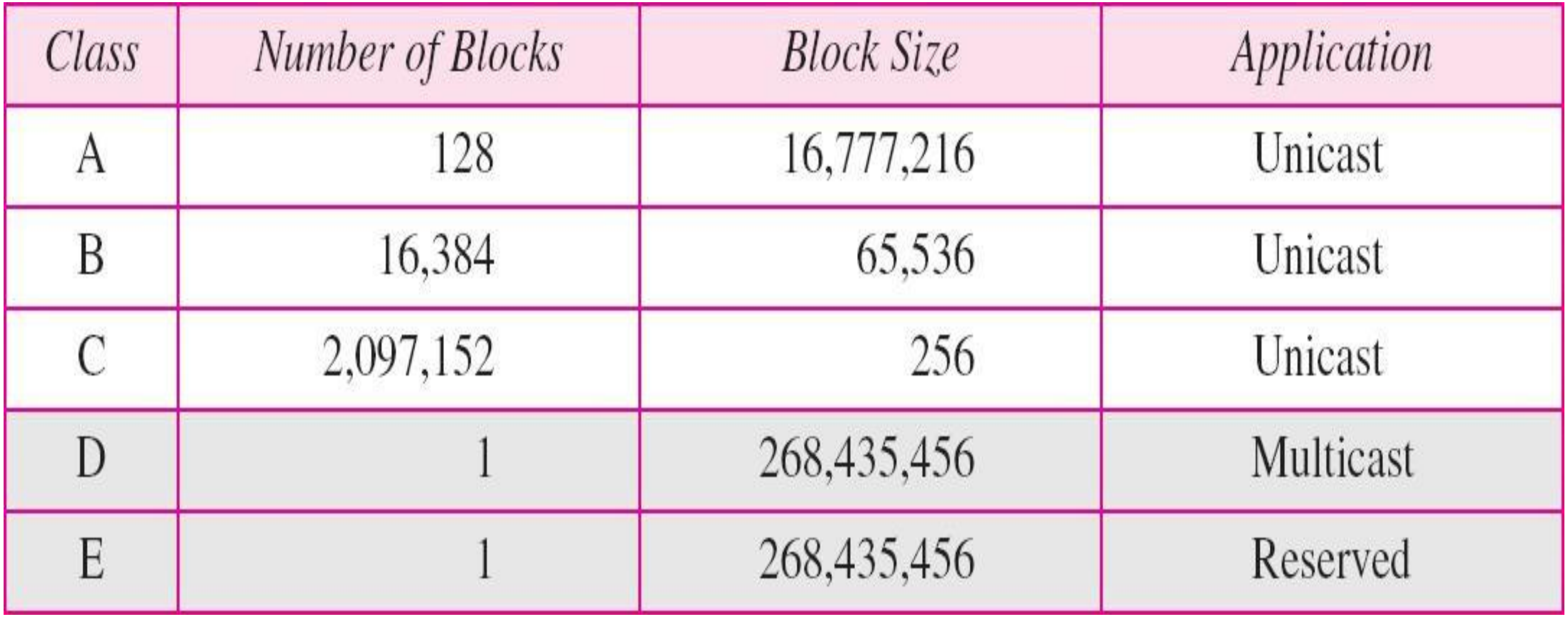
a. The first bit is 0. This is a class Aaddress.

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.

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

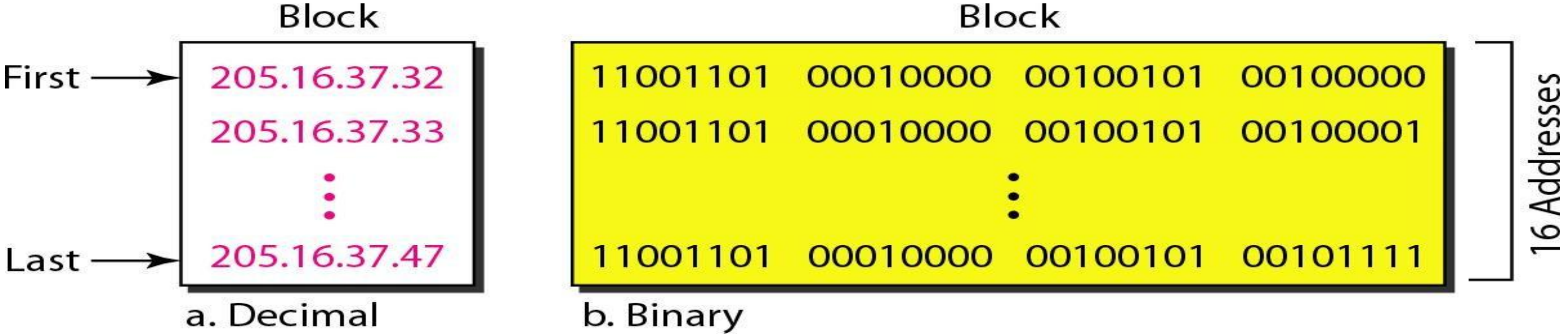
Note

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

Note

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

Classless Addresses :*Ablock of 16 addresses granted to a small organization*



205.16.37.32
205.16.37.33
⋮
205.16.37.47

a. Decimal

11001101 00010000 00100101 00100000
11001101 00010000 00100101 00100001
⋮
11001101 00010000 00100101 00101111

b. Binary

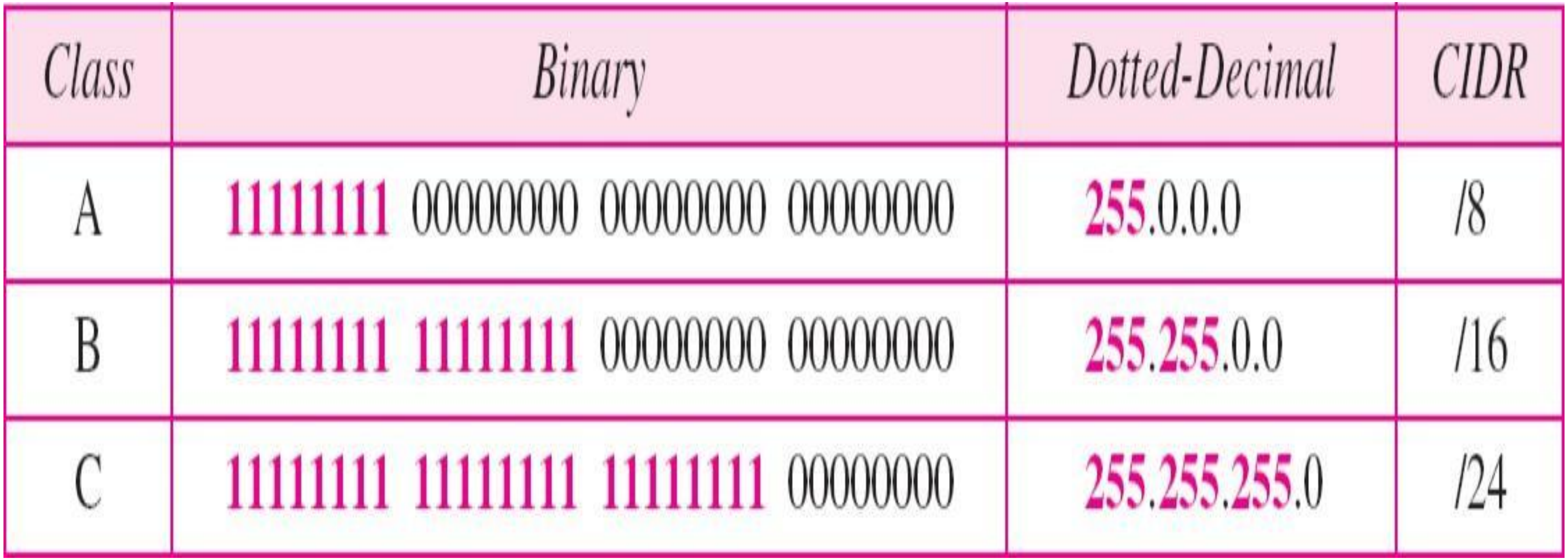
Note

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.

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

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 00100000

or 205.16.37.32.

.

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}.

Figure 19.4 *Anetwork configuration for the block 205.16.37.32/28*

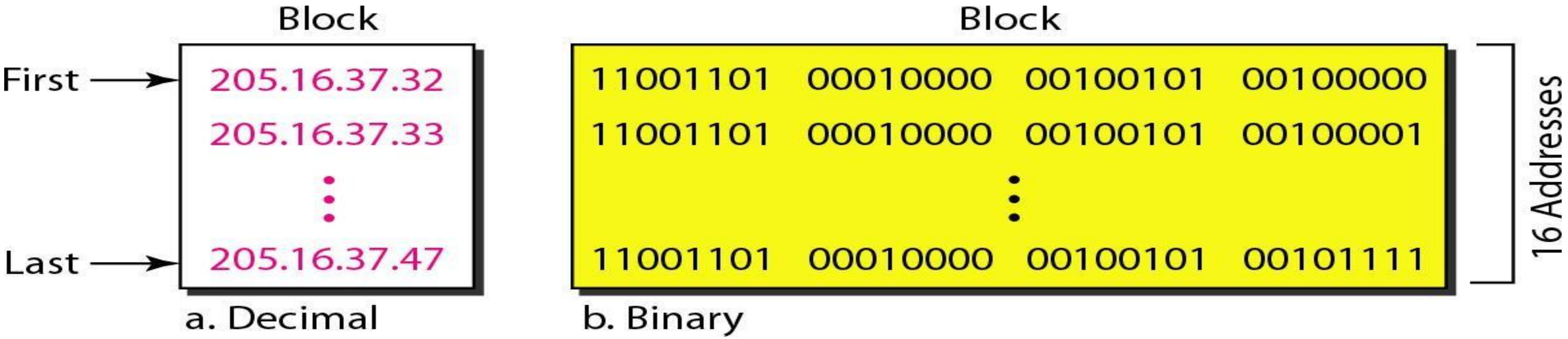
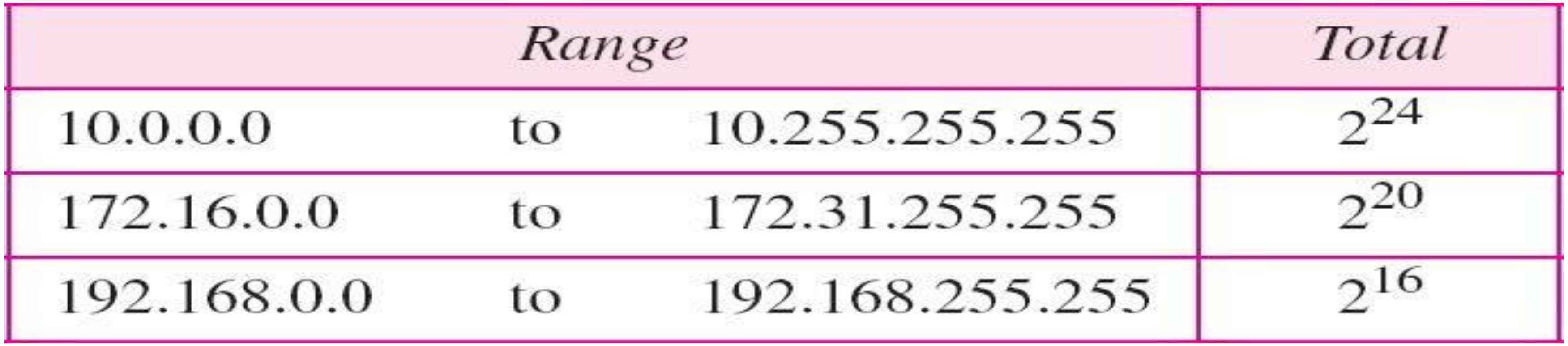
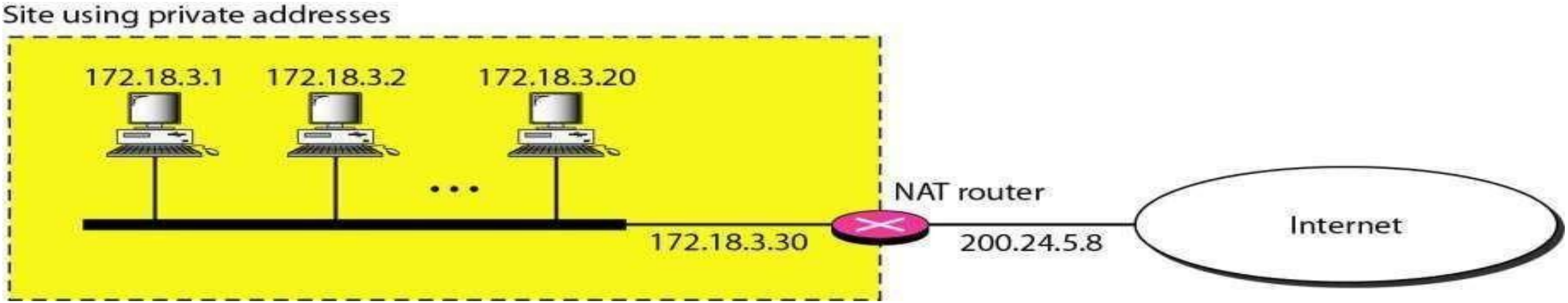


Table 19.3 *Addresses for private networks*

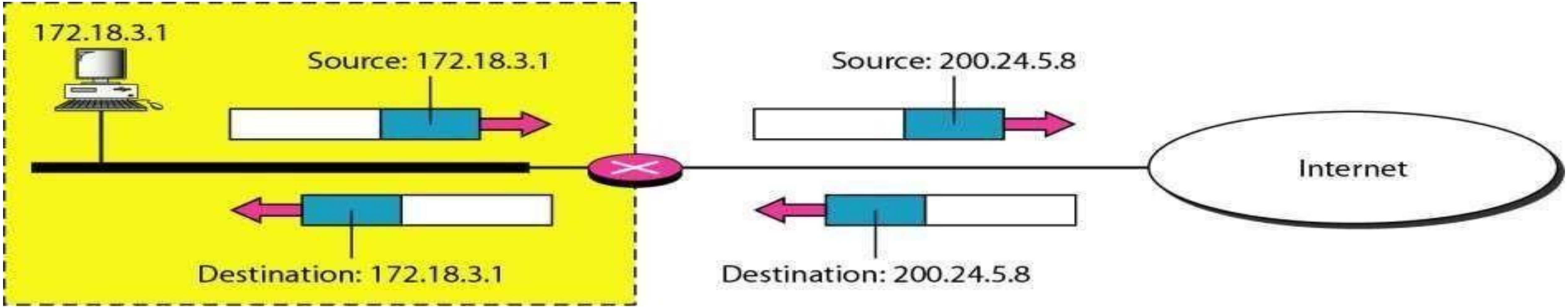


Range			*Total*
10.0.0.0	to	10.255.255.255	2^{24}
172.16.0.0	to	172.31.255.255	2^{20}
192.168.0.0	to	192.168.255.255	2^{16}

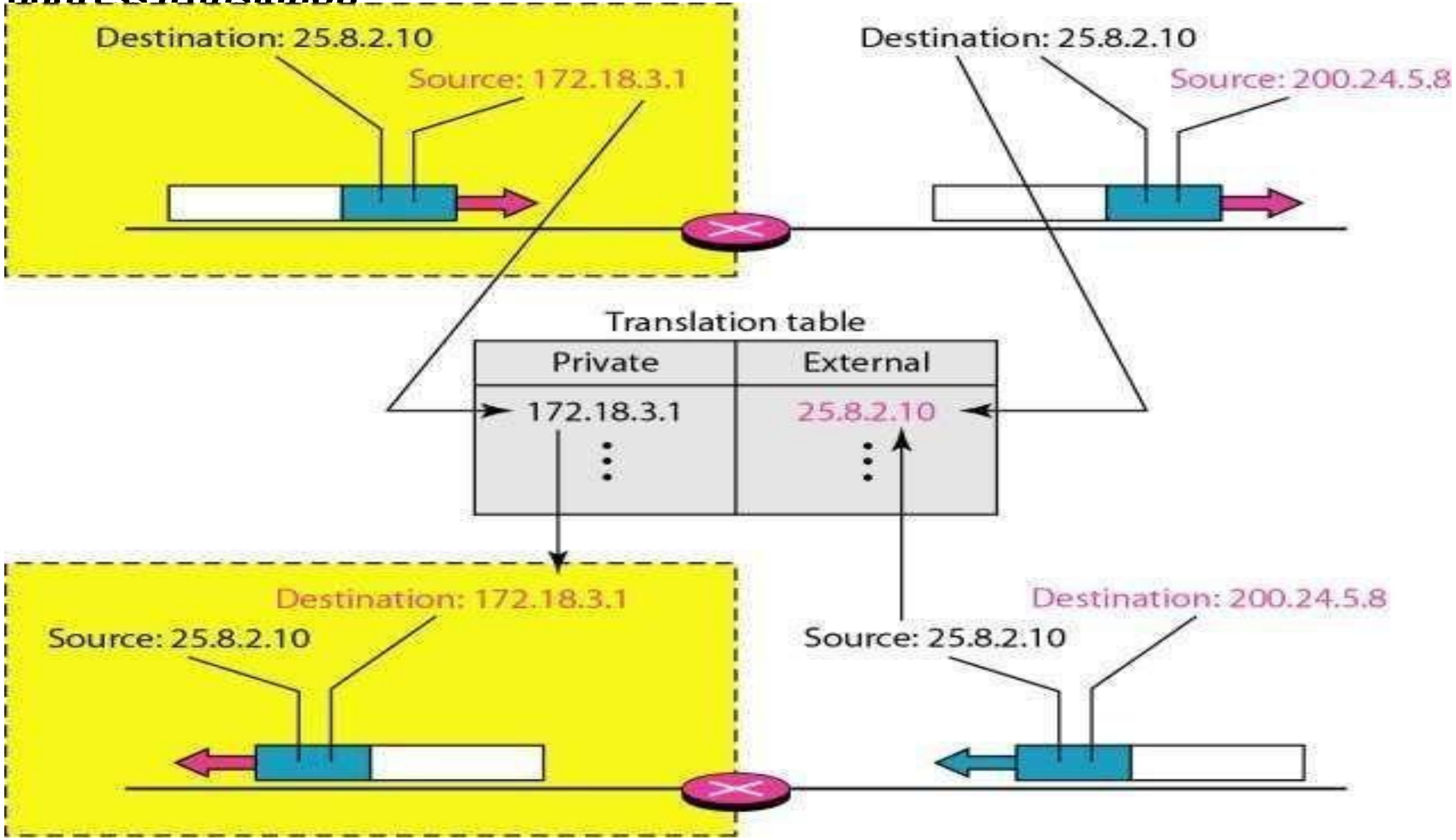
ANAT implementation



Addresses in a NAT



NAT address translation



IPv6 ADDRESSES

IPv6 ADDRESSES

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

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

Topics discussed in this section:

Topics discussed in this section:

Structure Address Space

Note

An IPv6 address is 128 bits long.

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

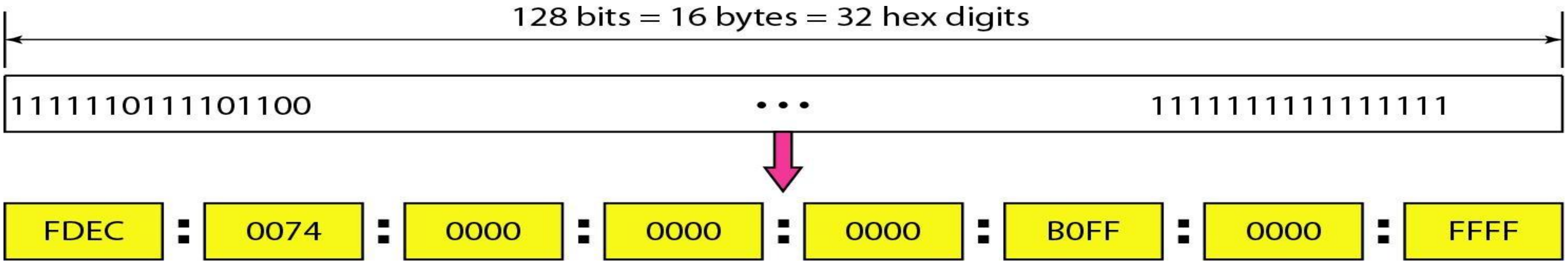
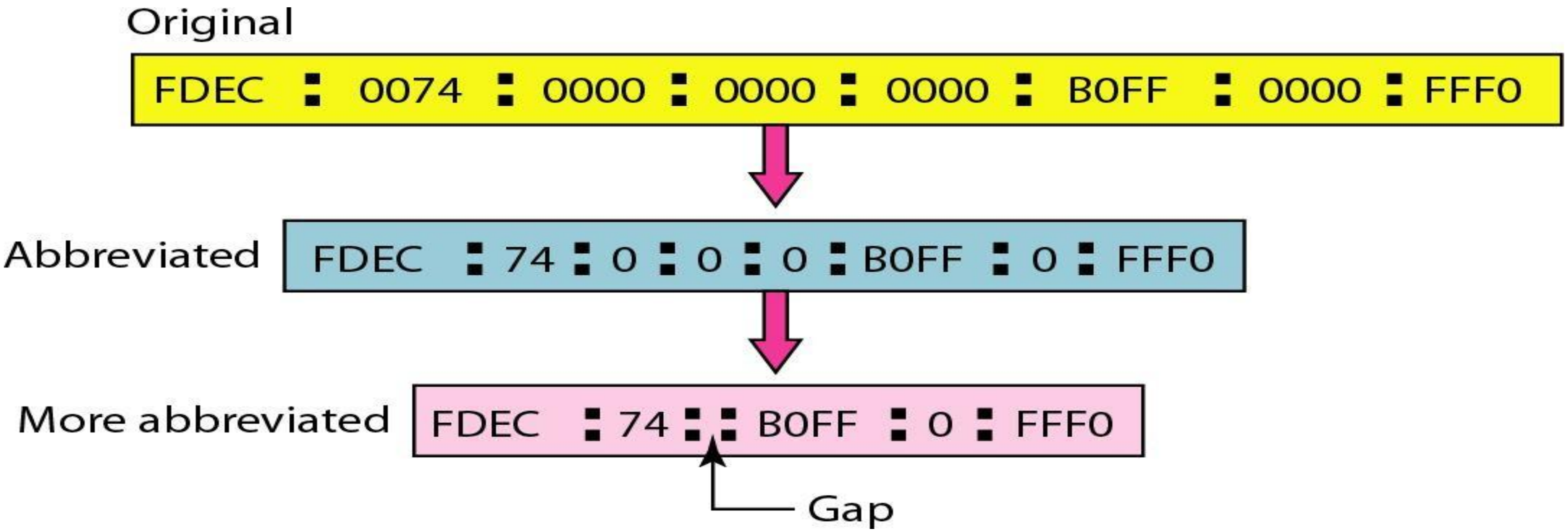


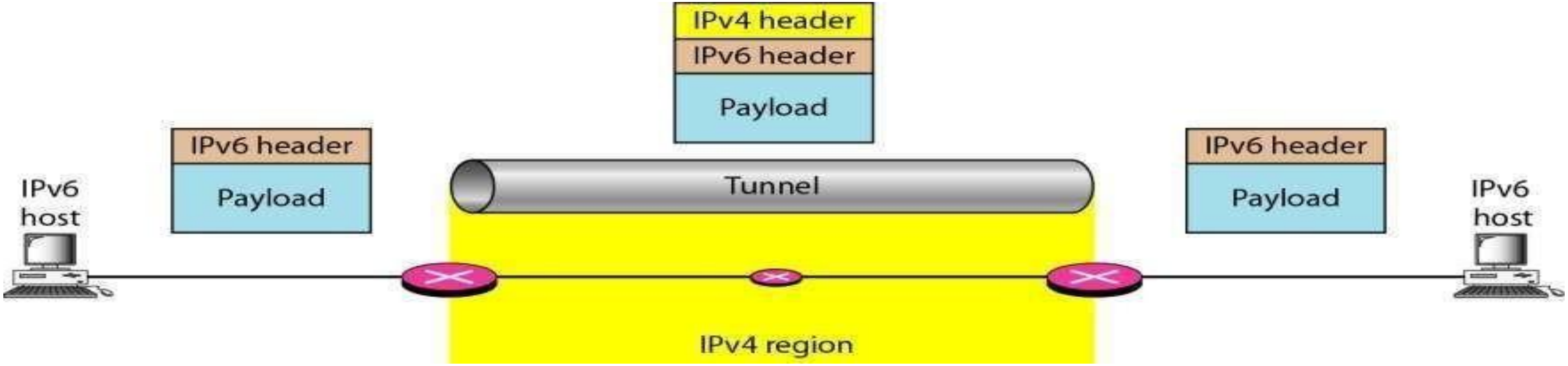
Figure 19.15 *Abbreviated IPv6 addresses*


Original
FDEC : 0074 : 0000 : 0000 : 0000 : B0FF : 0000 : FFF0
Abbreviated
FDEC : 74 : 0 : 0 : 0 : B0FF : 0 : FFF0
More abbreviated
FDEC : 74 : : B0FF : 0 : FFF0
Gap

Chapter 20

- Network Layer: Internet Protocol

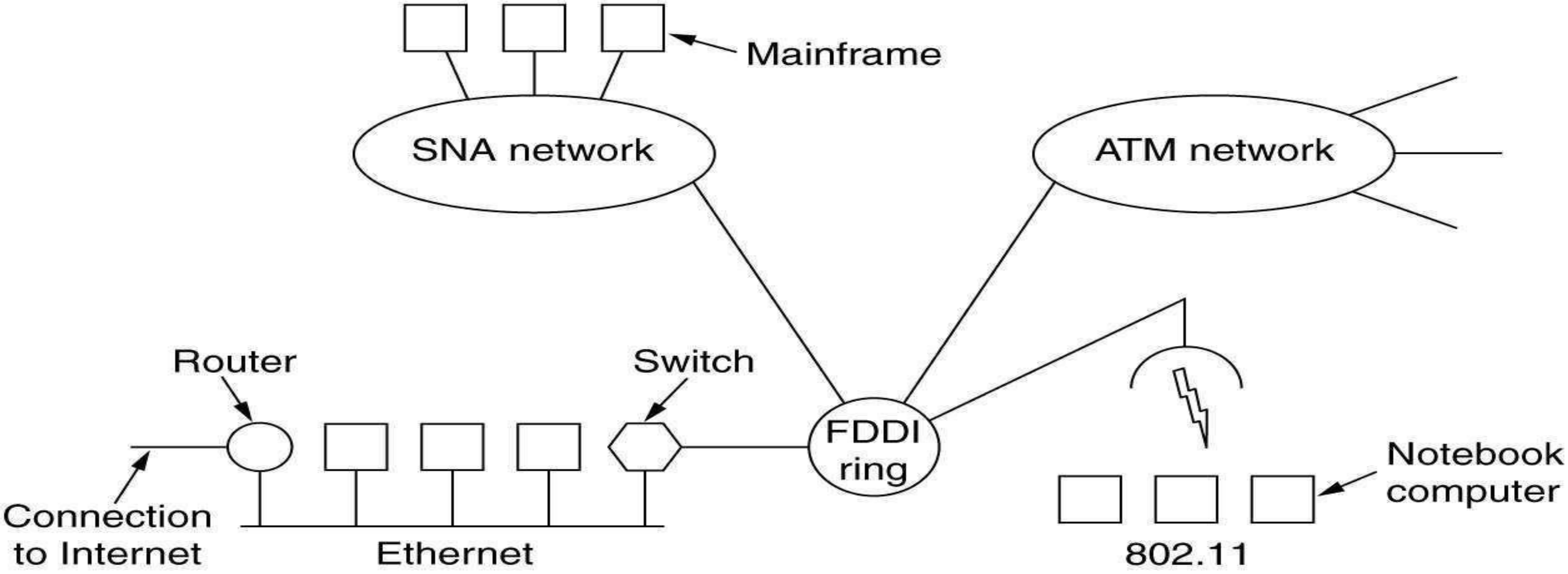
Tunneling


IPv4 header
IPv6 header
Payload
IPv6 header
Payload
Tunnel
IPv6 header
Payload
IPv6 host
IPv6 host
IPv4 region

Internetworking

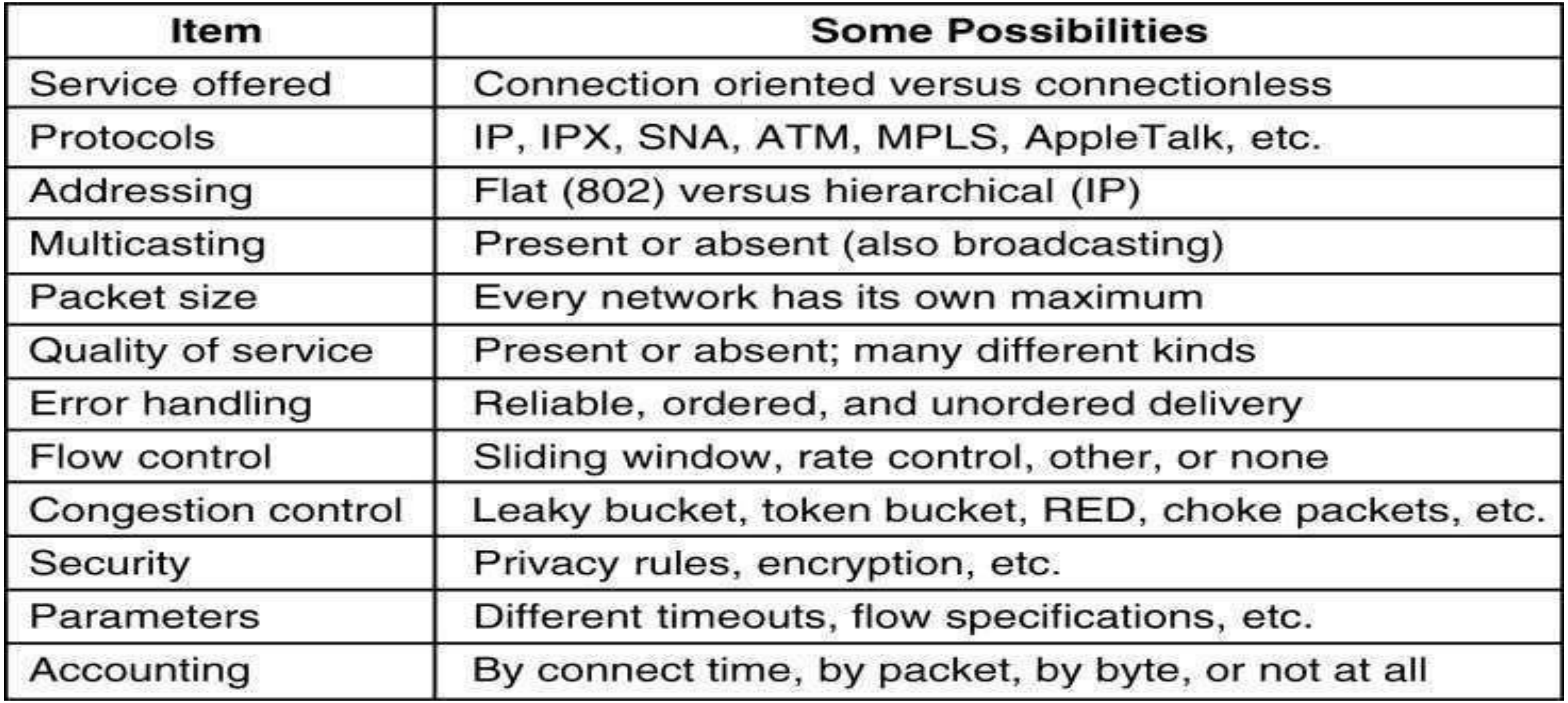
- How Networks Differ
- How Networks Can Be Connected
- Concatenated Virtual Circuits
- Connectionless Internetworking
- Tunneling
- Internetwork Routing
- Fragmentation

Connecting Networks



A collection of interconnected networks.

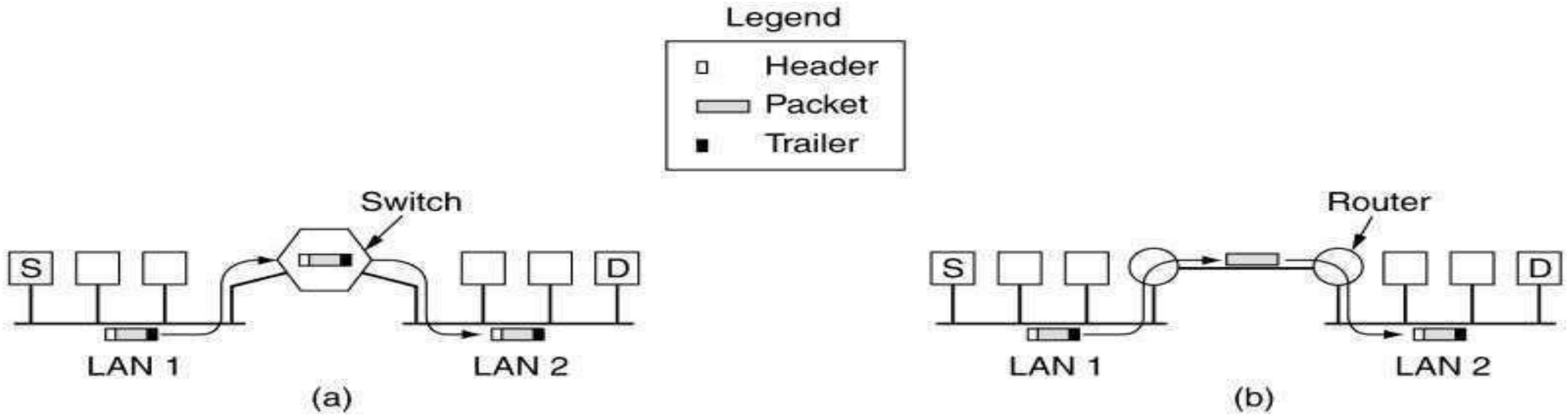
How Networks Differ



Item	Some Possibilities
Service offered	Connection oriented versus connectionless
Protocols	IP, IPX, SNA, ATM, MPLS, AppleTalk, etc.
Addressing	Flat (802) versus hierarchical (IP)
Multicasting	Present or absent (also broadcasting)
Packet size	Every network has its own maximum
Quality of service	Present or absent; many different kinds
Error handling	Reliable, ordered, and unordered delivery
Flow control	Sliding window, rate control, other, or none
Congestion control	Leaky bucket, token bucket, RED, choke packets, etc.
Security	Privacy rules, encryption, etc.
Parameters	Different timeouts, flow specifications, etc.
Accounting	By connect time, by packet, by byte, or not at all

Some of the many ways networks can differ.

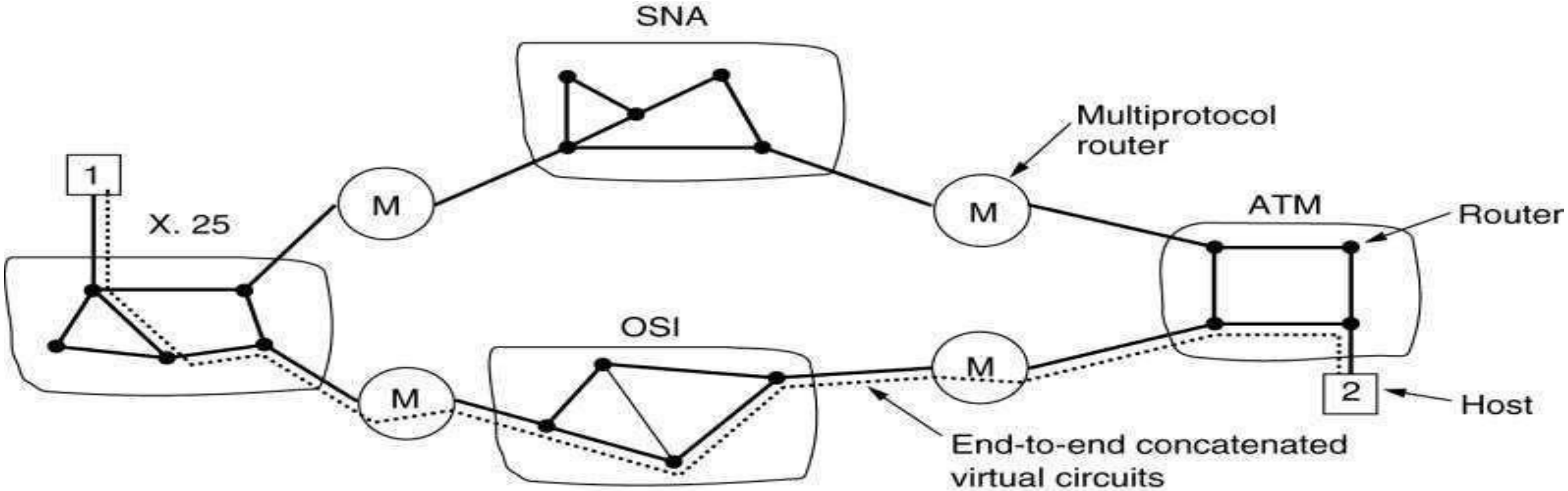
How Networks Can Be Connected



(a) Two Ethernets connected by a switch.

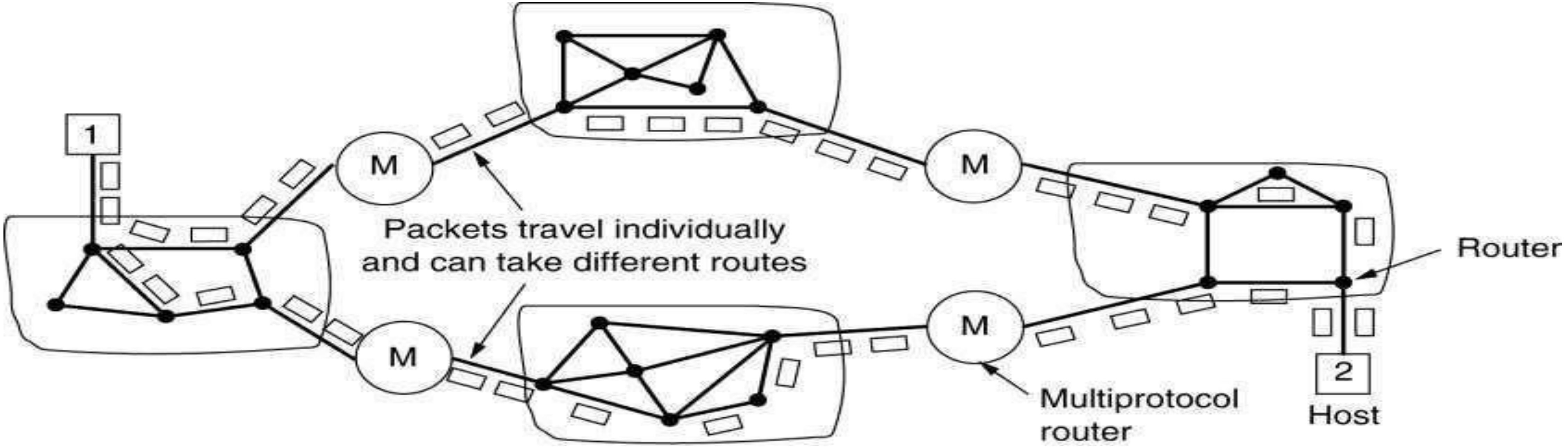
(b) Two Ethernets connected by routers.

Concatenated Virtual Circuits



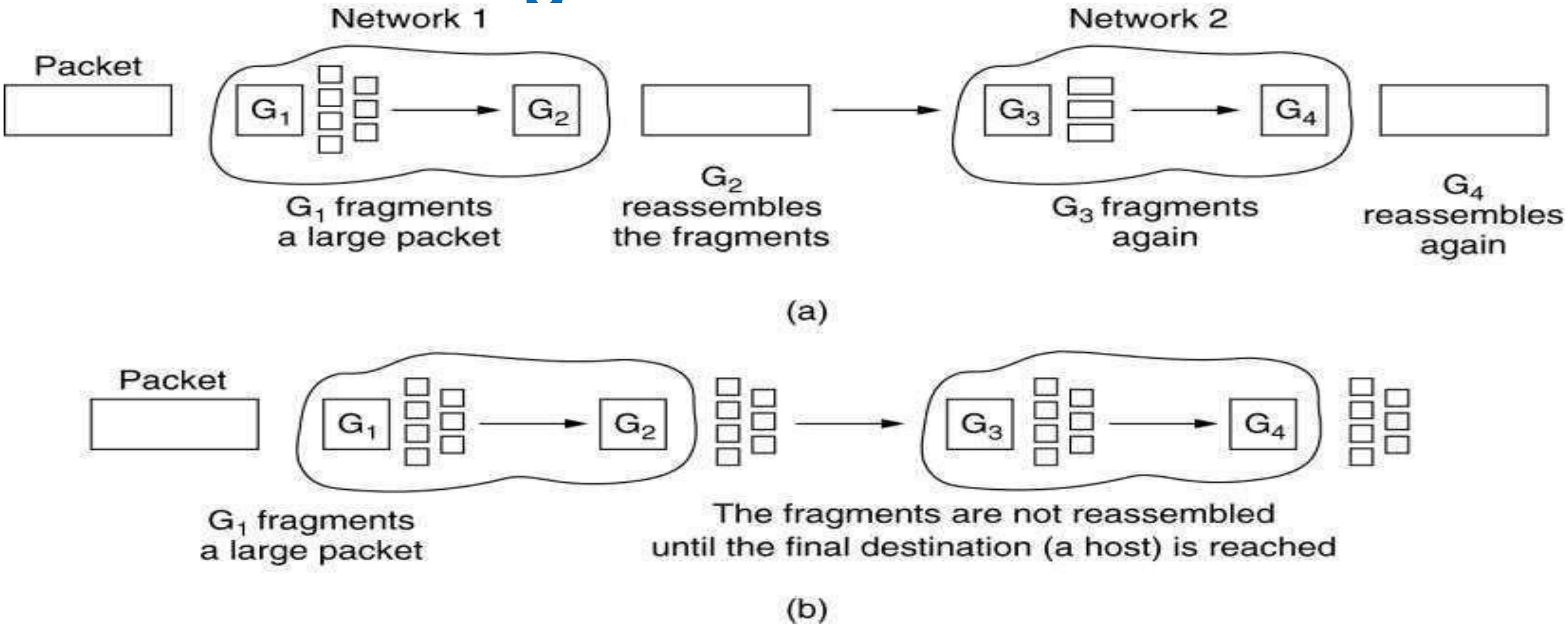
Internetworking using concatenated virtual circuits.

Connectionless Internetworking



A connectionless internet.

Fragmentation


Network 1
Network 2
Packet
G_1
G_2
G_3
G_4
G_1 fragments a large packet
G_2 reassembles the fragments
G_3 fragments again
G_4 reassembles again
(a)
Packet
G_1
G_2
G_3
G_4
G_1 fragments a large packet
The fragments are not reassembled until the final destination (a host) is reached
(b)

(a) Transparent fragmentation. (b) Nontransparent fragmentation.

Network Layer: Address Mapping

ADDRESS MAPPING

ADDRESS MAPPING

The delivery of a packet to a host or a router requires two levels of addressing: logical and physical. We need to be able to map a logical address to its corresponding physical address and vice versa. This can be done by using either static or dynamic mapping.

can be done by using either static or dynamic mapping.

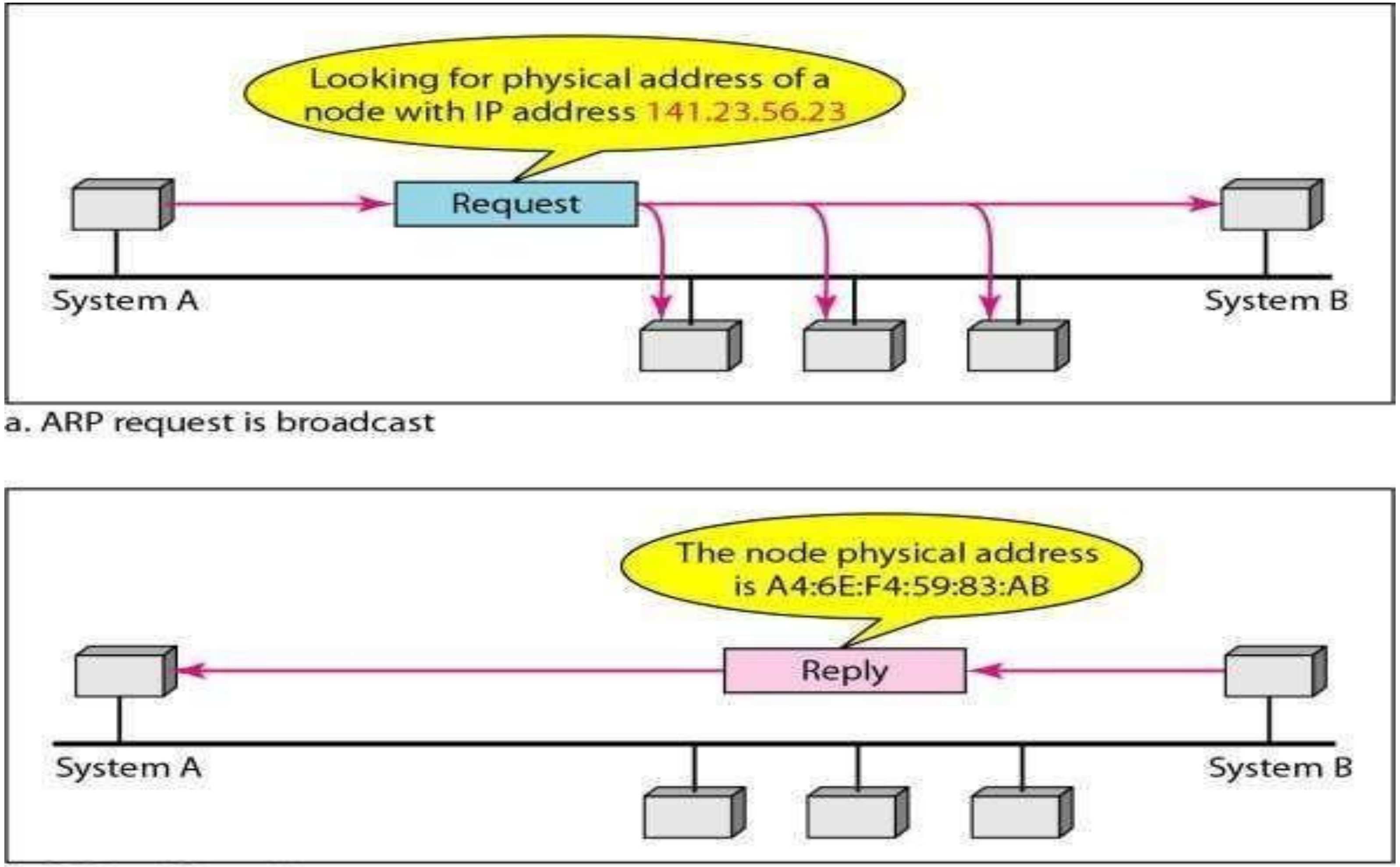
Topics discussed in this section: **Mapping Logical to Physical Address Mapping Physical to Logical Address**

Topics discussed in this section:

Note

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

Figure 21.1 *ARP operation*



Looking for physical address of a node with IP address 141.23.56.23
Request
System A
System B

a. ARP request is broadcast

The node physical address is A4:6E:F4:59:83:AB
Reply
System A
System B

b. ARP reply is unicast

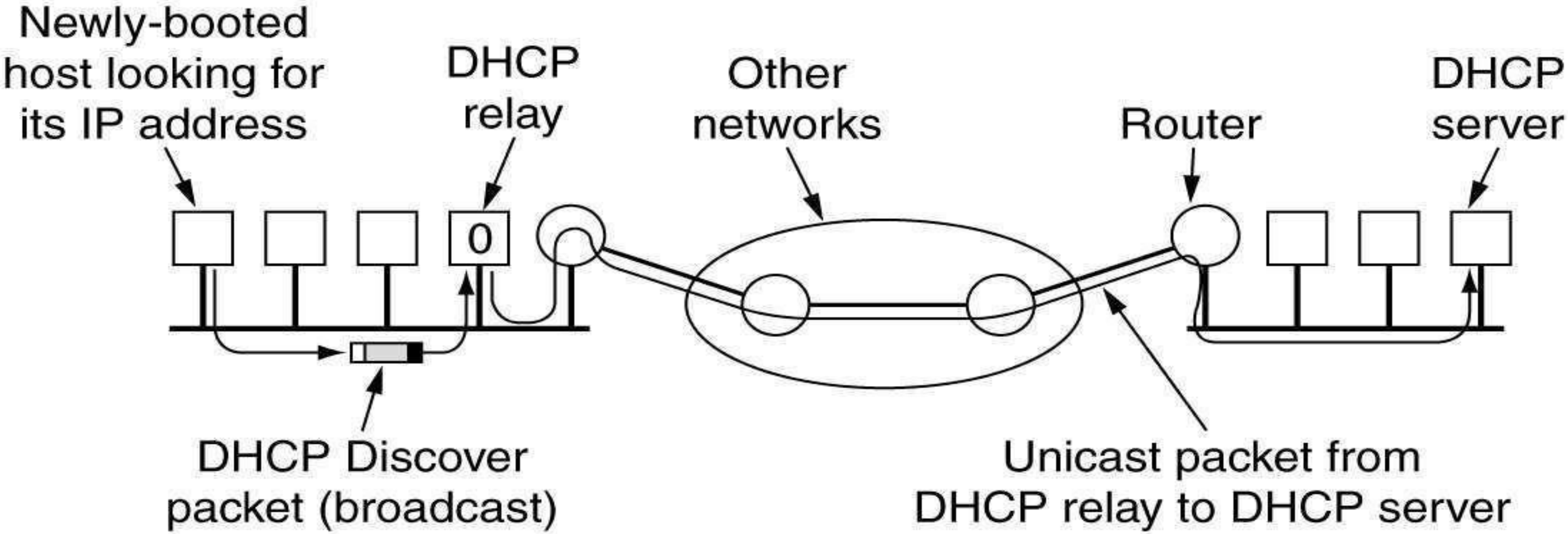
Reverse Address Resolution Protocol (RARP)

TO find logical address for a machine that knows only its physical address

Note

DHCP provides static and dynamic address allocation that can be manual or automatic.

Dynamic Host Configuration Protocol


Newly-booted host looking for its IP address
DHCP relay
Other networks
Router
DHCP server
0
DHCP Discover packet (broadcast)
Unicast packet from DHCP relay to DHCP server

Operation of DHCP.

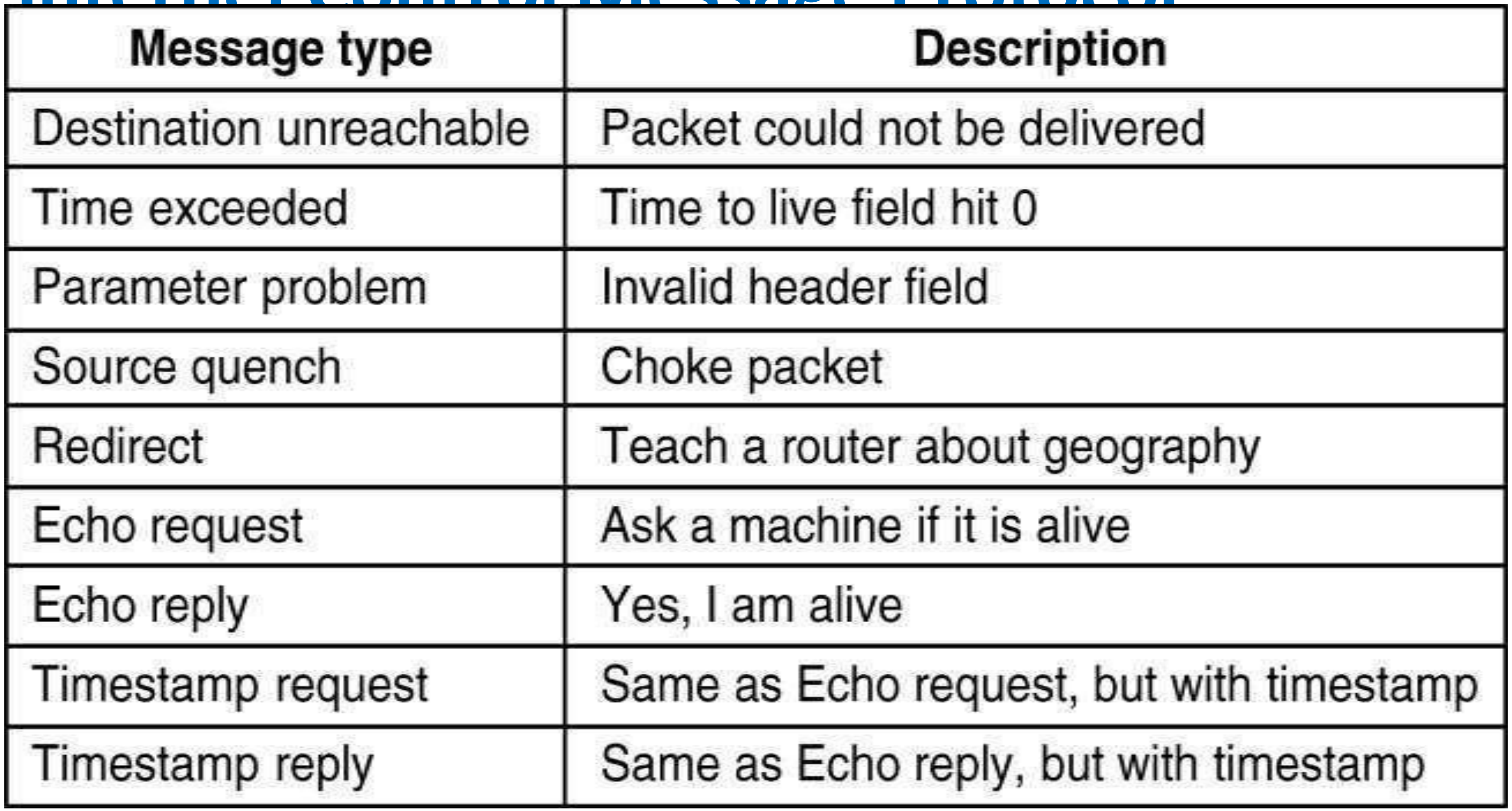
ICMP

The IP protocol has no error-reporting or error- correcting mechanism. The IP protocol also lacks a mechanism for host and management queries. The ***Internet Control Message Protocol (ICMP)*** ***has been designed to compensate for the above two deficiencies. It is a companion to the IP protocol.***

Topics discussed in this section:

Types of Messages Message Format
Error Reporting and Query Debugging Tools

Internet Control Message Protocol



Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo request	Ask a machine if it is alive
Echo reply	Yes, I am alive
Timestamp request	Same as Echo request, but with timestamp
Timestamp reply	Same as Echo reply, but with timestamp

The principal ICMP message types.

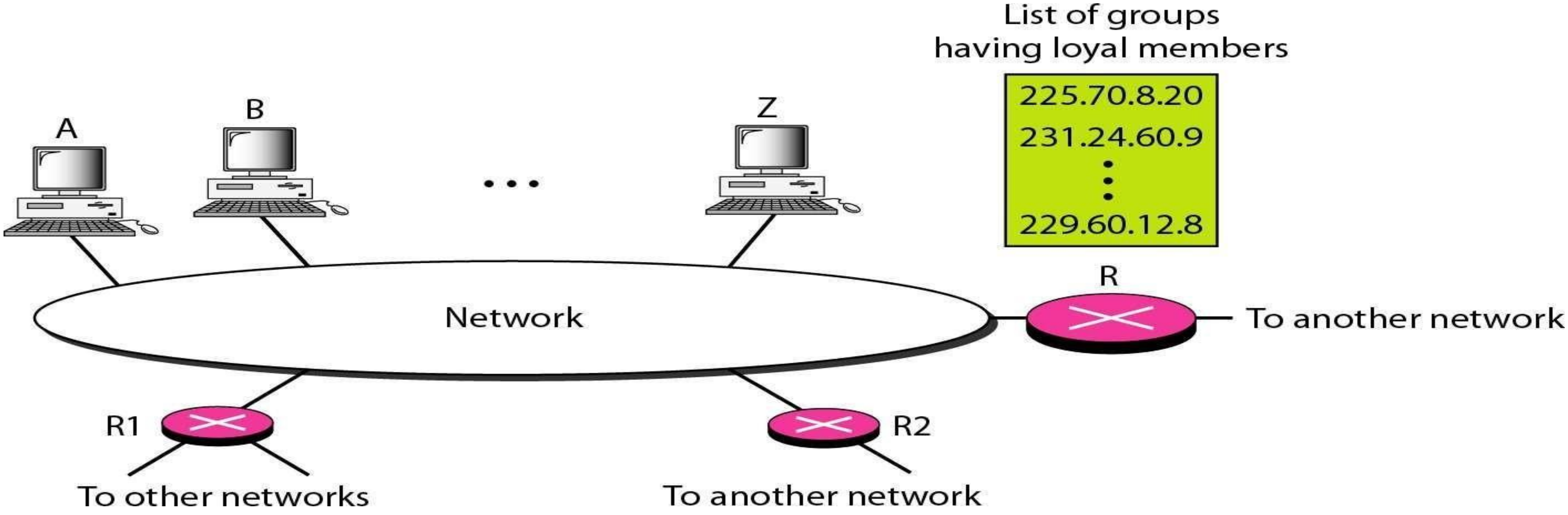
IGMP

The IP protocol can be involved in two types of communication: unicasting and multicasting. The Internet Group Management Protocol (IGMP) is one of the necessary, but not sufficient, protocols that is involved in multicasting. IGMP is a companion to the IP protocol.

IGMP

The IP protocol can be involved in two types of communication: unicasting and multicasting. The Internet Group Management Protocol (IGMP) is one of the necessary, but not sufficient, protocols that is involved in multicasting. IGMP is a companion to the IP protocol.

IGMP operation

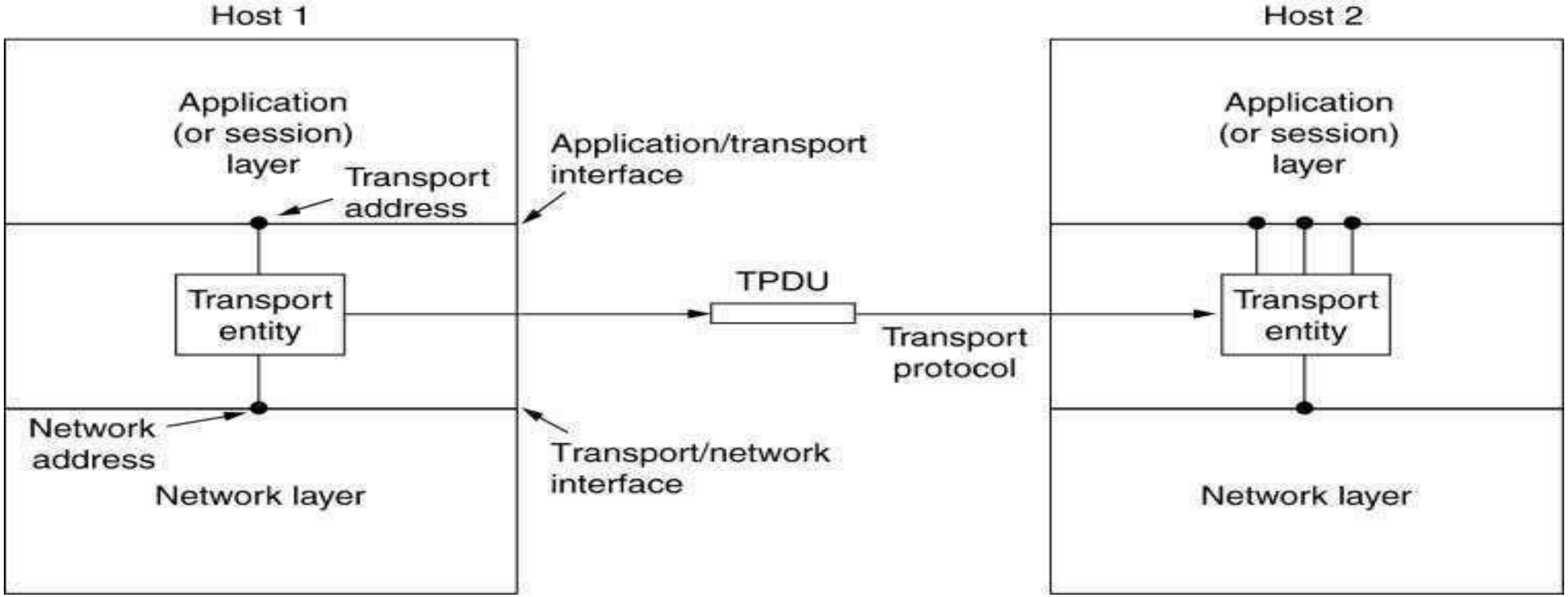


The Transport Layer

The Transport Service

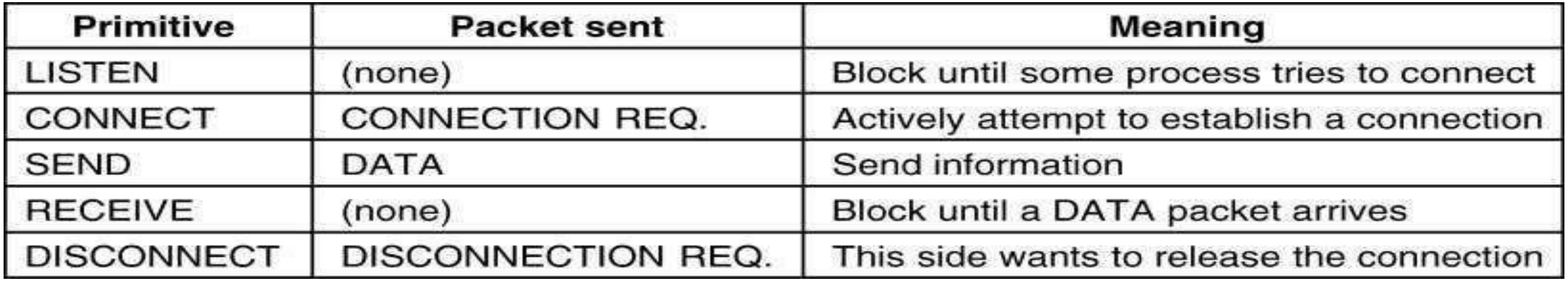
- Services Provided to the Upper Layers
- Transport Service Primitives

Services Provided to the Upper Layers



The network, transport, and application layers.

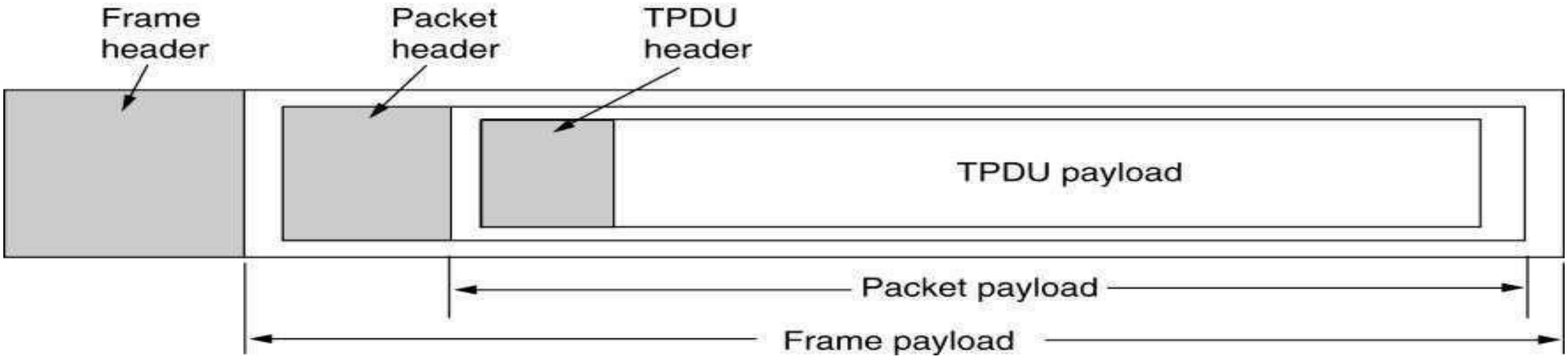
Transport Service Primitives



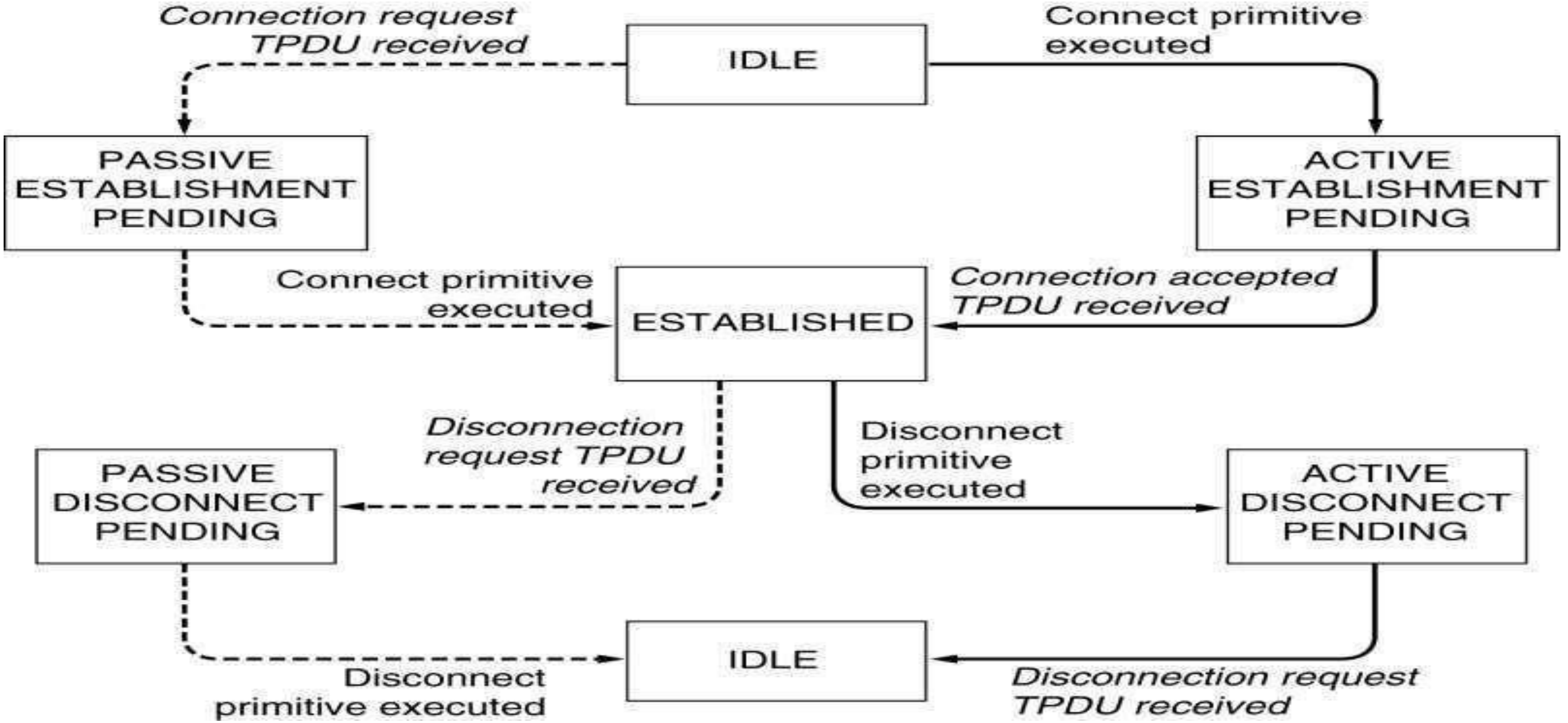
Primitive	Packet sent	Meaning
LISTEN	(none)	Block until some process tries to connect
CONNECT	CONNECTION REQ.	Actively attempt to establish a connection
SEND	DATA	Send information
RECEIVE	(none)	Block until a DATA packet arrives
DISCONNECT	DISCONNECTION REQ.	This side wants to release the connection

The primitives for a simple transport service.

Transport Service Primitives (2)



The nesting of TPDUs, packets, and frames.

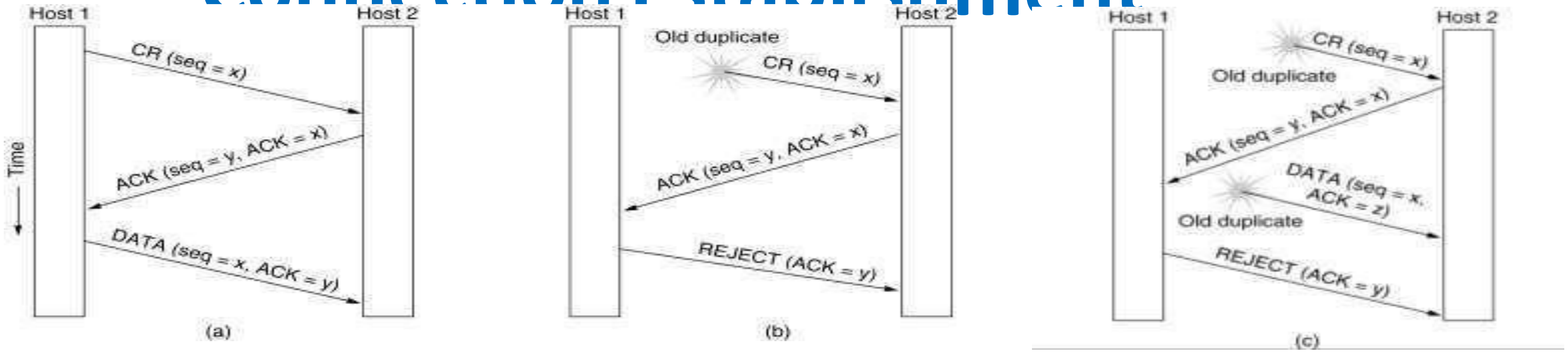

Connection request TPDU received
IDLE
Connect primitive executed
PASSIVE ESTABLISHMENT PENDING
ACTIVE ESTABLISHMENT PENDING
Connect primitive executed
ESTABLISHED
Connection accepted TPDU received
Disconnection request TPDU received
Disconnect primitive executed
PASSIVE DISCONNECT PENDING
ACTIVE DISCONNECT PENDING
IDLE
Disconnect primitive executed
Disconnection request TPDU received

A state diagram for a simple connection management scheme. Transitions labeled in italics are caused by packet arrivals. The **solid lines** show the **client's state sequence.** The **dashed lines** show the **server's state sequence.**

Elements of Transport Protocols

- Addressing
- Connection Establishment
- Connection Release
- Flow Control and Buffering
- Multiplexing
- Crash Recovery

Connection Establishment



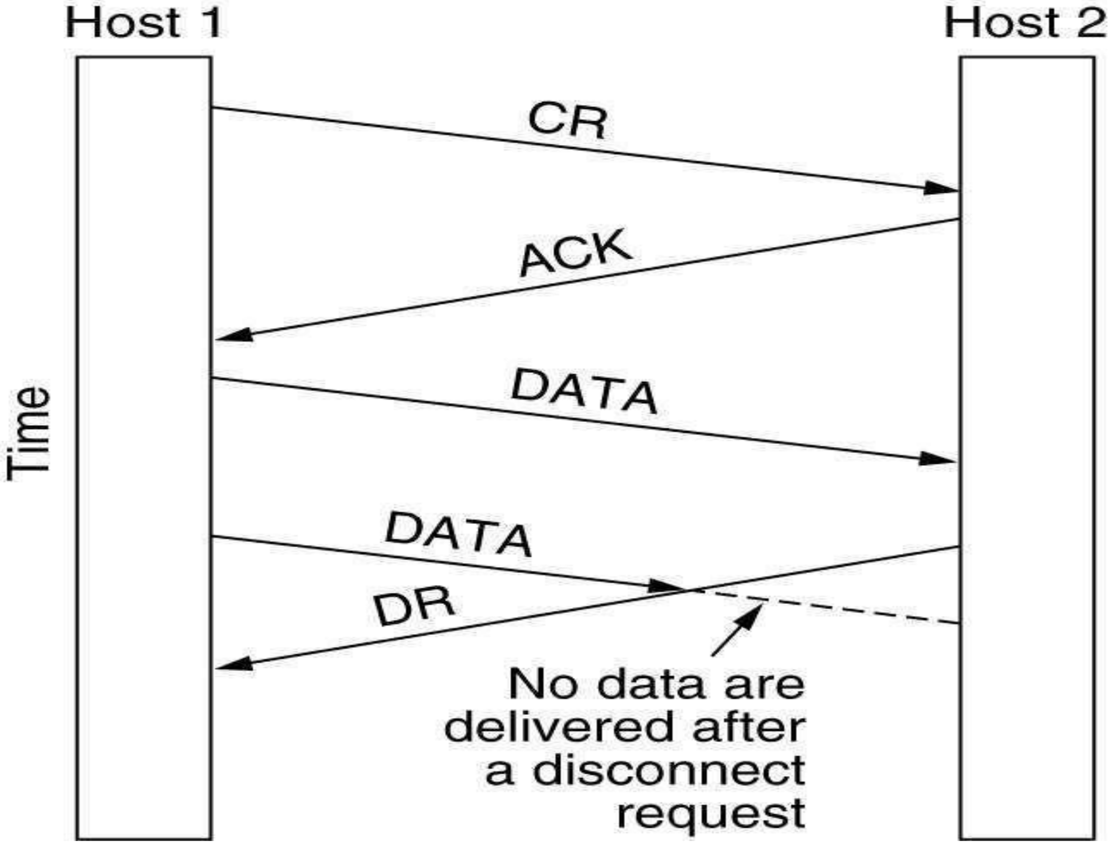
Three protocol scenarios for establishing a connection using a three-way handshake. CR denotes CONNECTION REQUEST.

(a) Normal operation,

(b) Old CONNECTION REQUEST appearing out of nowhere.

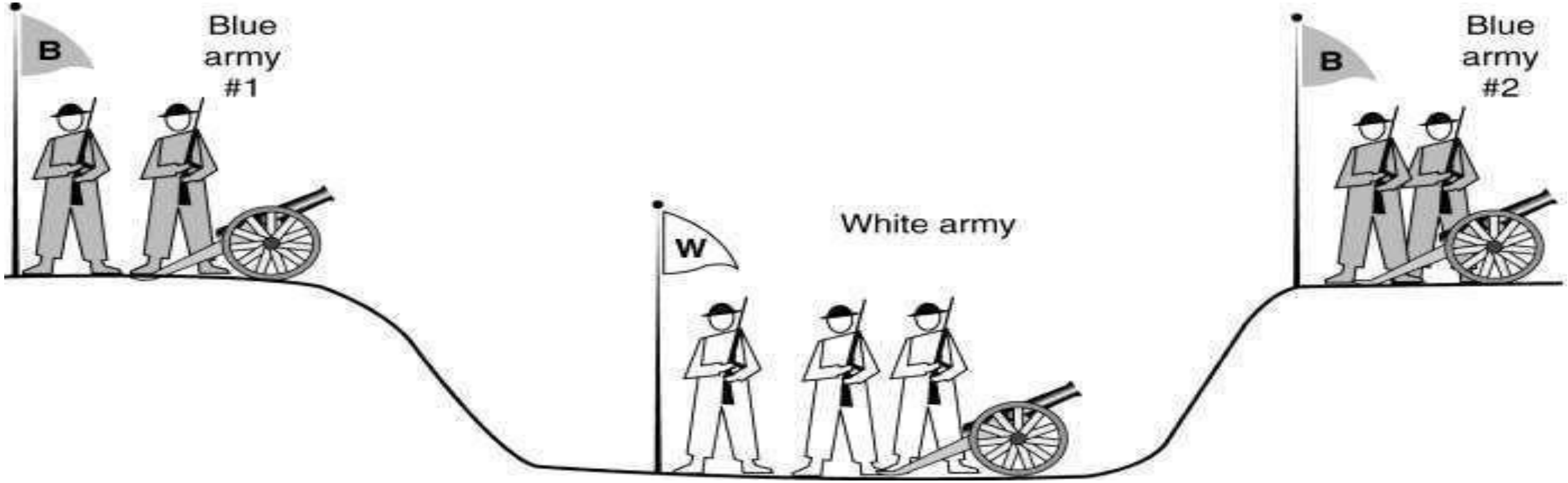
(c) Duplicate CONNECTION REQUEST and duplicate ACK.

Connection Release



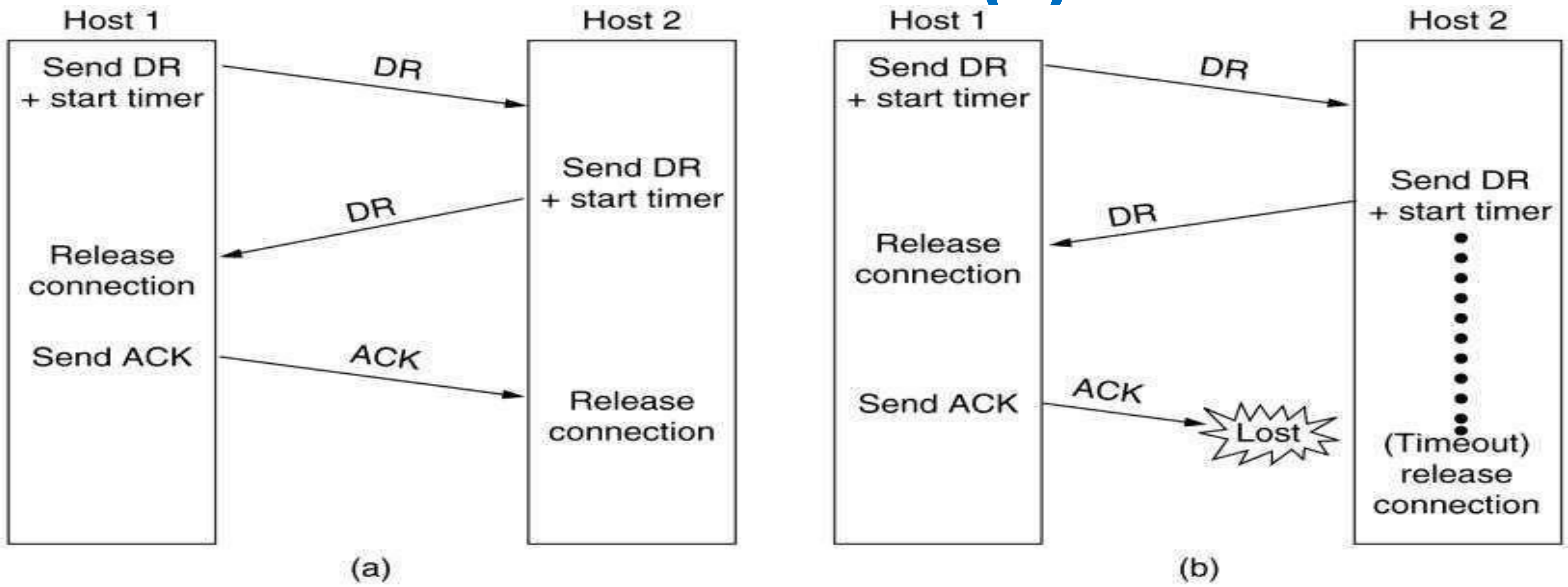
Abrupt disconnection with loss of data.

Connection Release (2)



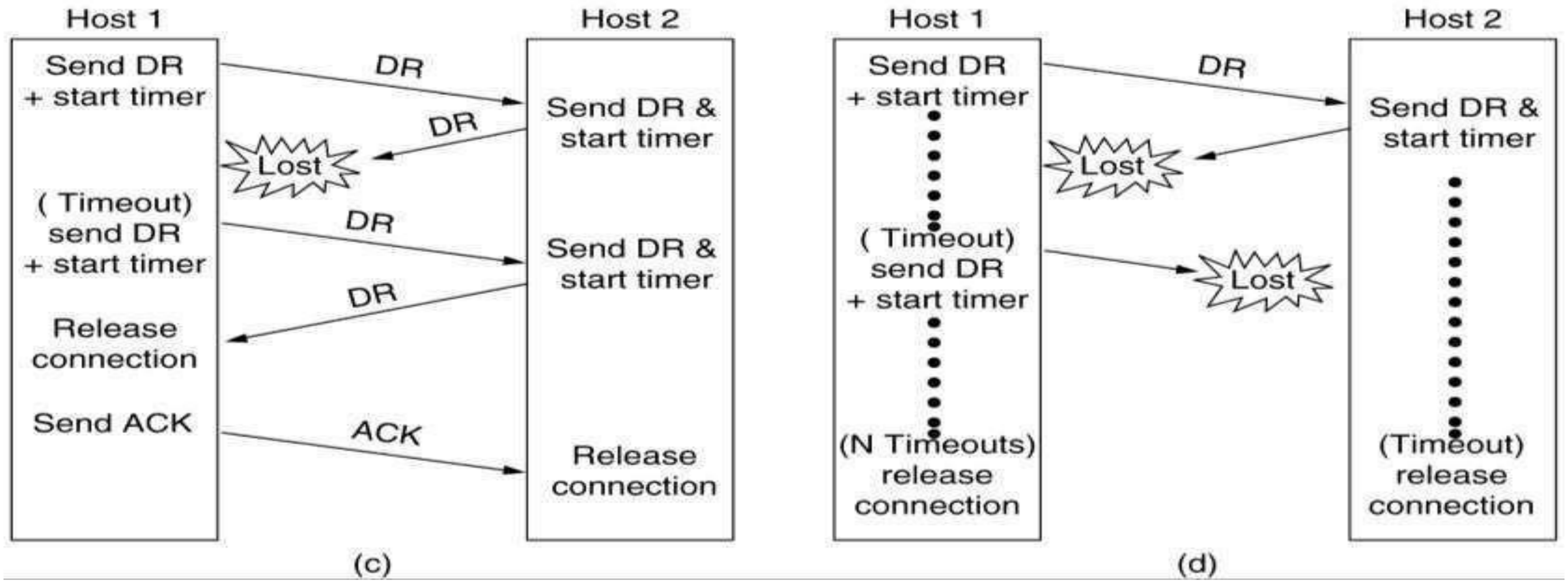
The two-army problem.

Connection Release (3)



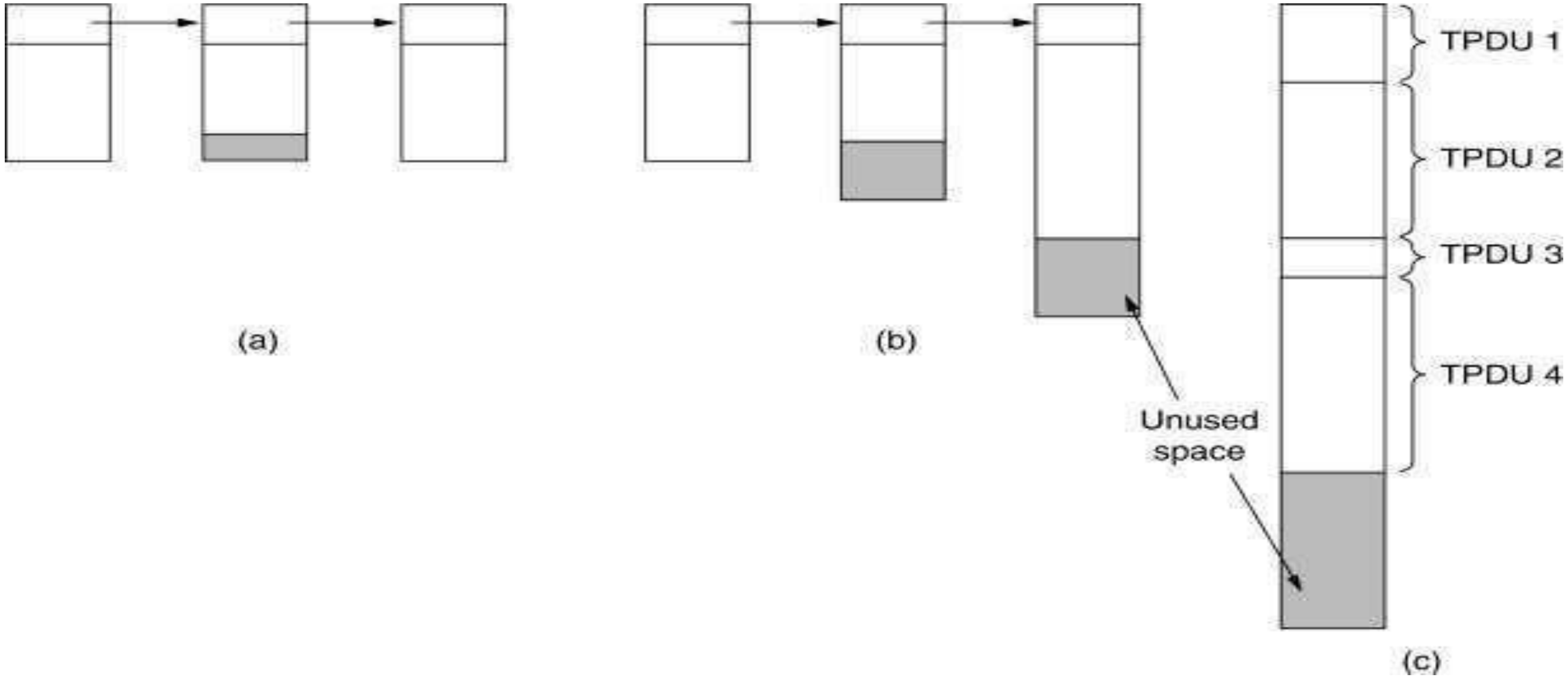
Four protocol scenarios for releasing a connection. (a) Normal case of a three-way handshake. (b) final ACK lost.

Connection Release (4)



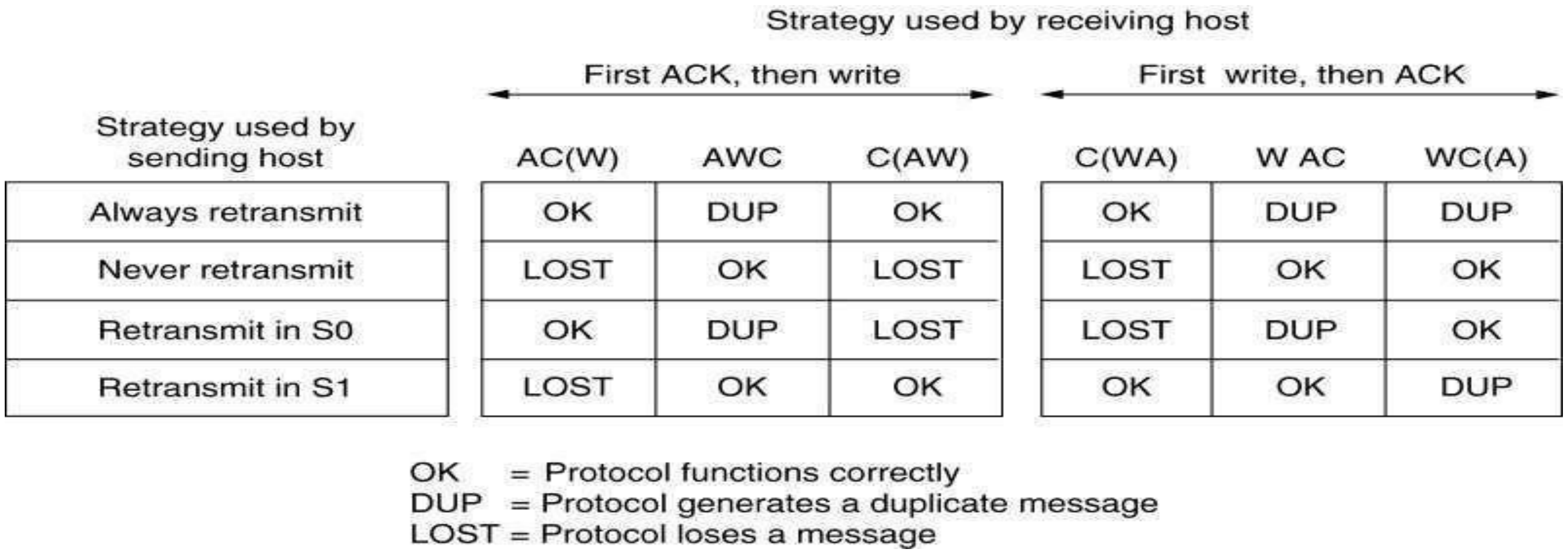
(c) Response lost. (d) Response lost and subsequent DRs lost.

Flow Control and Buffering



(a) Chained fixed-size buffers. (b) Chained variable-sized buffers.
(c) One large circular buffer per connection.

Crash Recovery



	Strategy used by receiving host					
	First ACK, then write			First write, then ACK		
Strategy used by sending host	AC(W)	AWC	C(AW)	C(WA)	W AC	WC(A)
Always retransmit	OK	DUP	OK	OK	DUP	DUP
Never retransmit	LOST	OK	LOST	LOST	OK	OK
Retransmit in S0	OK	DUP	LOST	LOST	DUP	OK
Retransmit in S1	LOST	OK	OK	OK	OK	DUP

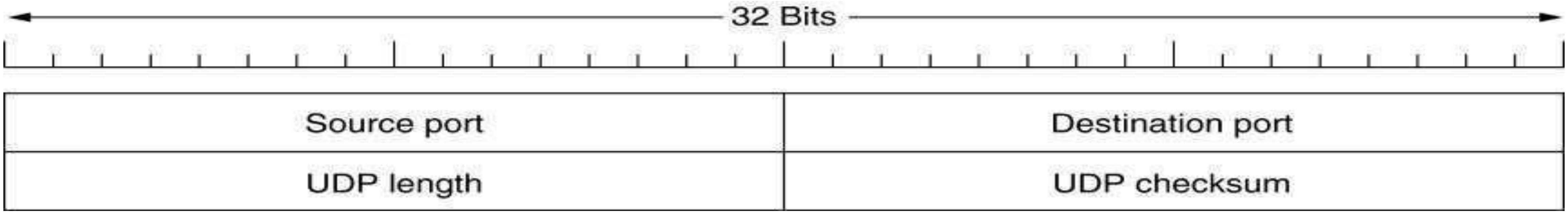
OK = Protocol functions correctly
DUP = Protocol generates a duplicate message
LOST = Protocol loses a message

Different combinations of client and server strategy.

The Internet Transport Protocols: UDP

- Introduction to UDP
- Remote Procedure Call
- The Real-Time Transport Protocol

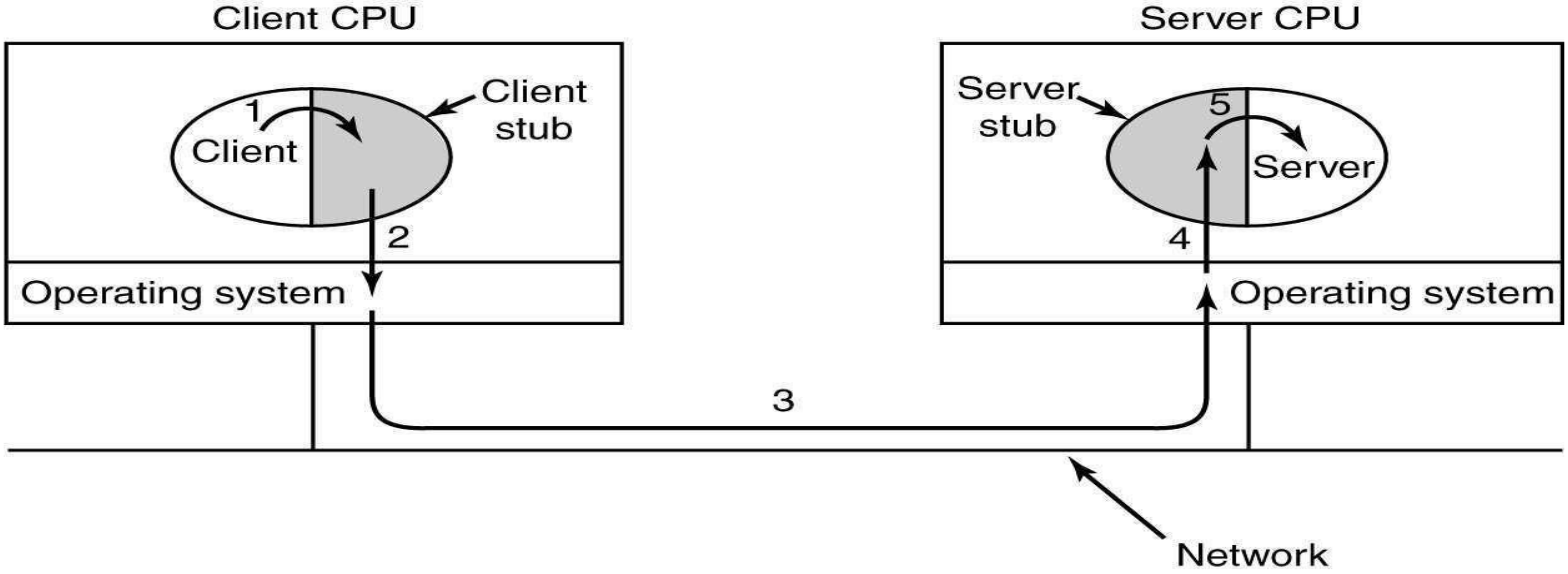
Introduction to UDP



Source port	Destination port
UDP length	UDP checksum

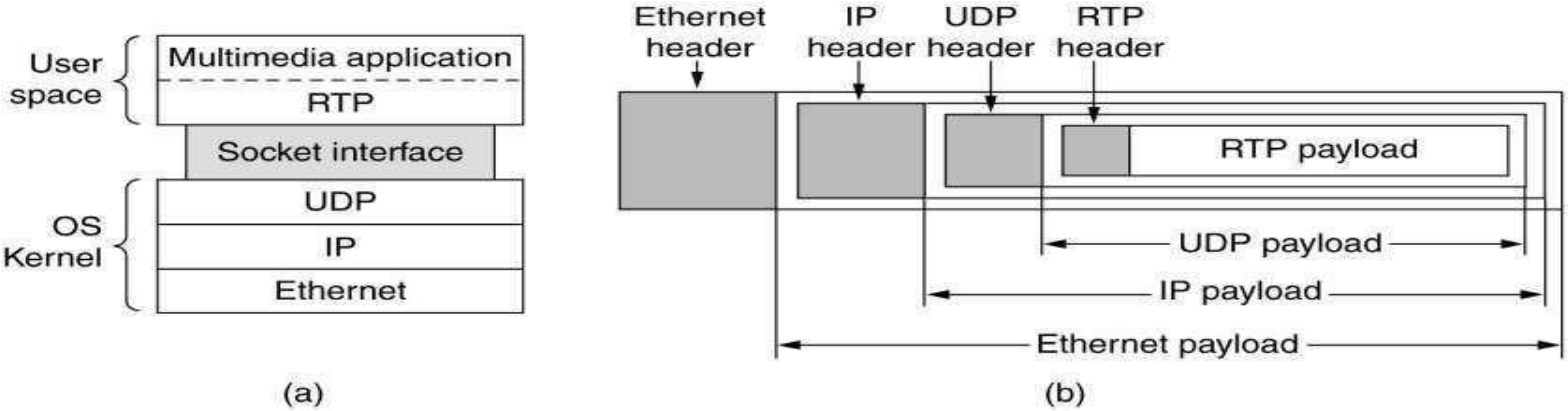
The UDP header.

Remote Procedure Call



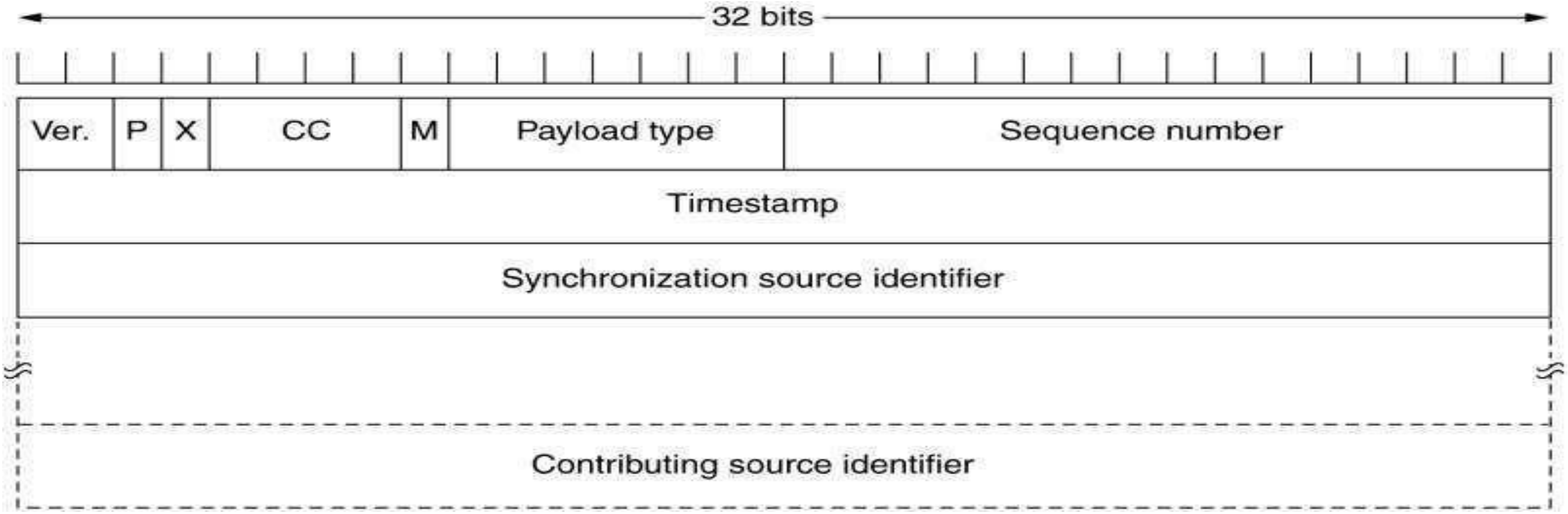
Steps in making a remote procedure call. The stubs are shaded.

The Real-Time Transport Protocol



(a) The position of RTP in the protocol stack. (b) Packet nesting.

The Real-Time Transport Protocol (2)



32 bits

Ver.	P	X	CC	M	Payload type	Sequence number
Timestamp						
Synchronization source identifier						
Contributing source identifier						

The RTP header.

The Internet Transport Protocols: TCP

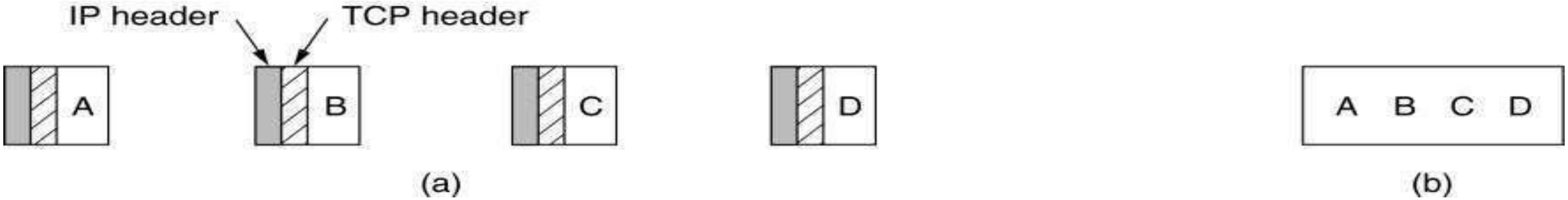
- Introduction to TCP
- The TCP Service Model
- The TCP Protocol
- The TCP Segment Header
- TCP Connection Establishment
- TCP Connection Release
- TCP Connection Management Modeling
- TCP Transmission Policy
- TCP Congestion Control
- TCP Timer Management
- Wireless TCP and UDP
- Transactional TCP

The TCP Service Model

Port	Protocol	Use
21	FTP	File transfer
23	Telnet	Remote login
25	SMTP	E-mail
69	TFTP	Trivial File Transfer Protocol
79	Finger	Lookup info about a user
80	HTTP	World Wide Web
110	POP-3	Remote e-mail access
119	NNTP	USENET news

Some assigned ports.

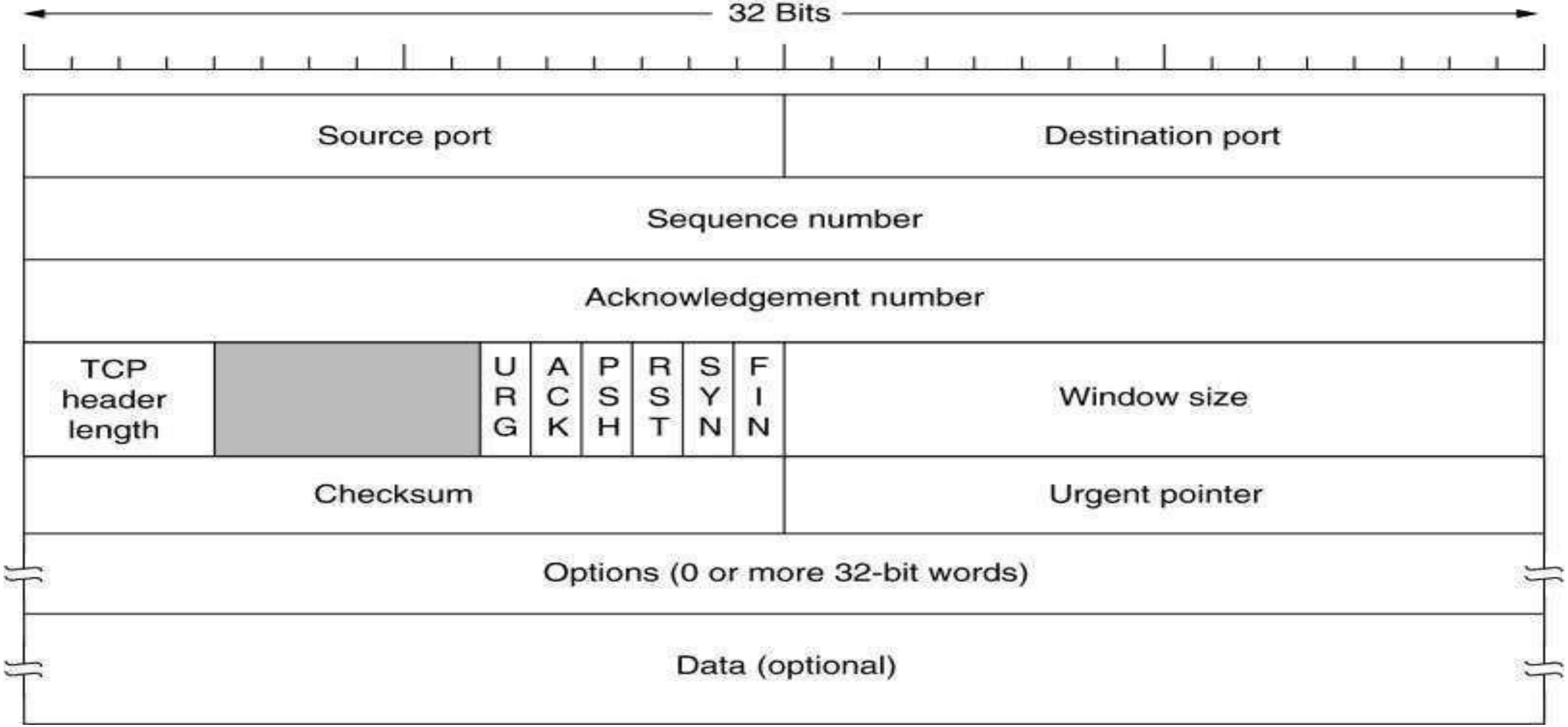
The TCP Service Model (2)



(a) Four 512-byte segments sent as separate IP datagrams.

(b) The 2048 bytes of data delivered to the application in a single READ CALL.

The TCP Segment Header



32 Bits

Source port | Destination port

Sequence number

Acknowledgement number

TCP header length | URG | ACK | PSH | RST | SYN | FIN | Window size

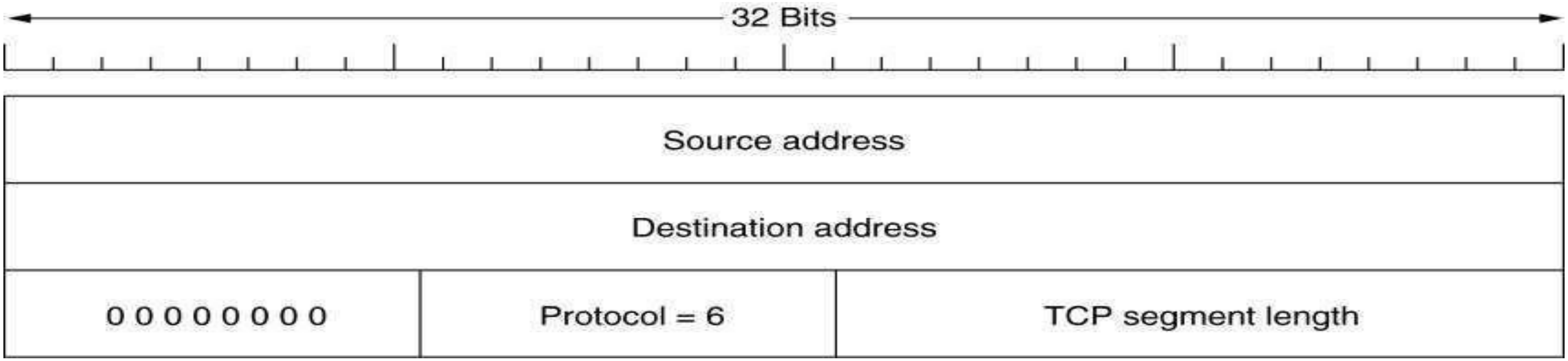
Checksum | Urgent pointer

Options (0 or more 32-bit words)

Data (optional)

TCP Header.

The TCP Segment Header (2)



32 Bits

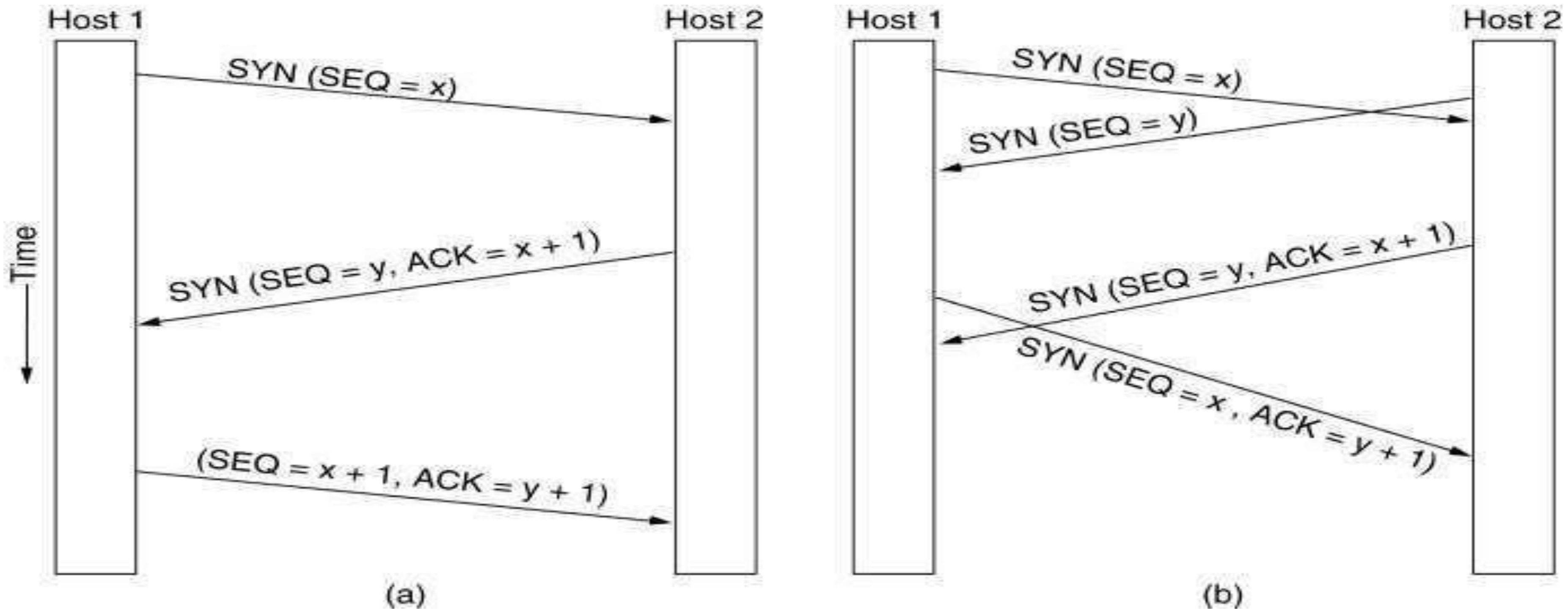
Source address

Destination address

0 0 0 0 0 0 0 0 | Protocol = 6 | TCP segment length

The pseudoheader included in the TCP checksum.

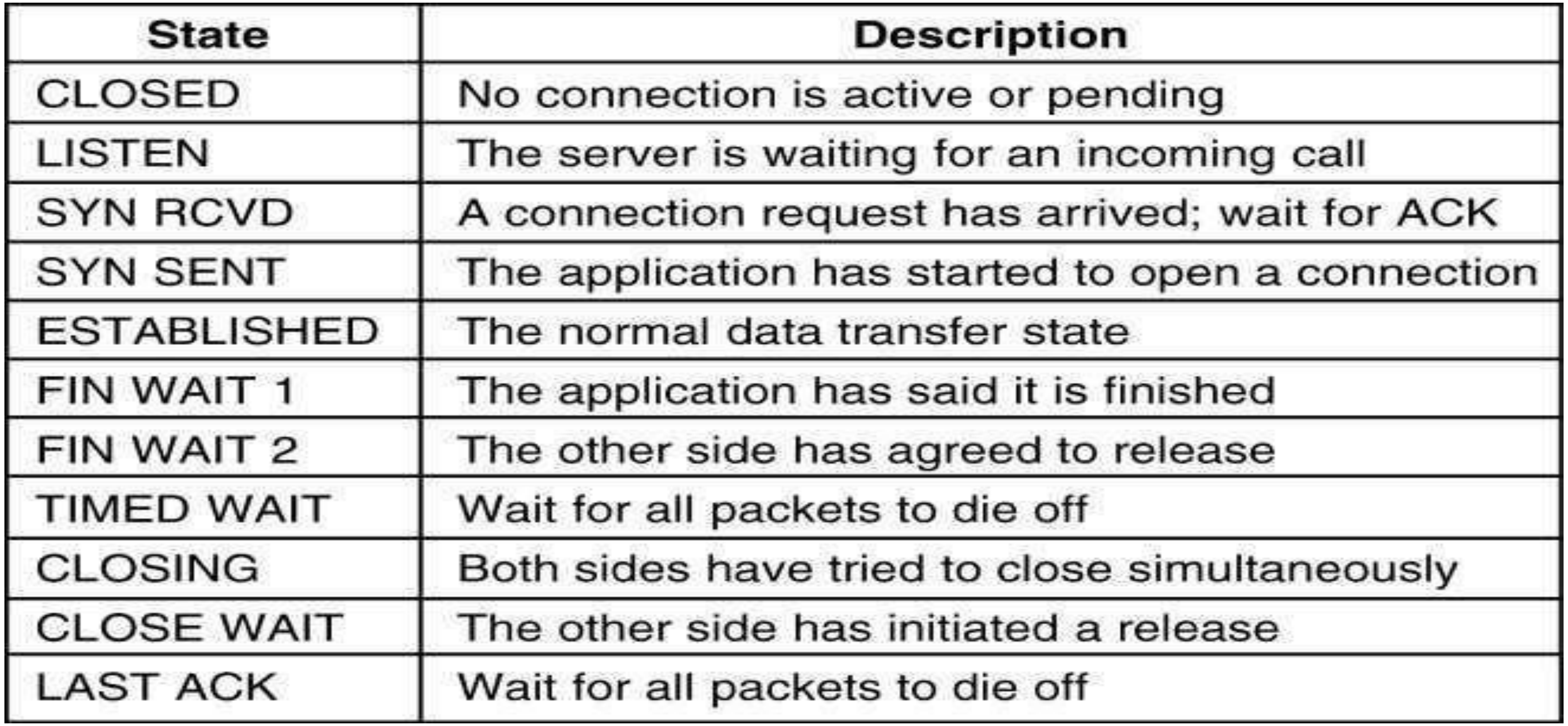
TCP Connection Establishment


Host 1
Host 2
SYN (SEQ = x)
SYN (SEQ = y, ACK = x + 1)
(SEQ = x + 1, ACK = y + 1)
Time
(a)
Host 1
Host 2
SYN (SEQ = x)
SYN (SEQ = y)
SYN (SEQ = y, ACK = x + 1)
SYN (SEQ = x , ACK = y + 1)
(b)

(a) TCP connection establishment in the normal case.

(b) Call collision.

TCP Connection Management Modeling

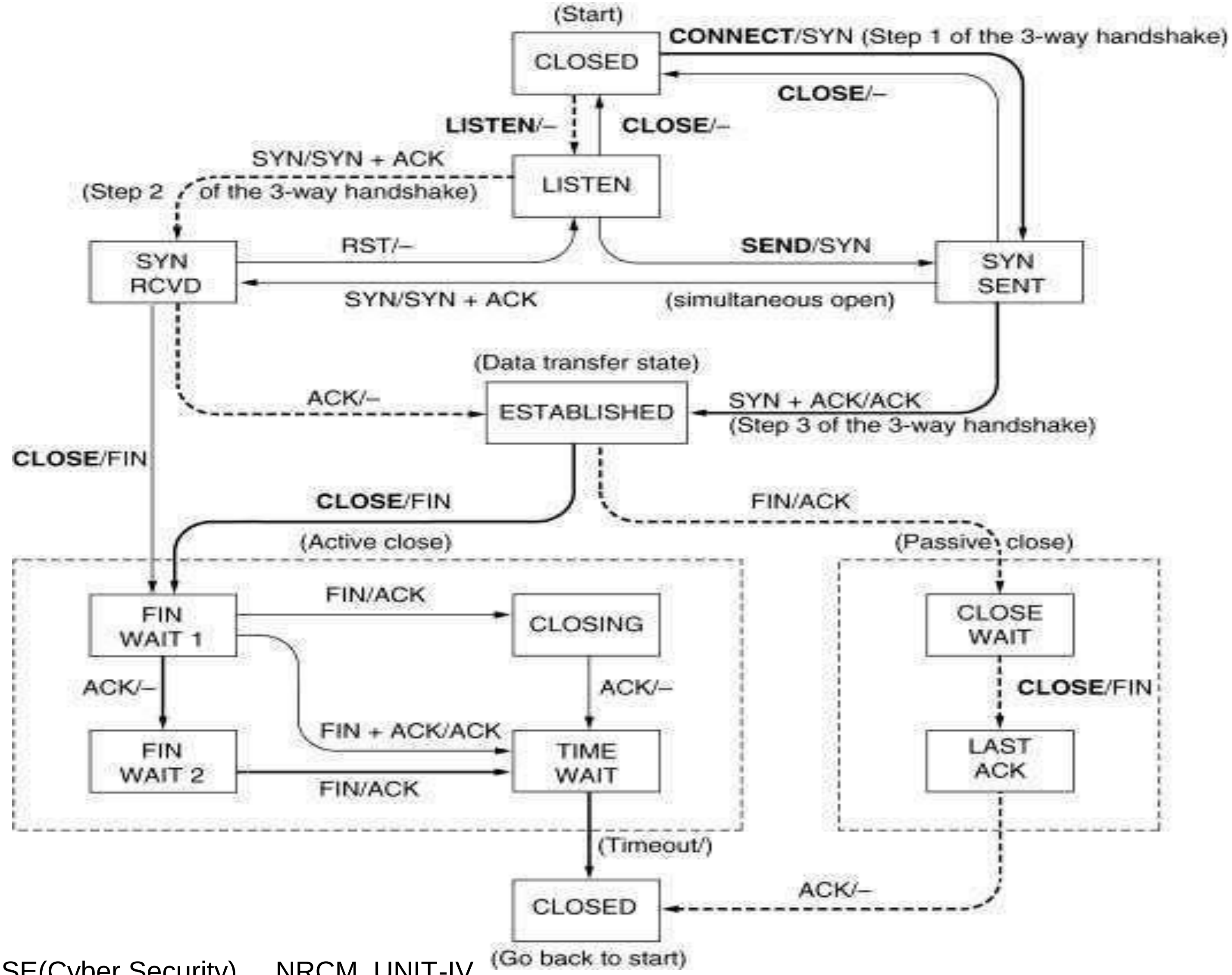


State	Description
CLOSED	No connection is active or pending
LISTEN	The server is waiting for an incoming call
SYN RCVD	A connection request has arrived; wait for ACK
SYN SENT	The application has started to open a connection
ESTABLISHED	The normal data transfer state
FIN WAIT 1	The application has said it is finished
FIN WAIT 2	The other side has agreed to release
TIMED WAIT	Wait for all packets to die off
CLOSING	Both sides have tried to close simultaneously
CLOSE WAIT	The other side has initiated a release
LAST ACK	Wait for all packets to die off

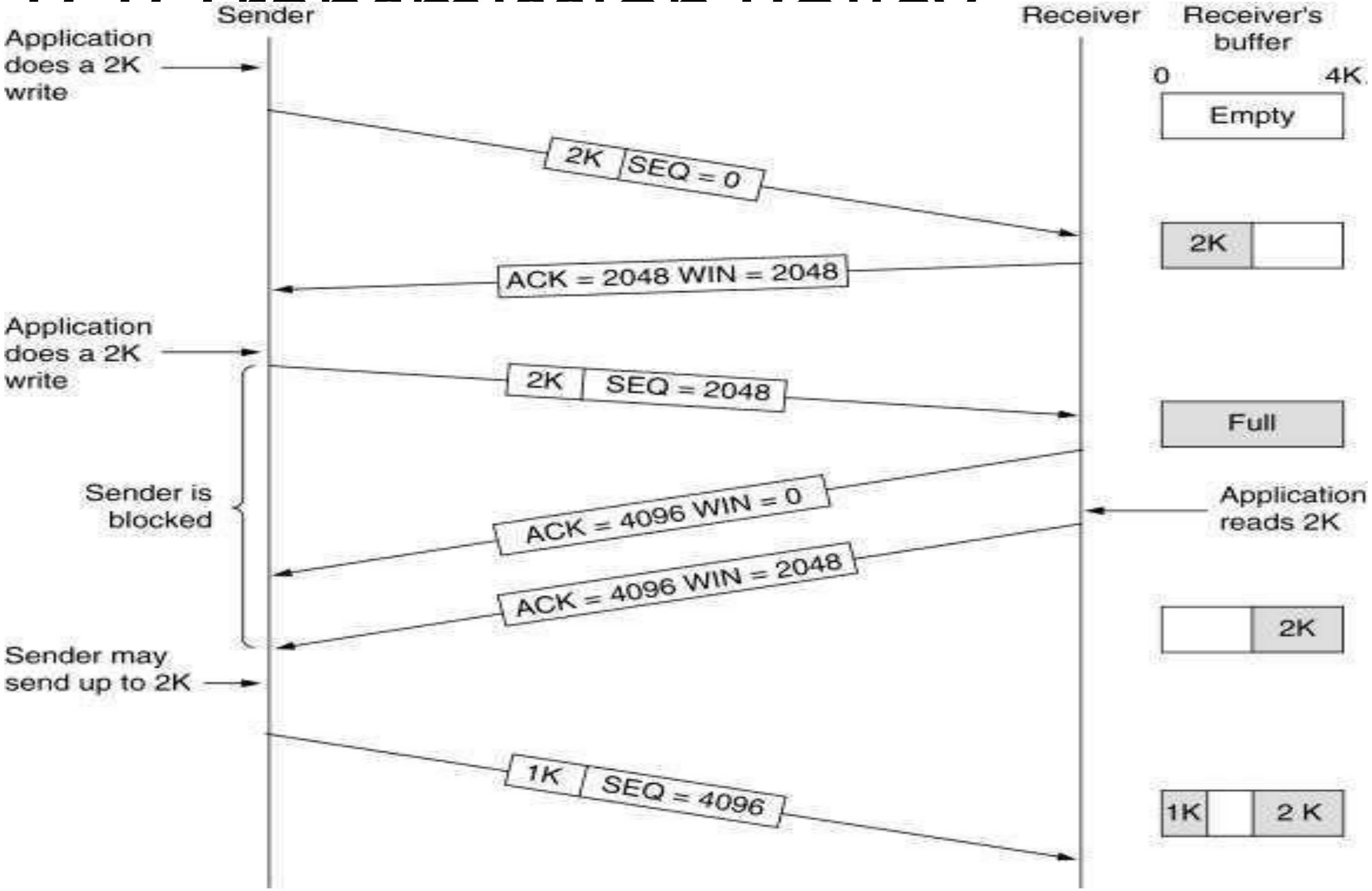
The states used in the TCP connection management finite state machine.

TCP Connection Management Modeling (2)

TCP connection management finite state machine. The heavy solid line is the normal path for a client.The heavy dashed line is the normal path for a server. The light lines are unusual events. Each transition is labeled by the event causing it and the action resulting from it, separated by a slash.

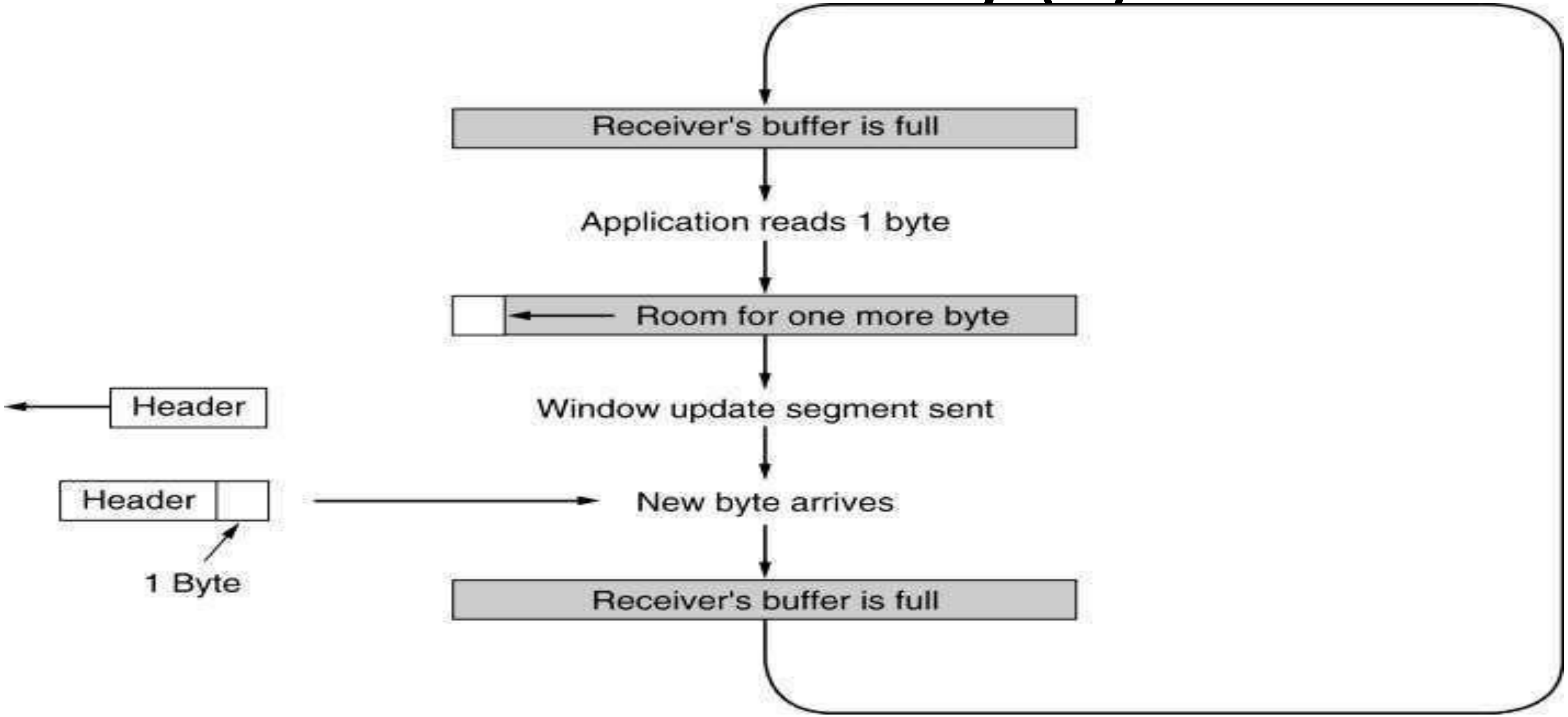


TCP Transmission Policy



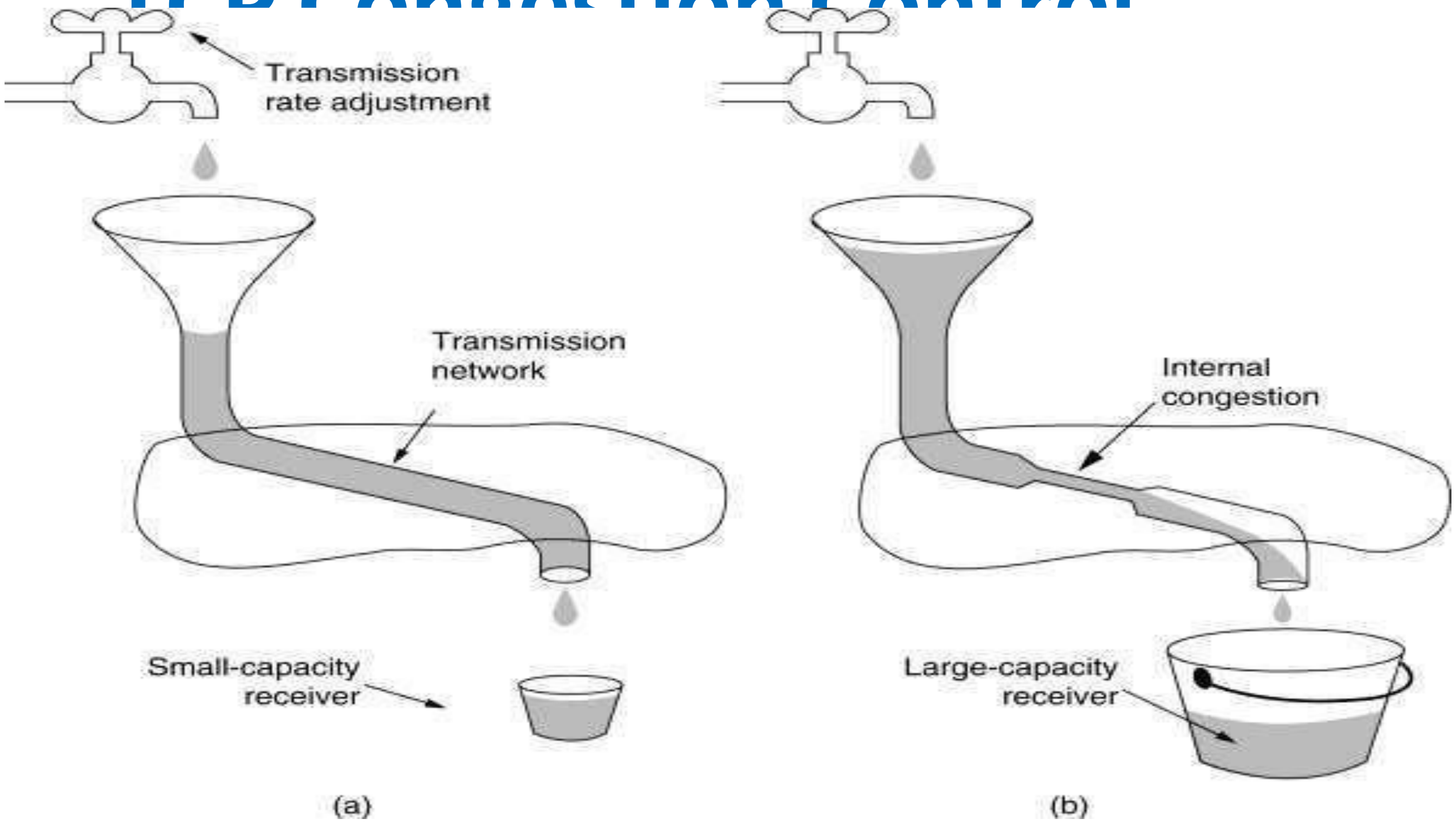
Window management in TCP.

TCP Transmission Policy (2)



Silly window syndrome.

TCP Congestion Control



(a) A fast network feeding a low capacity receiver.

(b) A slow network feeding a high-capacity receiver.

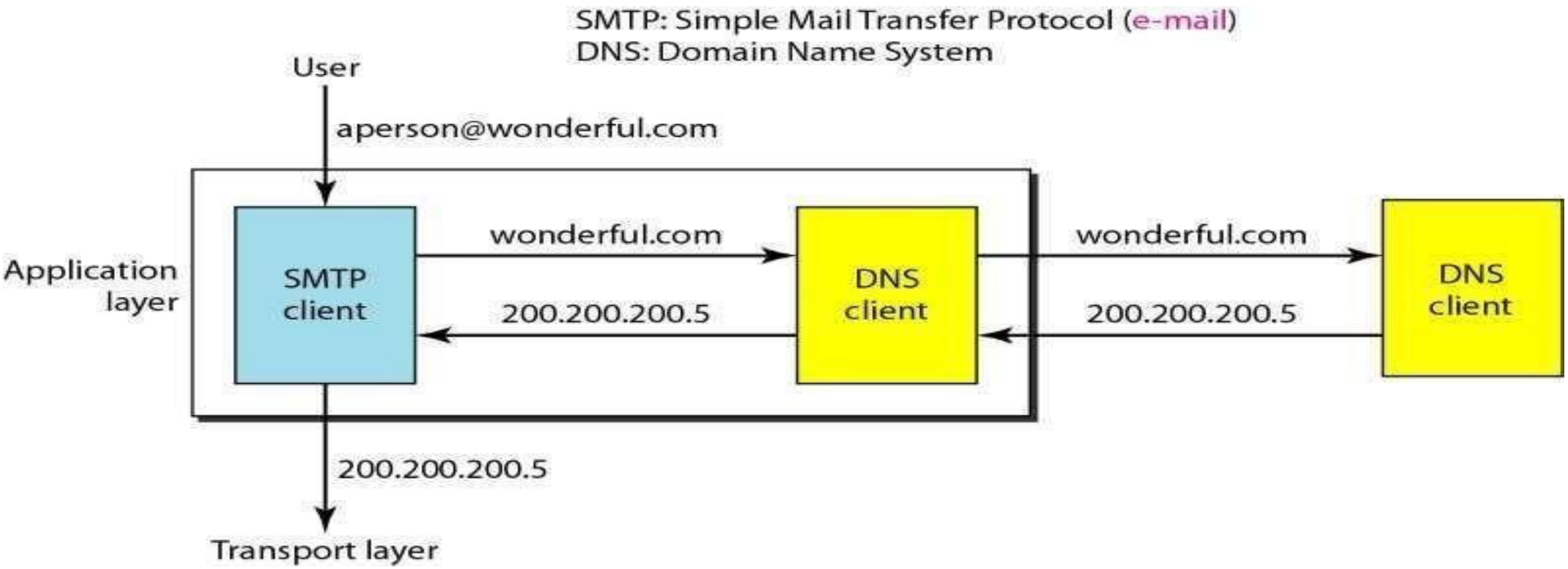
UNIT-V

The Application Layer

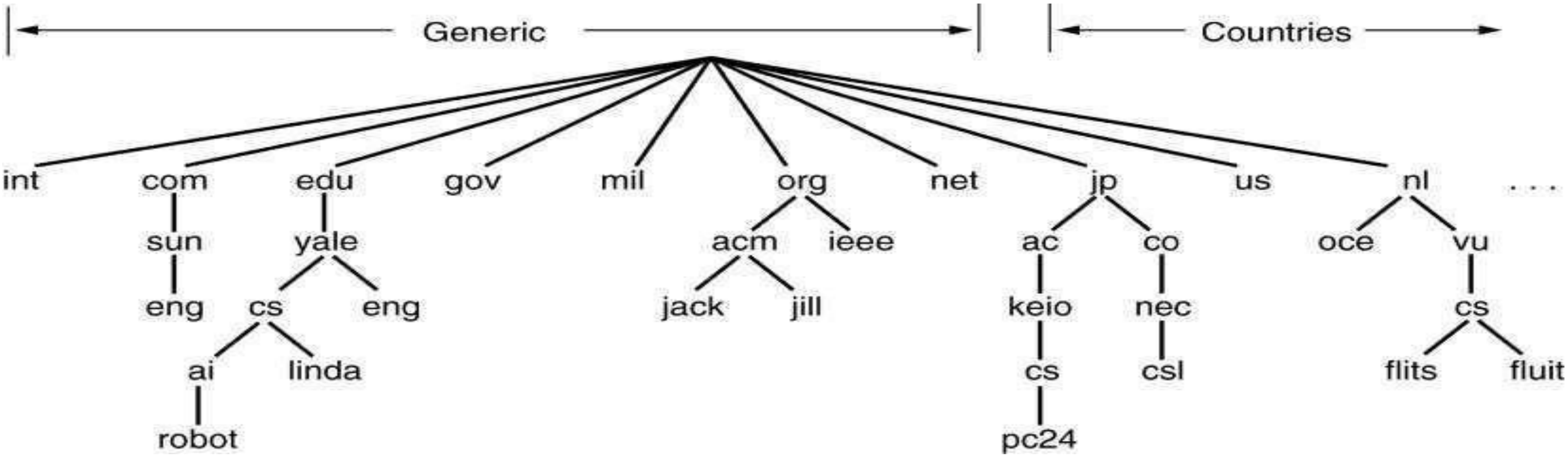
DNS – The Domain Name System

- The DNS Name Space

Figure 25.1 *Example of using the DNS service*

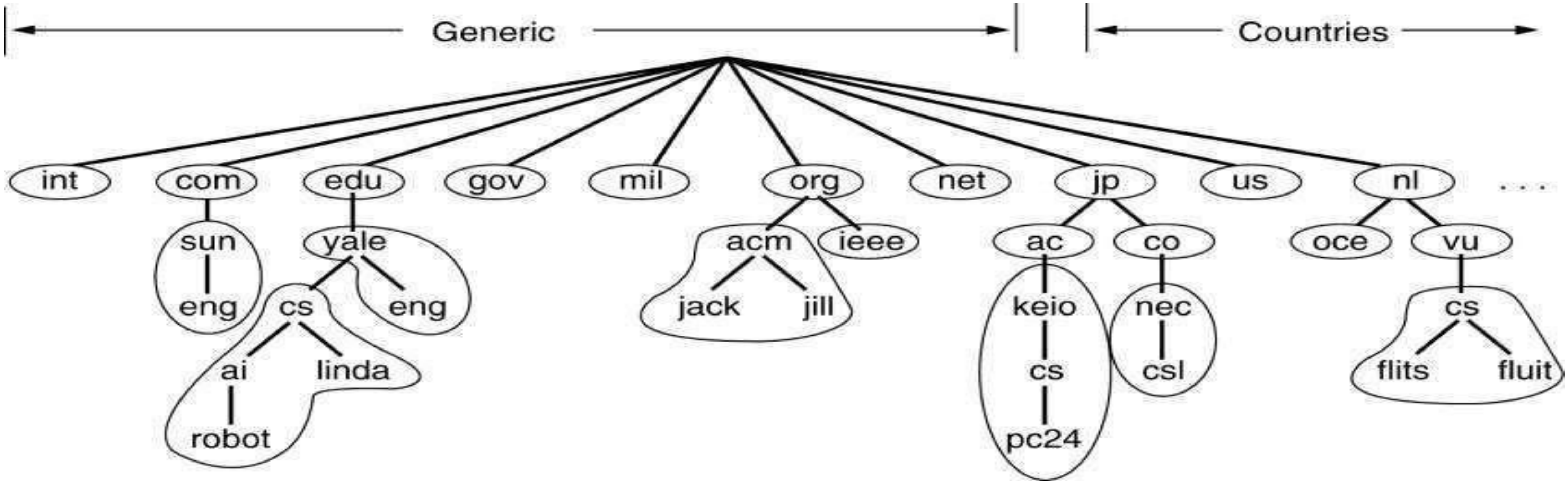


The DNS Name Space



A portion of the Internet domain name space.

Name Servers



Part of the DNS name space showing the division into zones.

Figure 25.3 *Domain names and labels*

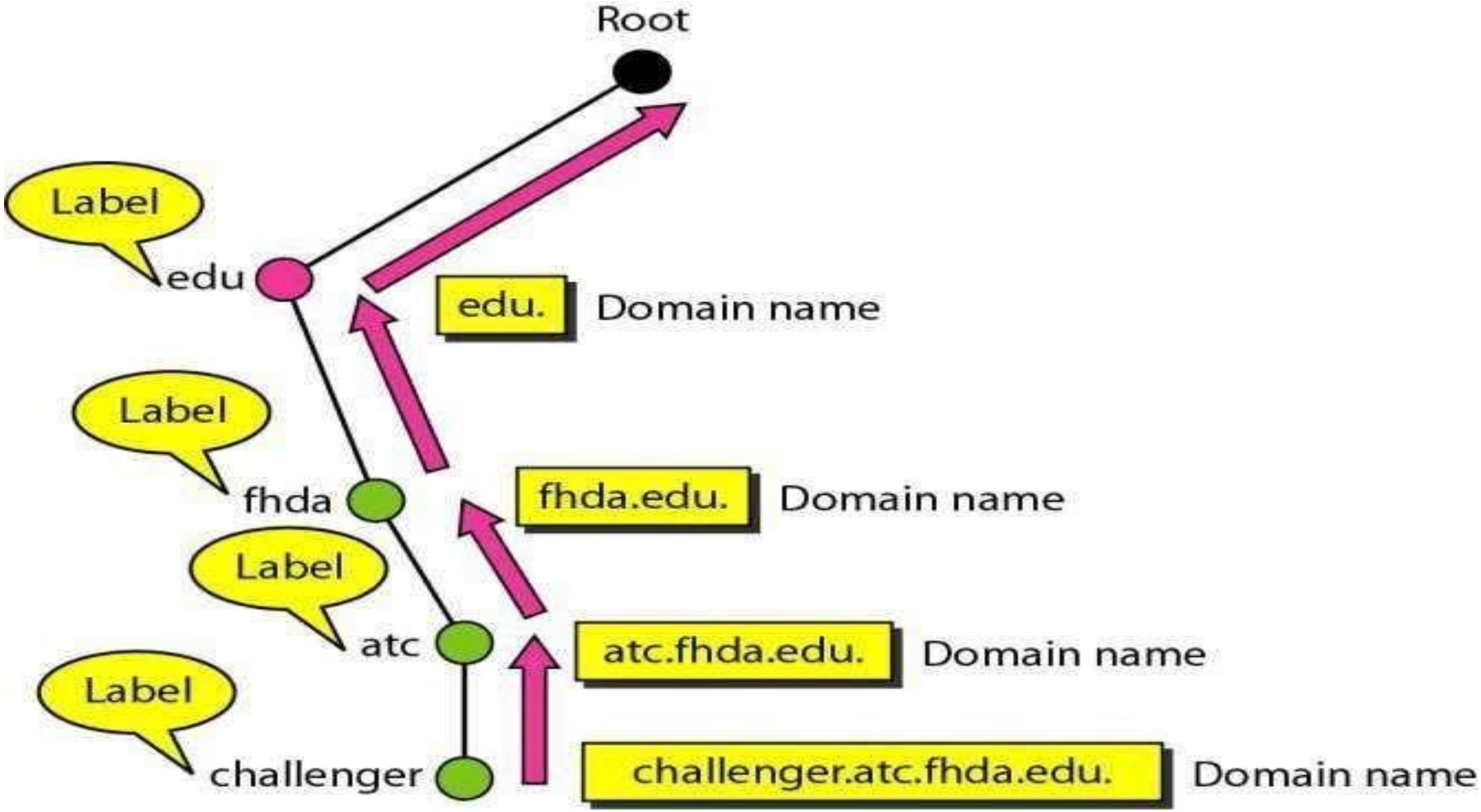
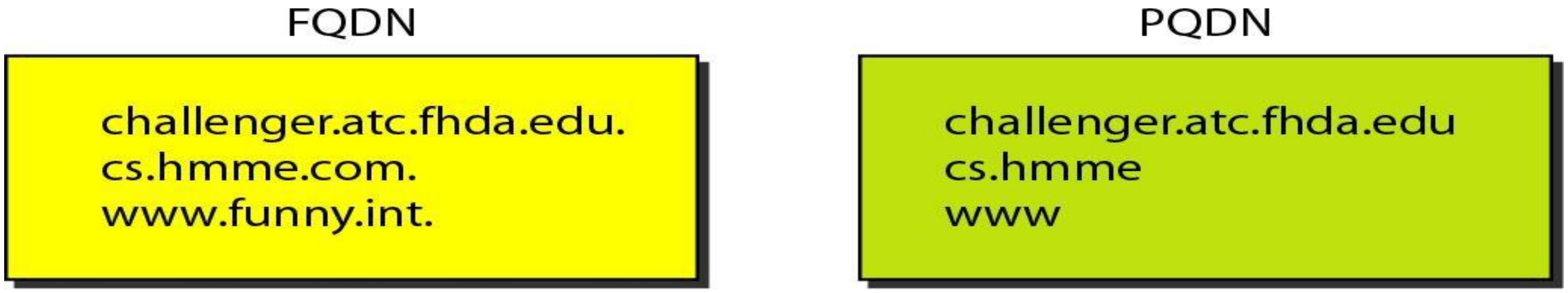


Figure 25.4 *FQDN and PQDN*



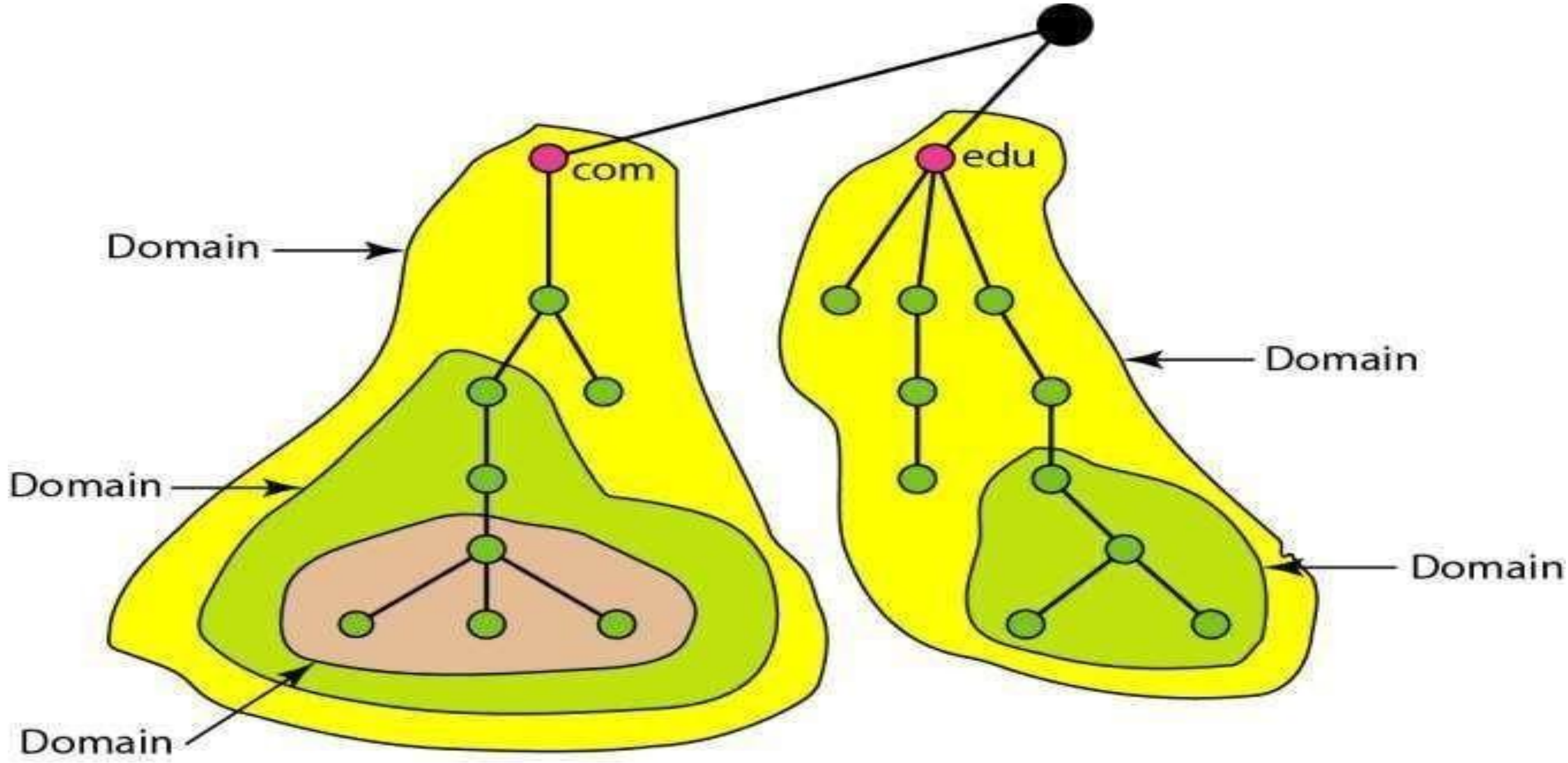
FQDN

challenger.atc.fhda.edu.
cs.hmme.com.
www.funny.int.

PQDN

challenger.atc.fhda.edu
cs.hmme
www

Figure 25.5 *Domains*



Electronic Mail

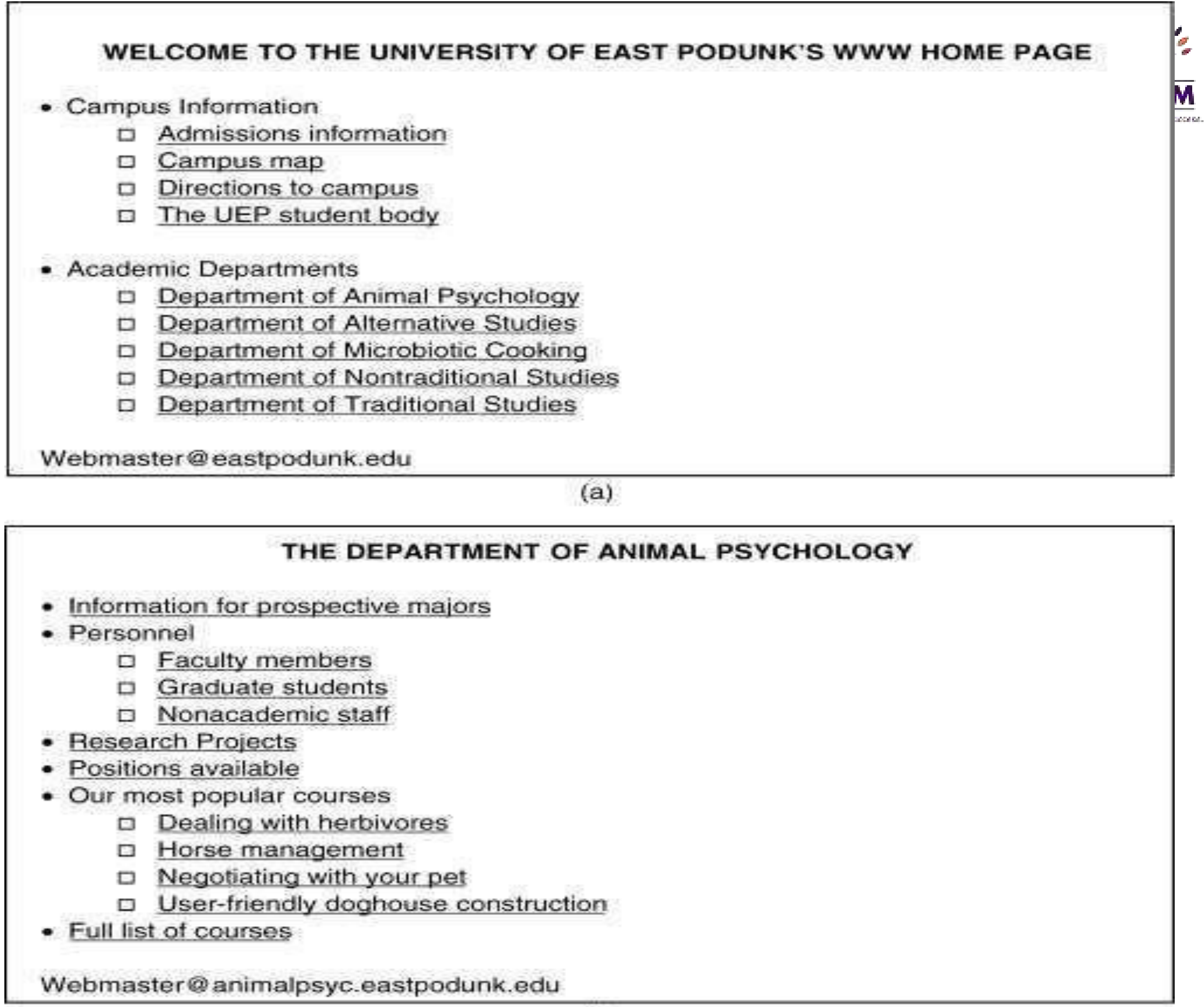
- Architecture and Services
- The User Agent
- Message Formats
- Message Transfer
- Final Delivery

SMTP
POP3
IMAP

The World Wide Web

- Architectural Overview
- Static Web Documents
- Dynamic Web Documents
- HTTP – The HyperText Transfer Protocol

Architectural Overview



WELCOME TO THE UNIVERSITY OF EAST PODUNK'S WWW HOME PAGE

- Campus Information
  - □ Admissions information
  - □ Campus map
  - □ Directions to campus
  - □ The UEP student body
- Academic Departments
  - □ Department of Animal Psychology
  - □ Department of Alternative Studies
  - □ Department of Microbiotic Cooking
  - □ Department of Nontraditional Studies
  - □ Department of Traditional Studies

Webmaster@eastpodunk.edu

(a)

THE DEPARTMENT OF ANIMAL PSYCHOLOGY

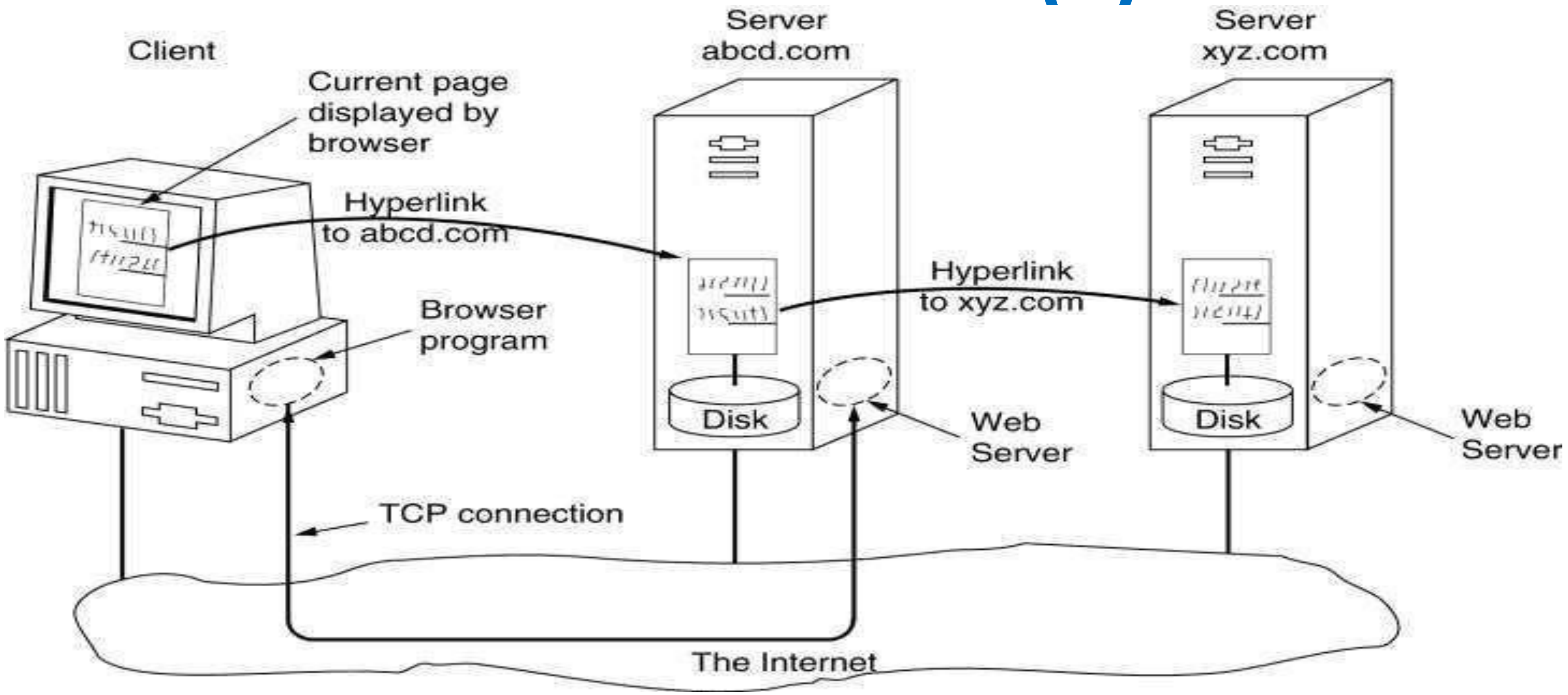
- Information for prospective majors
- Personnel
  - □ Faculty members
  - □ Graduate students
  - □ Nonacademic staff
- Research Projects
- Positions available
- Our most popular courses
  - □ Dealing with herbivores
  - □ Horse management
  - □ Negotiating with your pet
  - □ User-friendly doghouse construction
- Full list of courses

Webmaster@animalpsyc.eastpodunk.edu

(b)

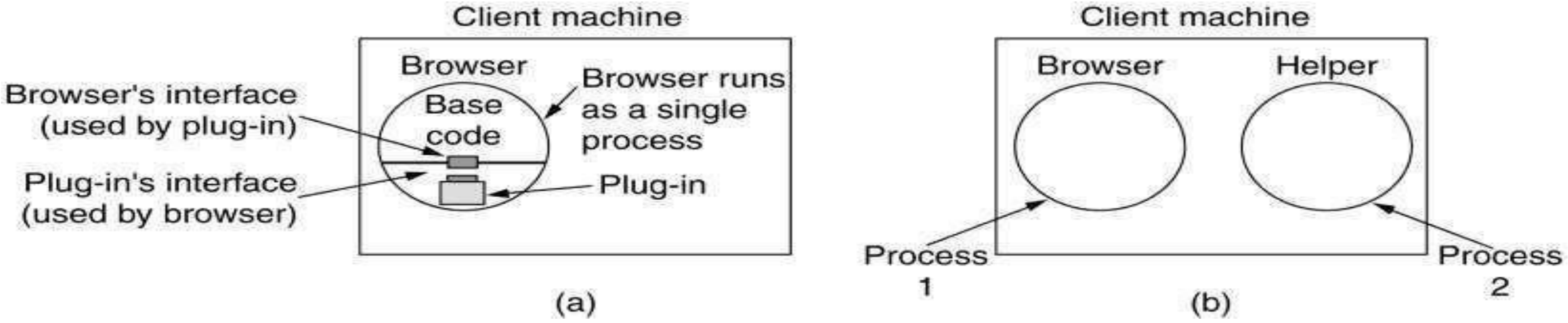
(a) A Web page (b) The page reached by clicking on Department of Animal Psychology.

Architectural Overview (2)



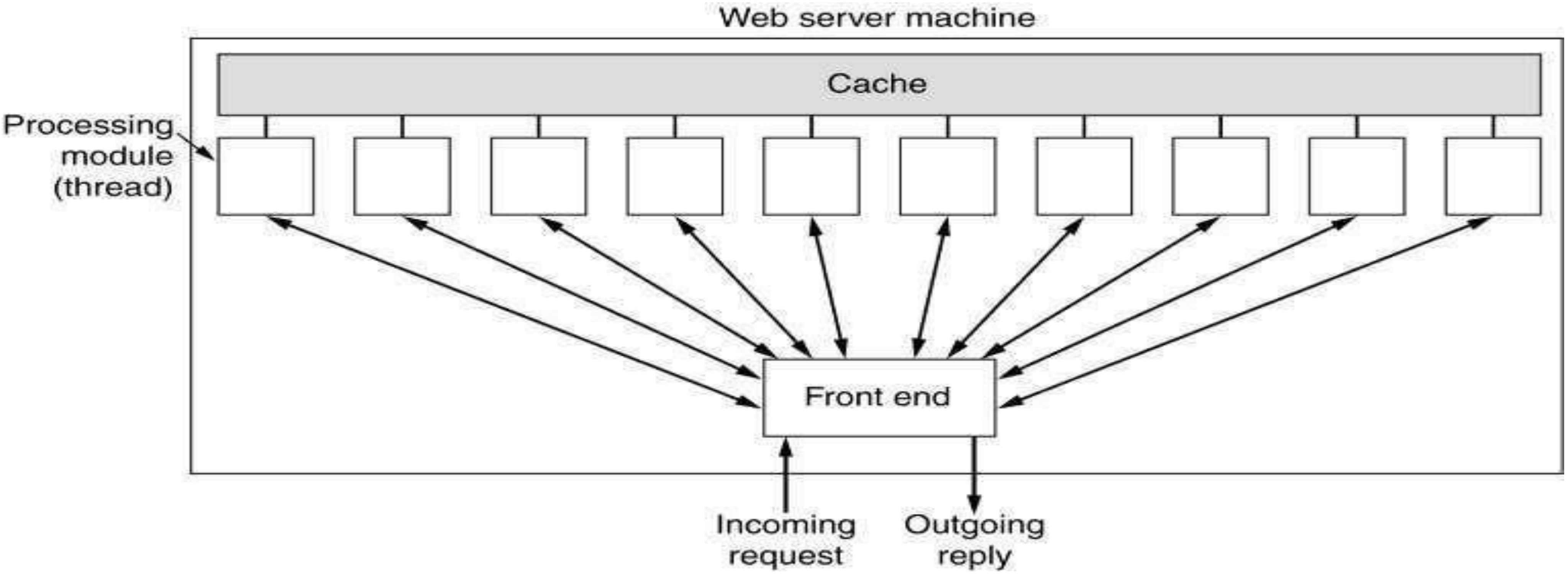
The parts of the Web model.

The Client Side



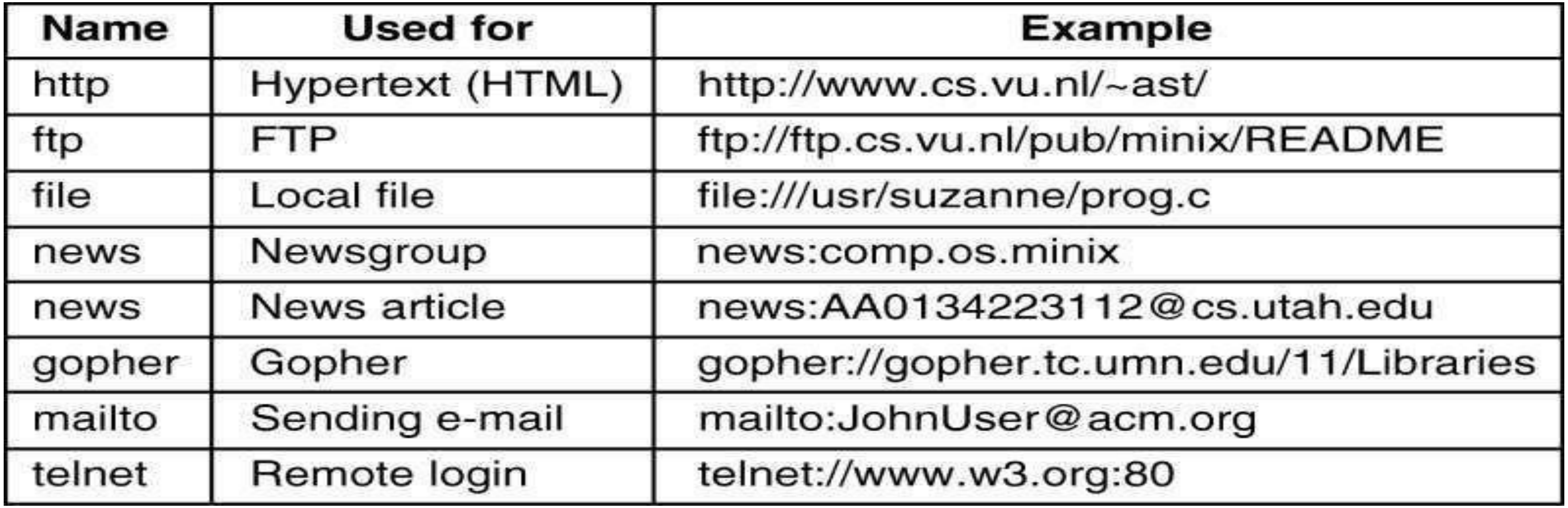
(a) A browser plug-in. (b) A helper application.

The Server Side



A multithreaded Web server with a front end and processing modules.

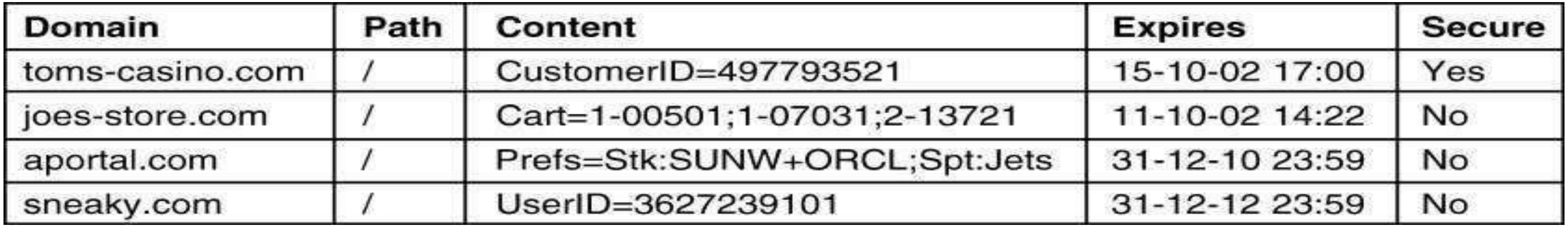
URLs – Uniform Resource Locaters



Name	Used for	Example
http	Hypertext (HTML)	http://www.cs.vu.nl/~ast/
ftp	FTP	ftp://ftp.cs.vu.nl/pub/minix/README
file	Local file	file:///usr/suzanne/prog.c
news	Newsgroup	news:comp.os.minix
news	News article	news:AA0134223112@cs.utah.edu
gopher	Gopher	gopher://gopher.tc.umn.edu/11/Libraries
mailto	Sending e-mail	mailto:JohnUser@acm.org
telnet	Remote login	telnet://www.w3.org:80

Some common URLs.

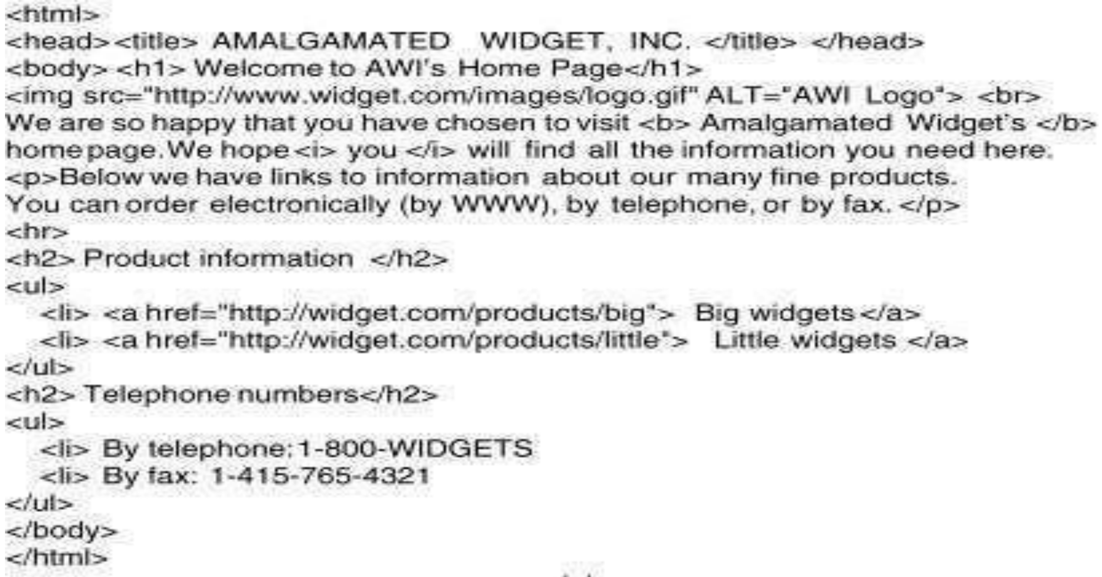
Statelessness and Cookies



Domain	Path	Content	Expires	Secure
toms-casino.com	/	CustomerID=497793521	15-10-02 17:00	Yes
joes-store.com	/	Cart=1-00501;1-07031;2-13721	11-10-02 14:22	No
aportal.com	/	Prefs=Stk:SUNW+ORCL;Spt:Jets	31-12-10 23:59	No
sneaky.com	/	UserID=3627239101	31-12-12 23:59	No

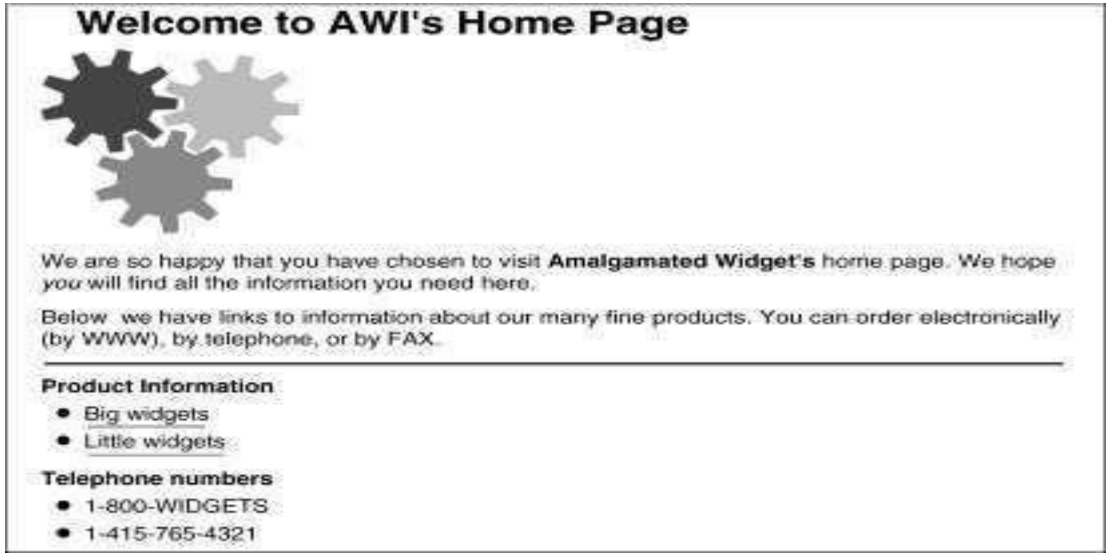
Some examples of cookies.

HTML – HyperText Markup Language



```
<html>
<head><title> AMALGAMATED  WIDGET, INC. </title> </head>
<body> <h1> Welcome to AWI's Home Page</h1>
<img src="http://www.widget.com/images/logo.gif" ALT="AWI  Logo"> <br>
We are so happy that you have chosen to visit <b> Amalgamated Widget's </b>
home page. We hope <i> you </i> will find all the information you need here.
<p>Below we have links to information about our many fine products.
You can order electronically (by WWW), by telephone, or by fax. </p>
<hr>
<h2> Product information  </h2>
<ul>
   <li> <a href="http://widget.com/products/big">  Big widgets </a>
   <li> <a href="http://widget.com/products/little">   Little widgets </a>
</ul>
<h2> Telephone numbers</h2>
<ul>
   <li> By telephone: 1-800-WIDGETS
   <li> By fax: 1-415-765-4321
</ul>
</body>
</html>
```

(a)



Welcome to AWI's Home Page

We are so happy that you have chosen to visit **Amalgamated Widget's** home page. We hope *you* will find all the information you need here.

Below we have links to information about our many fine products. You can order electronically (by WWW), by telephone, or by FAX.

Product Information

- Big widgets
- Little widgets

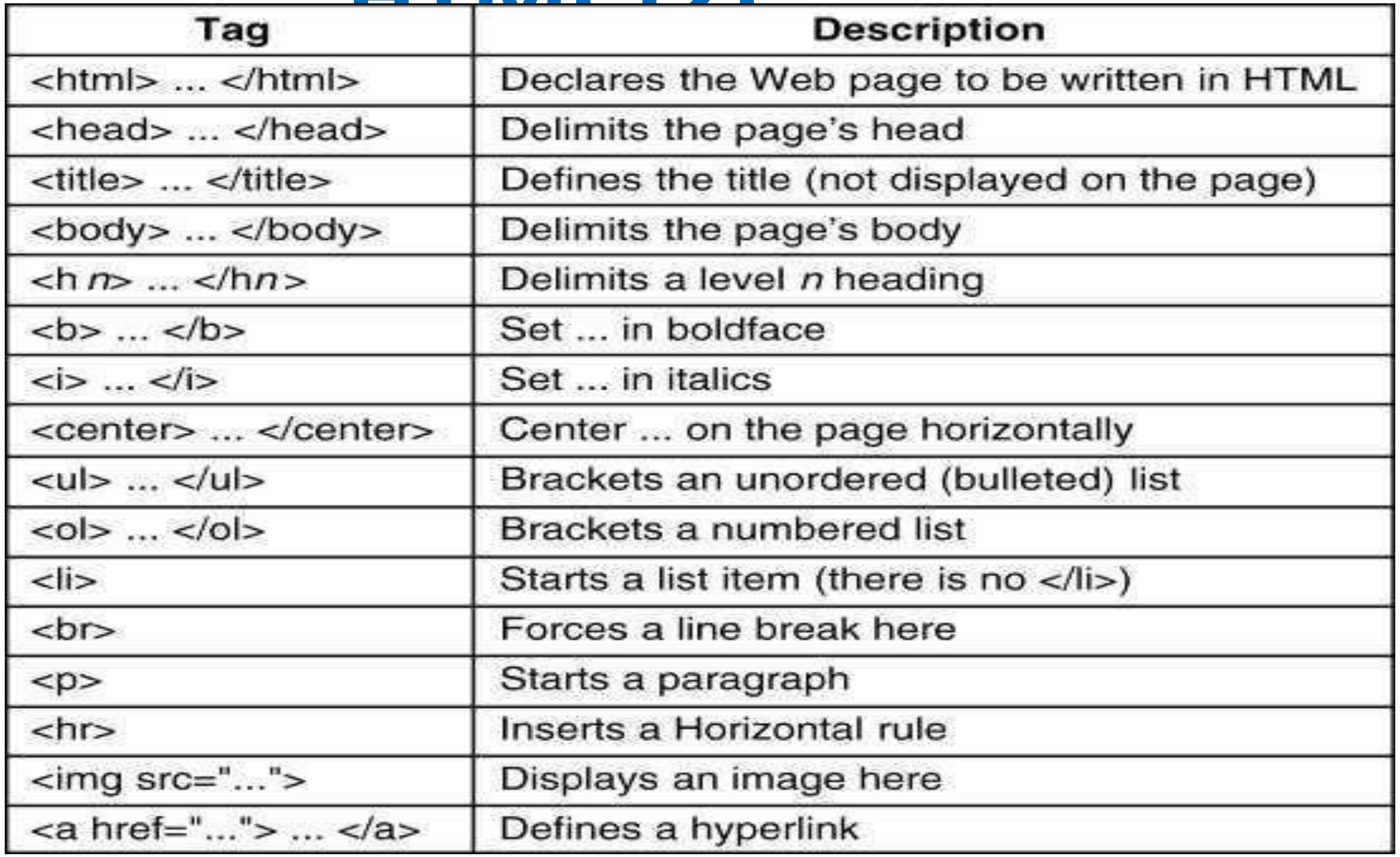
Telephone numbers

- 1-800-WIDGETS
- 1-415-765-4321

(b)

(a) The HTML for a sample Web page. (b) The formatted page.

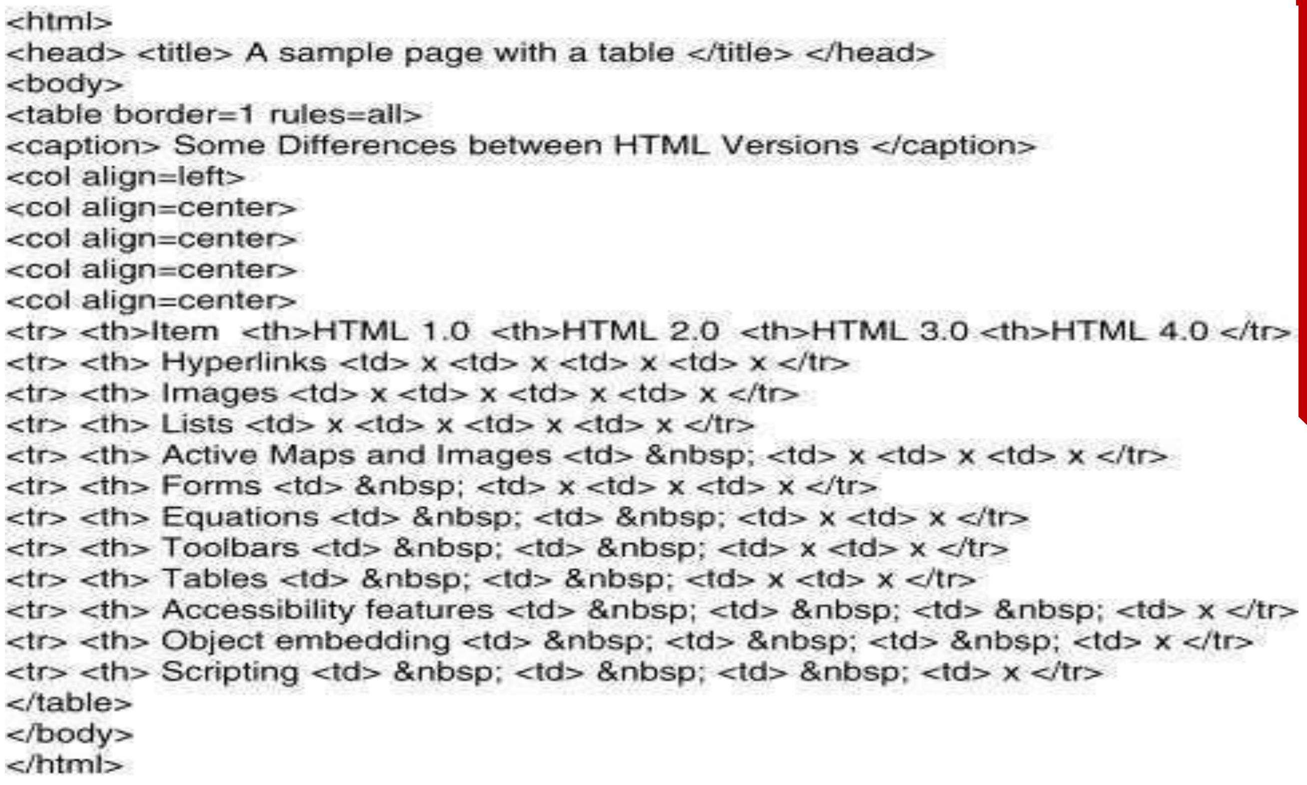
HTML (2)



Tag	Description
<html> ... </html>	Declares the Web page to be written in HTML
<head> ... </head>	Delimits the page's head
<title> ... </title>	Defines the title (not displayed on the page)
<body> ... </body>	Delimits the page's body
<h *n*> ... </h*n*>	Delimits a level *n* heading
<b> ... </b>	Set ... in boldface
<i> ... </i>	Set ... in italics
<center> ... </center>	Center ... on the page horizontally
<ul> ... </ul>	Brackets an unordered (bulleted) list
<ol> ... </ol>	Brackets a numbered list
<li>	Starts a list item (there is no </li>)
<br>	Forces a line break here
<p>	Starts a paragraph
<hr>	Inserts a Horizontal rule
<img src="...">	Displays an image here
<a href="..."> ... </a>	Defines a hyperlink

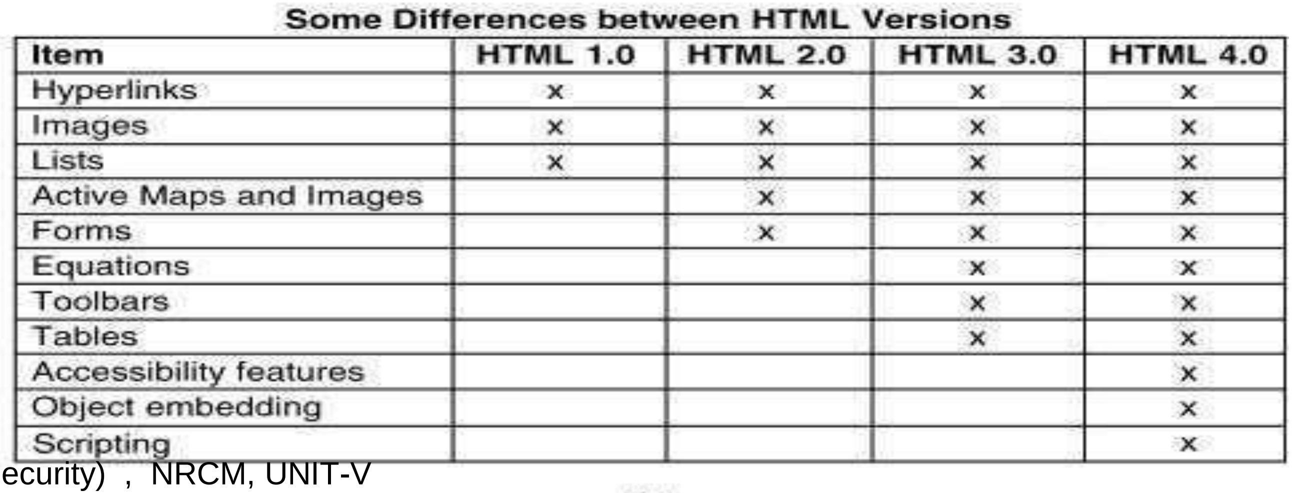
A selection of common HTML tags.
some can have additional parameters.

Forms



```
<html>
<head> <title> A sample page with a table </title> </head>
<body>
<table border=1 rules=all>
<caption> Some Differences between HTML Versions </caption>
<col align=left>
<col align=center>
<col align=center>
<col align=center>
<col align=center>
<tr> <th>Item  <th>HTML 1.0  <th>HTML 2.0  <th>HTML 3.0 <th>HTML 4.0 </tr>
<tr> <th> Hyperlinks <td> x <td> x <td> x <td> x </tr>
<tr> <th> Images <td> x <td> x <td> x <td> x </tr>
<tr> <th> Lists <td> x <td> x <td> x <td> x </tr>
<tr> <th> Active Maps and Images <td> &nbsp; <td> x <td> x <td> x </tr>
<tr> <th> Forms <td> &nbsp; <td> x <td> x <td> x </tr>
<tr> <th> Equations <td> &nbsp; <td> &nbsp; <td> x <td> x </tr>
<tr> <th> Toolbars <td> &nbsp; <td> &nbsp; <td> x <td> x </tr>
<tr> <th> Tables <td> &nbsp; <td> &nbsp; <td> x <td> x </tr>
<tr> <th> Accessibility features <td> &nbsp; <td> &nbsp; <td> &nbsp; <td> x </tr>
<tr> <th> Object embedding <td> &nbsp; <td> &nbsp; <td> &nbsp; <td> x </tr>
<tr> <th> Scripting <td> &nbsp; <td> &nbsp; <td> &nbsp; <td> x </tr>
</table>
</body>
</html>
```

(a)



Some Differences between HTML Versions

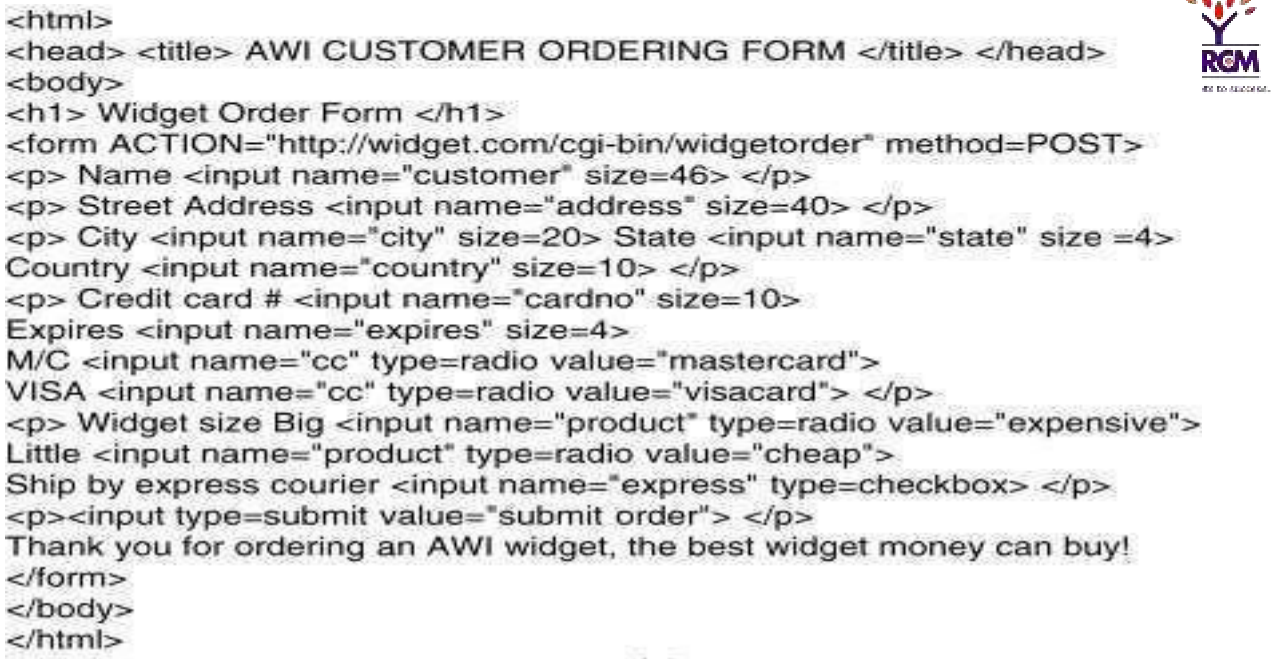
Item	HTML 1.0	HTML 2.0	HTML 3.0	HTML 4.0
Hyperlinks	x	x	x	x
Images	x	x	x	x
Lists	x	x	x	x
Active Maps and Images		x	x	x
Forms		x	x	x
Equations			x	x
Toolbars			x	x
Tables			x	x
Accessibility features				x
Object embedding				x
Scripting				x

(b)

(a) An HTML table.

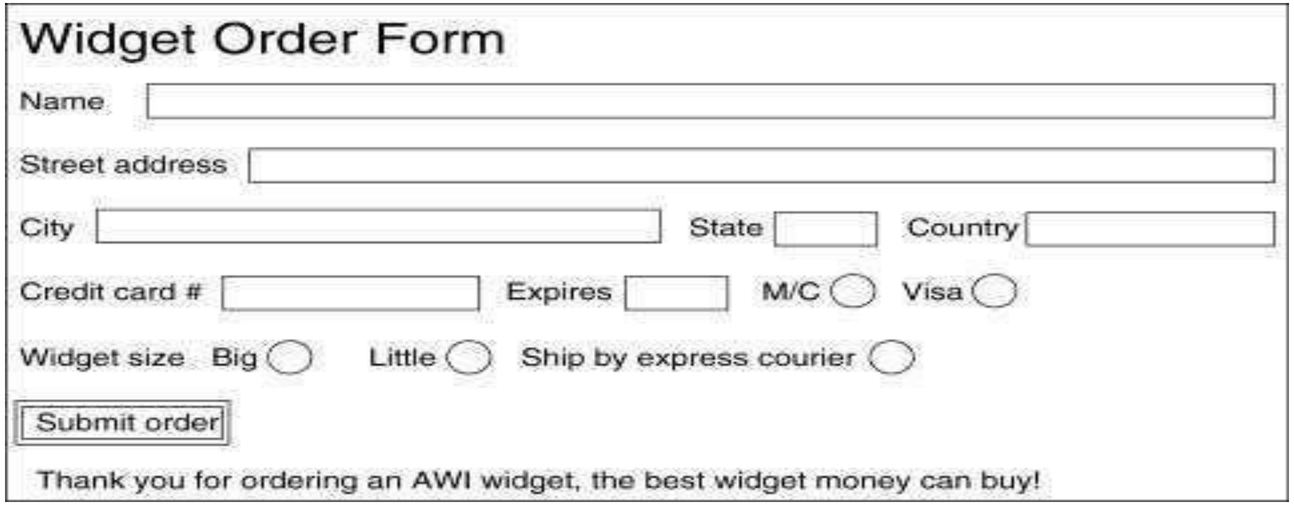
(b) A possible rendition of this table.

Forms (2)



```
<html>
<head> <title> AWI CUSTOMER ORDERING FORM </title> </head>
<body>
<h1> Widget Order Form </h1>
<form ACTION="http://widget.com/cgi-bin/widgetorder" method=POST>
<p> Name <input name="customer" size=46> </p>
<p> Street Address <input name="address" size=40> </p>
<p> City <input name="city" size=20> State <input name="state" size =4>
Country <input name="country" size=10> </p>
<p> Credit card # <input name="cardno" size=10>
Expires <input name="expires" size=4>
M/C <input name="cc" type=radio value="mastercard">
VISA <input name="cc" type=radio value="visacard"> </p>
<p> Widget size Big <input name="product" type=radio value="expensive">
Little <input name="product" type=radio value="cheap">
Ship by express courier <input name="express" type=checkbox> </p>
<p><input type=submit value="submit order"> </p>
Thank you for ordering an AWI widget, the best widget money can buy!
</form>
</body>
</html>
```

(a)



Widget Order Form

Name

Street address

City State Country

Credit card # Expires M/C ○ Visa ○

Widget size Big ○ Little ○ Ship by express courier ○

Submit order

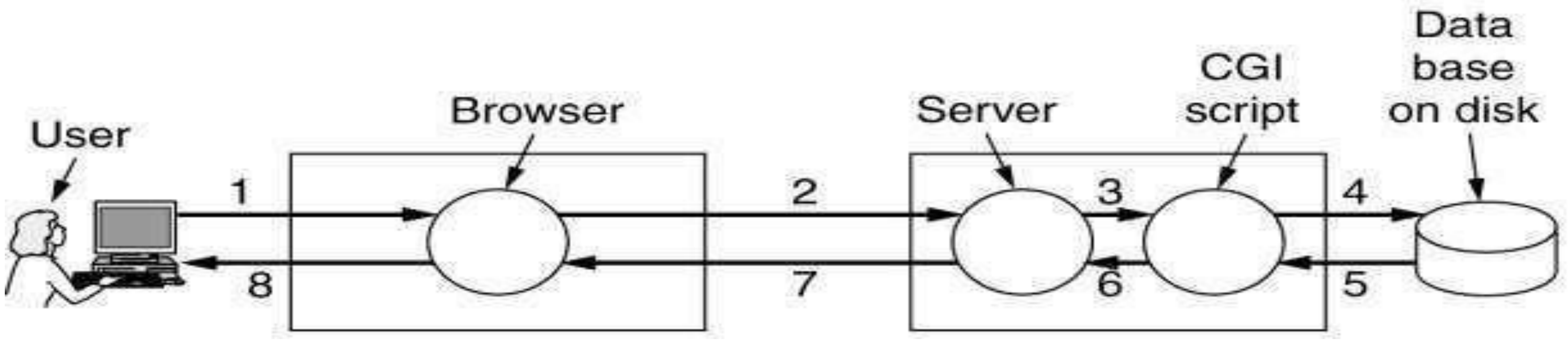
Thank you for ordering an AWI widget, the best widget money can buy!

(b)

(a) The HTML for an order form.

(b) The formatted page.

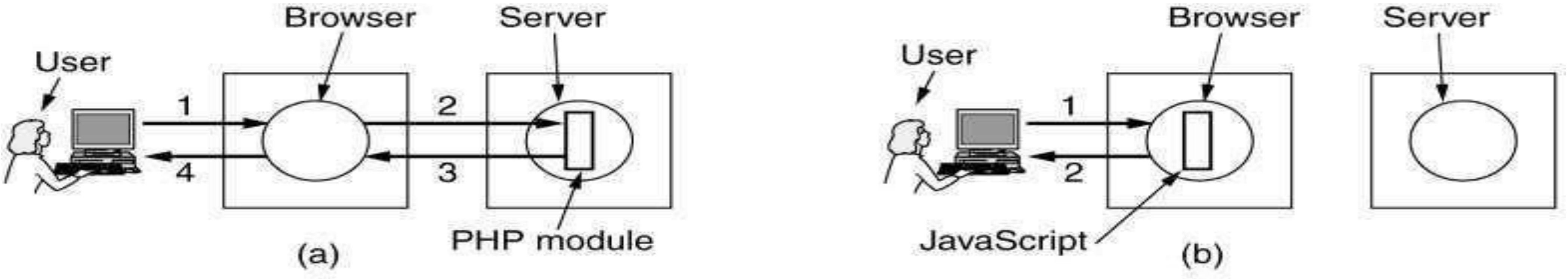
Dynamic Web Documents



1. User fills in form
2. Form sent back
3. Handed to CGI
4. CGI queries DB
5. Record found
6. CGI builds page
7. Page returned
8. Page displayed

Steps in processing the information from an HTML form.

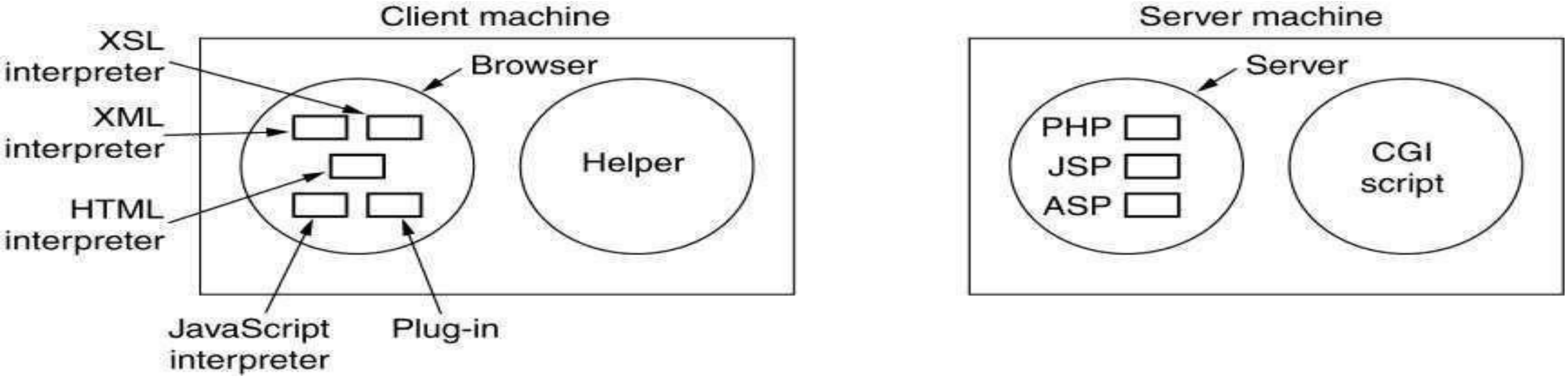
Client-Side Dynamic Web Page Generation (2)



(a) Server-side scripting with PHP.

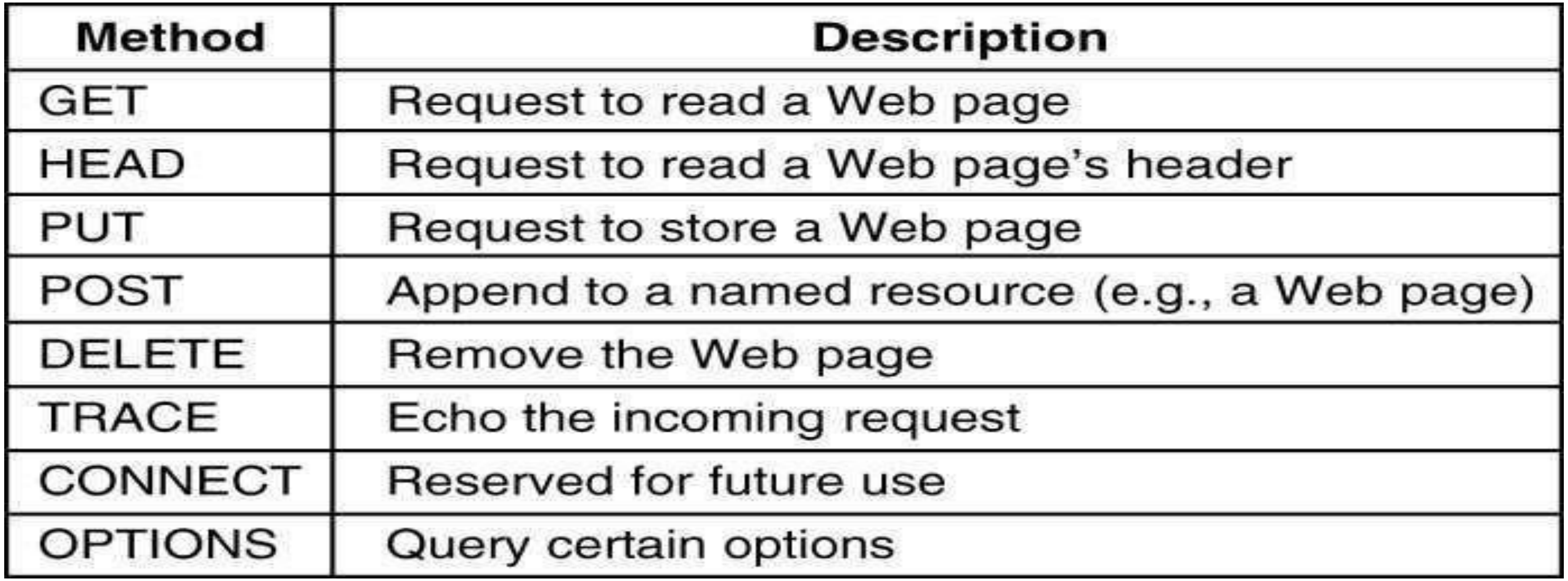
(b) Client-side scripting with JavaScript.

Client-Side Dynamic Web Page Generation (5)



The various ways to generate and display content.

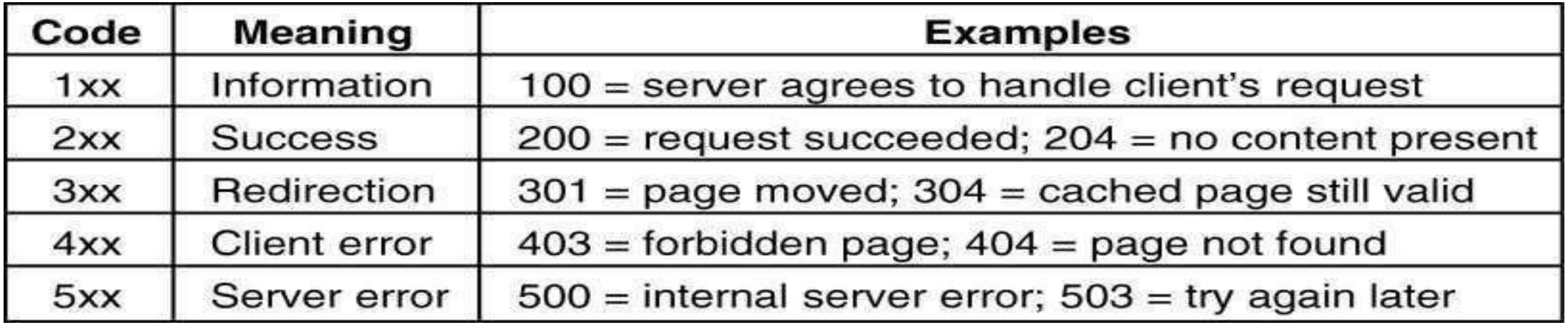
HTTP Methods



Method	Description
GET	Request to read a Web page
HEAD	Request to read a Web page's header
PUT	Request to store a Web page
POST	Append to a named resource (e.g., a Web page)
DELETE	Remove the Web page
TRACE	Echo the incoming request
CONNECT	Reserved for future use
OPTIONS	Query certain options

The built-in HTTP request methods.

HTTP Methods (2)



Code	Meaning	Examples
1xx	Information	100 = server agrees to handle client's request
2xx	Success	200 = request succeeded; 204 = no content present
3xx	Redirection	301 = page moved; 304 = cached page still valid
4xx	Client error	403 = forbidden page; 404 = page not found
5xx	Server error	500 = internal server error; 503 = try again later

The status code response groups.

26-1 REMOTE LOGGING(TELNET)

It would be impossible to write a specific client/server program for each demand. The better solution is a general-purpose client/server program that lets a user access any application program on a remote computer.

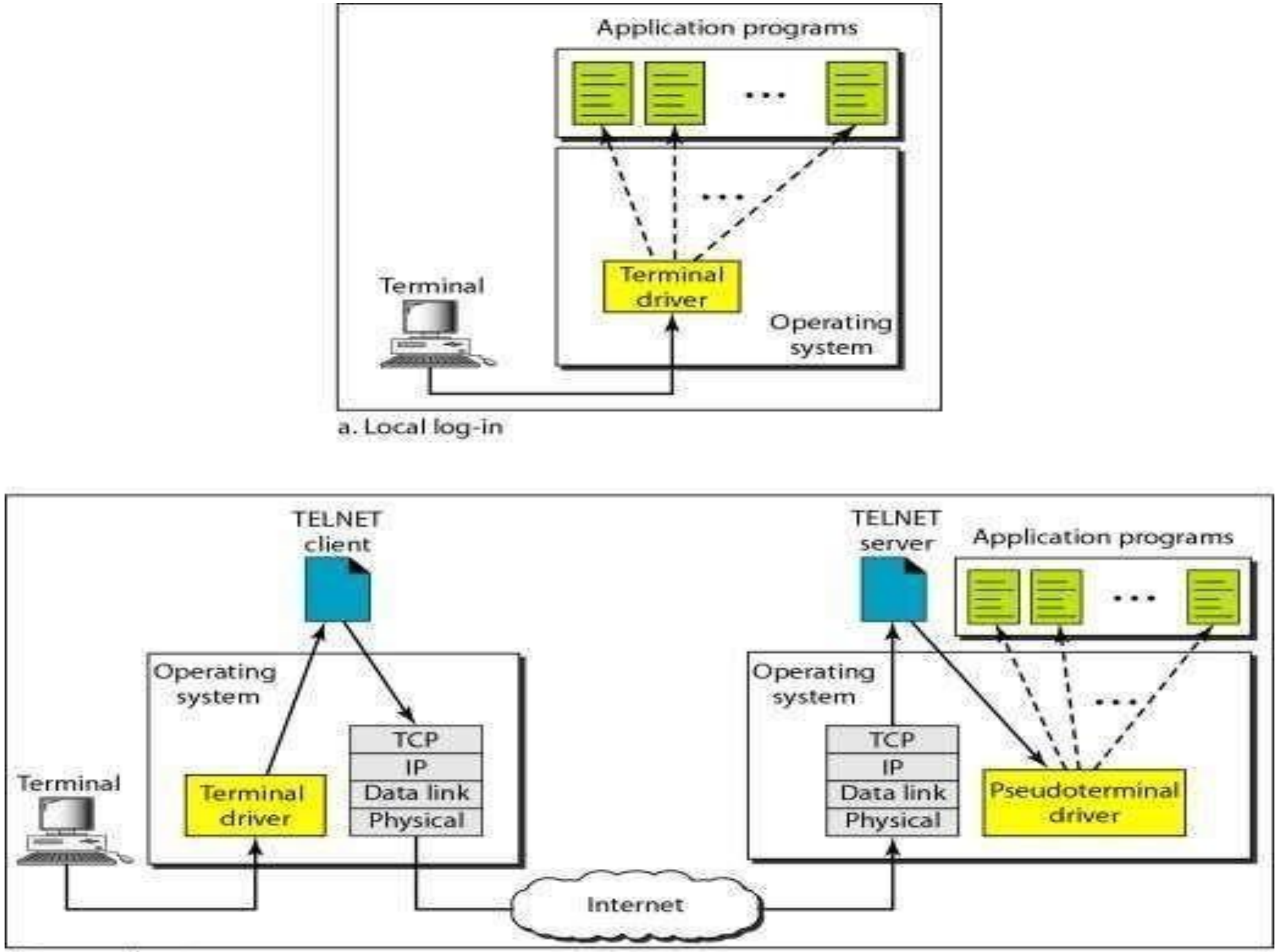
Topics discussed in this section:

TELNET

Note

TELNET is a general-purpose client/server application program.

Figure 26.1 *Local and remote log-in*



a. Local log-in

b. Remote log-in

Figure 26.2 *Concept of NVT*

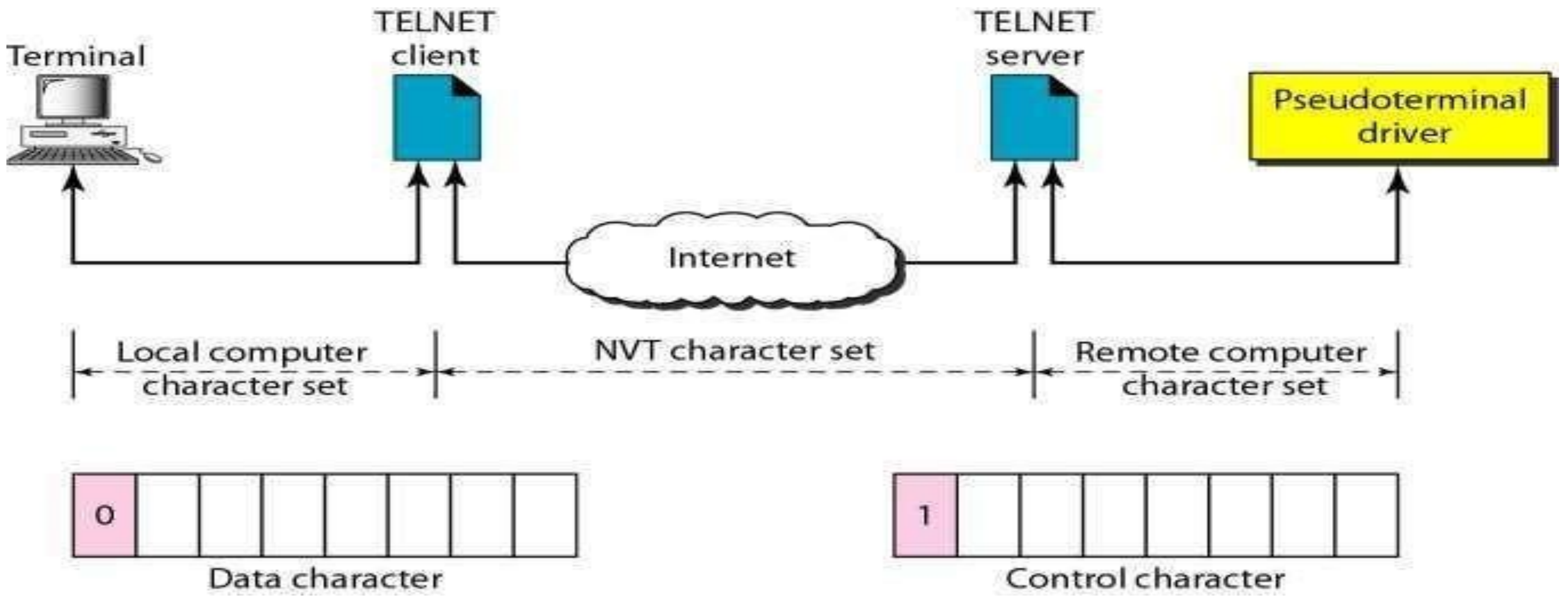
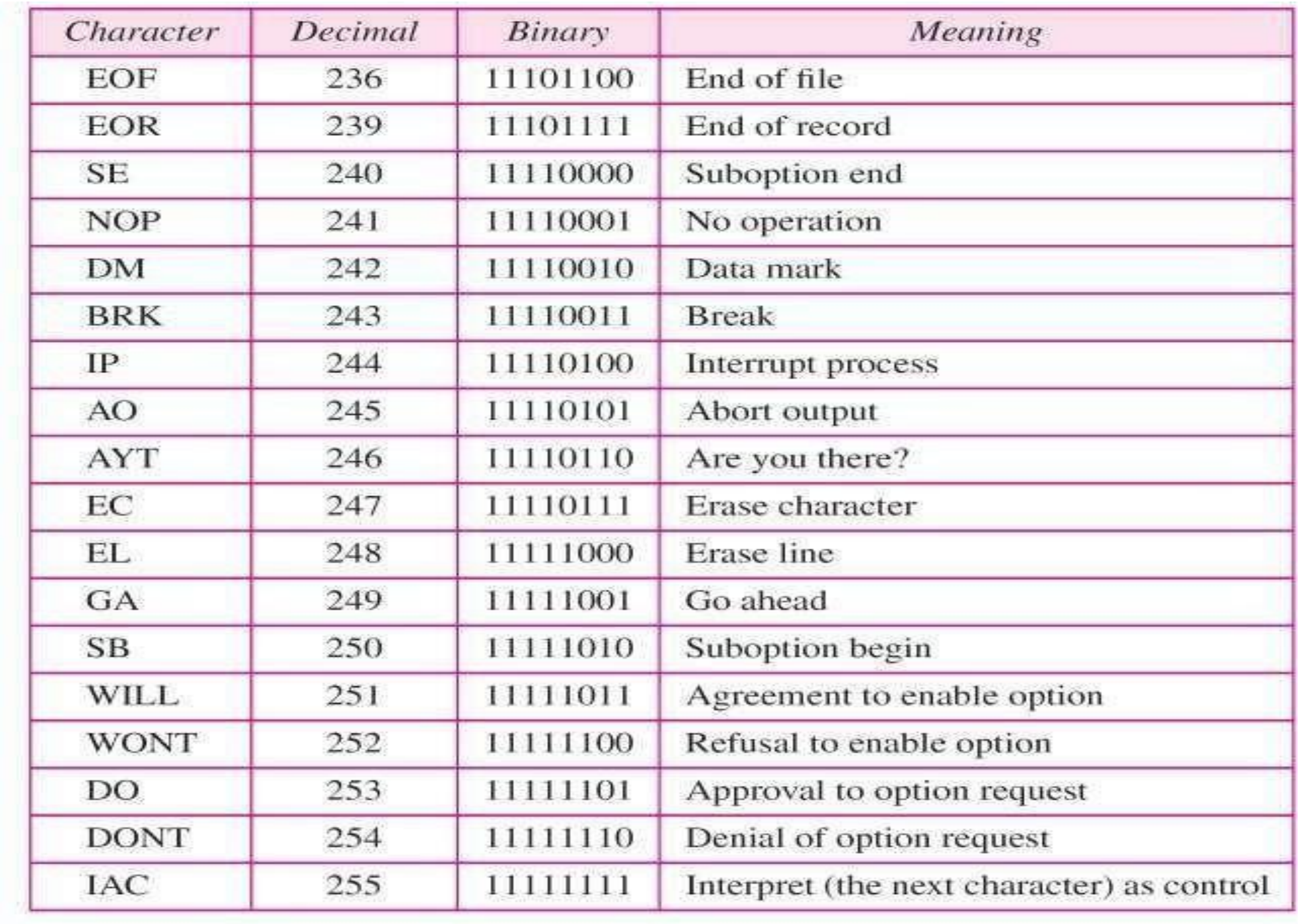
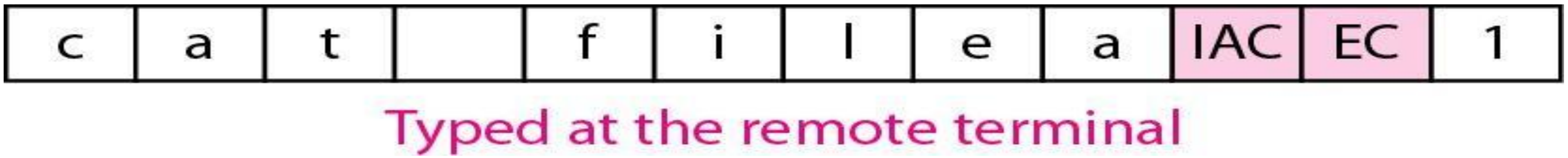


Table 26.1 *Some NVT control characters*



Character	*Decimal*	*Binary*	*Meaning*
EOF	236	11101100	End of file
EOR	239	11101111	End of record
SE	240	11110000	Suboption end
NOP	241	11110001	No operation
DM	242	11110010	Data mark
BRK	243	11110011	Break
IP	244	11110100	Interrupt process
AO	245	11110101	Abort output
AYT	246	11110110	Are you there?
EC	247	11110111	Erase character
EL	248	11111000	Erase line
GA	249	11111001	Go ahead
SB	250	11111010	Suboption begin
WILL	251	11111011	Agreement to enable option
WONT	252	11111100	Refusal to enable option
DO	253	11111101	Approval to option request
DONT	254	11111110	Denial of option request
IAC	255	11111111	Interpret (the next character) as control

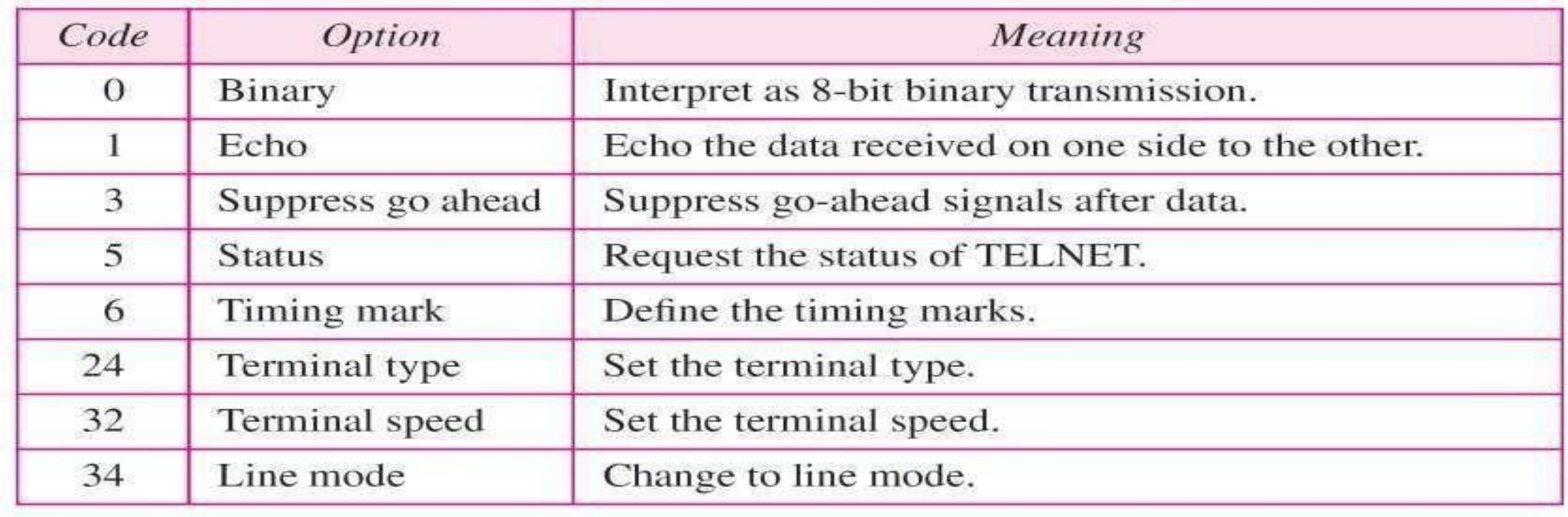
Figure 26.3 *An example of embedding*



c	a	t		f	i	l	e	a	IAC	EC	1

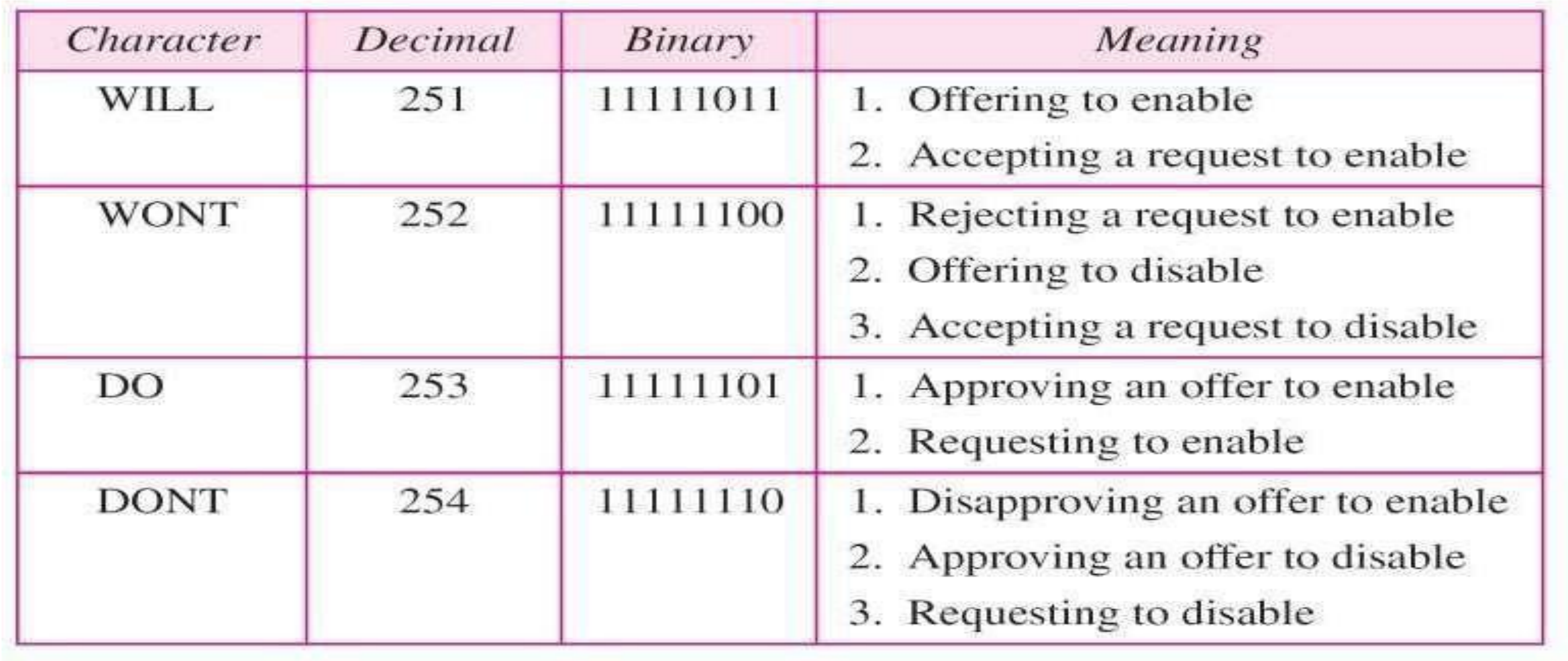
Typed at the remote terminal

Table 26.2 *Options*



Code	*Option*	*Meaning*
0	Binary	Interpret as 8-bit binary transmission.
1	Echo	Echo the data received on one side to the other.
3	Suppress go ahead	Suppress go-ahead signals after data.
5	Status	Request the status of TELNET.
6	Timing mark	Define the timing marks.
24	Terminal type	Set the terminal type.
32	Terminal speed	Set the terminal speed.
34	Line mode	Change to line mode.

Table 26.3 *NVT character set for option negotiation*



Character	*Decimal*	*Binary*	*Meaning*
WILL	251	11111011	1. Offering to enable 2. Accepting a request to enable
WONT	252	11111100	1. Rejecting a request to enable 2. Offering to disable 3. Accepting a request to disable
DO	253	11111101	1. Approving an offer to enable 2. Requesting to enable
DONT	254	11111110	1. Disapproving an offer to enable 2. Approving an offer to disable 3. Requesting to disable

-3 FILE TRANSFER

Transferring files from one computer to another is one of the most common tasks expected from a networking or internetworking environment. As a matter of fact, the greatest volume of data exchange in the Internet today is due to file transfer.

Topics discussed in this section:

File Transfer Protocol (FTP) Anonymous FTP

Note

- FTP uses the services of TCP.
- It needs two TCP connections.

- The well-known port 21 is used for the control connection and the well-known port 20 for the data connection.

Figure 26.21 *FTP*

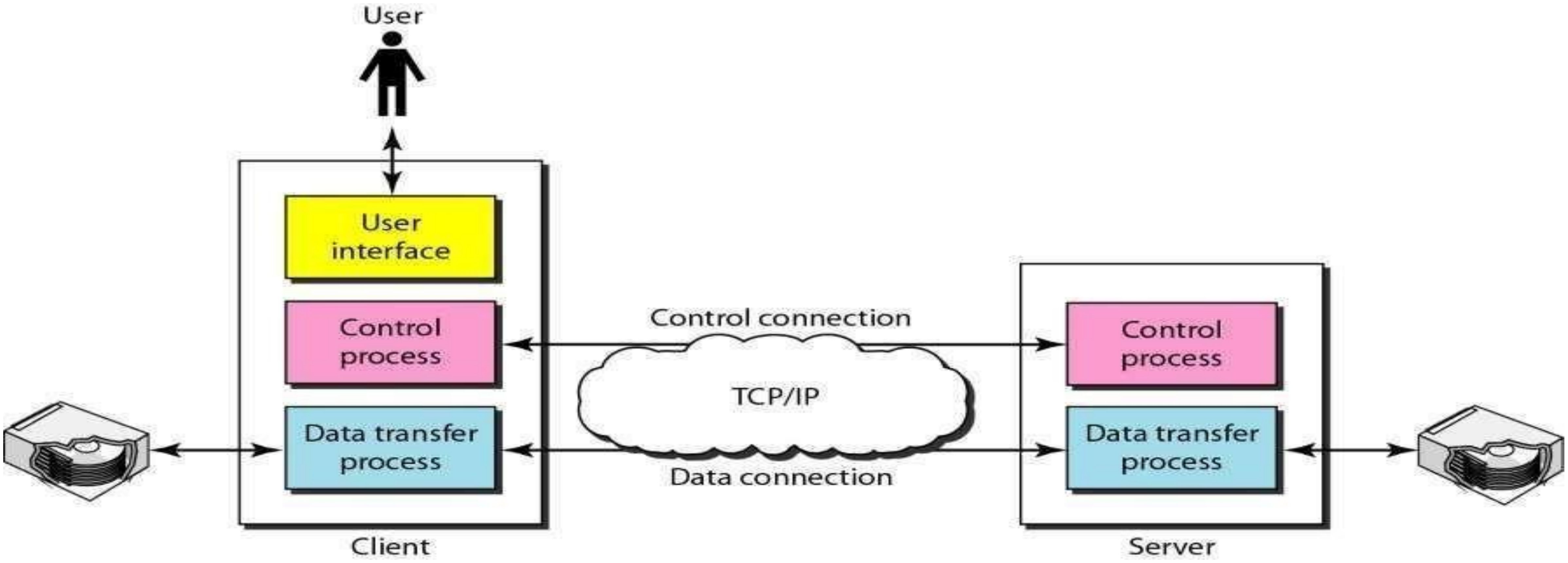


Figure 26.22 *Using the control connection*

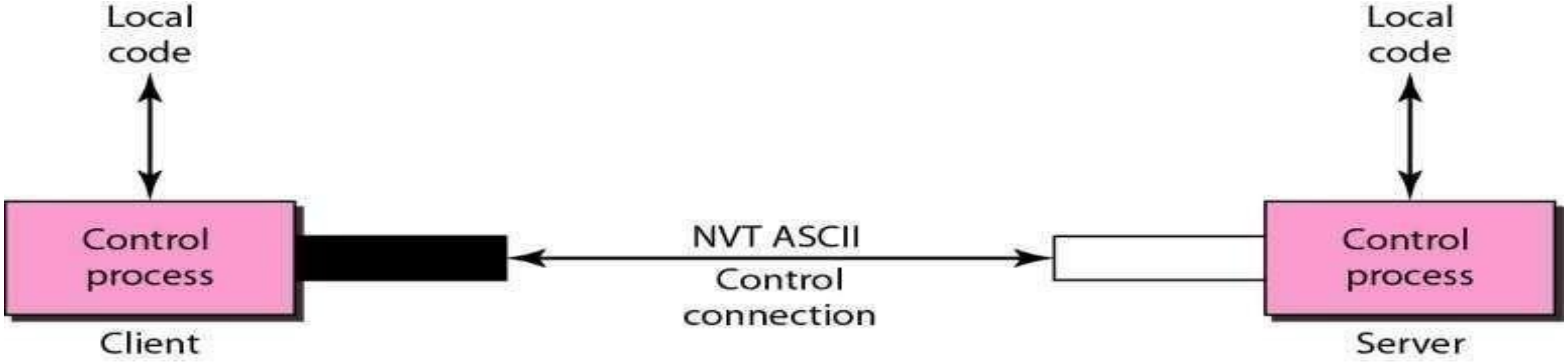
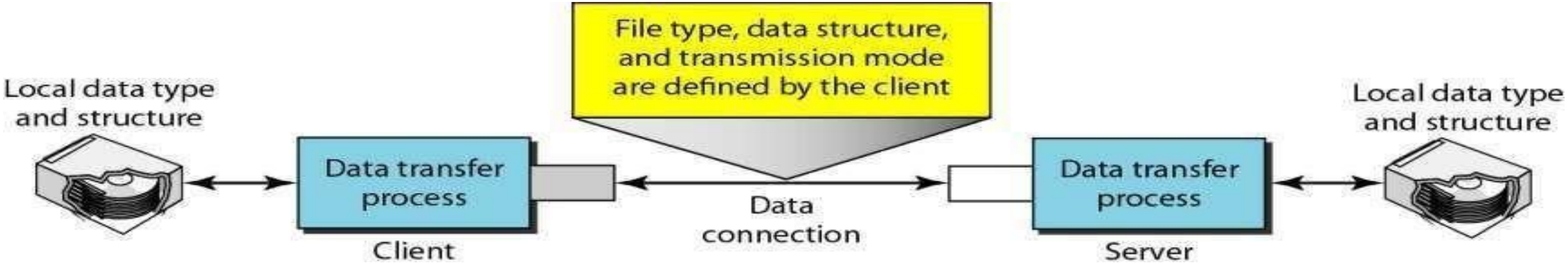


Figure 26.23 *Using the data connection*


File type, data structure, and transmission mode are defined by the client
Local data type and structure
Data transfer process
Client
Data connection
Data transfer process
Server
Local data type and structure

Thank You..