

# UNIT-II

# Data link layer

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

*Note*

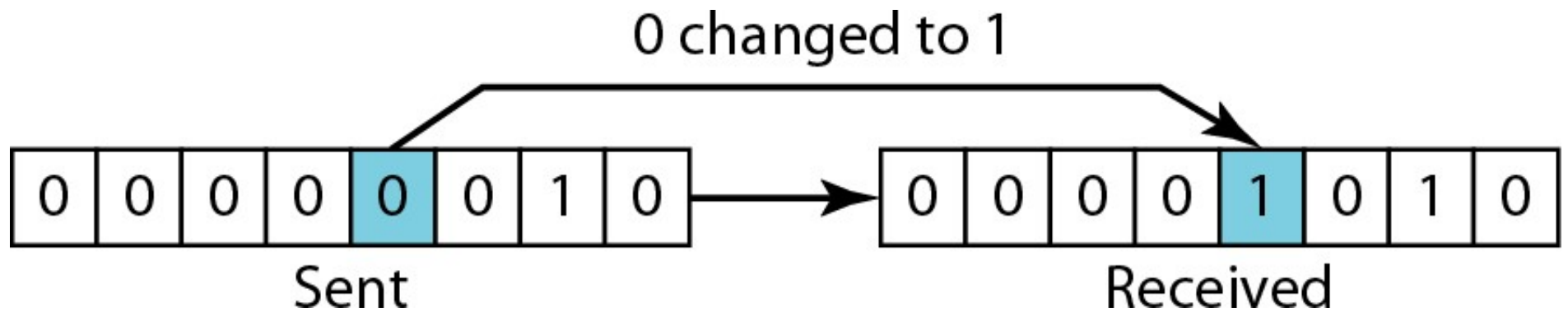
**Data can be corrupted  
during transmission.**

**Some applications require that  
errors be detected and corrected.**

*Note*

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

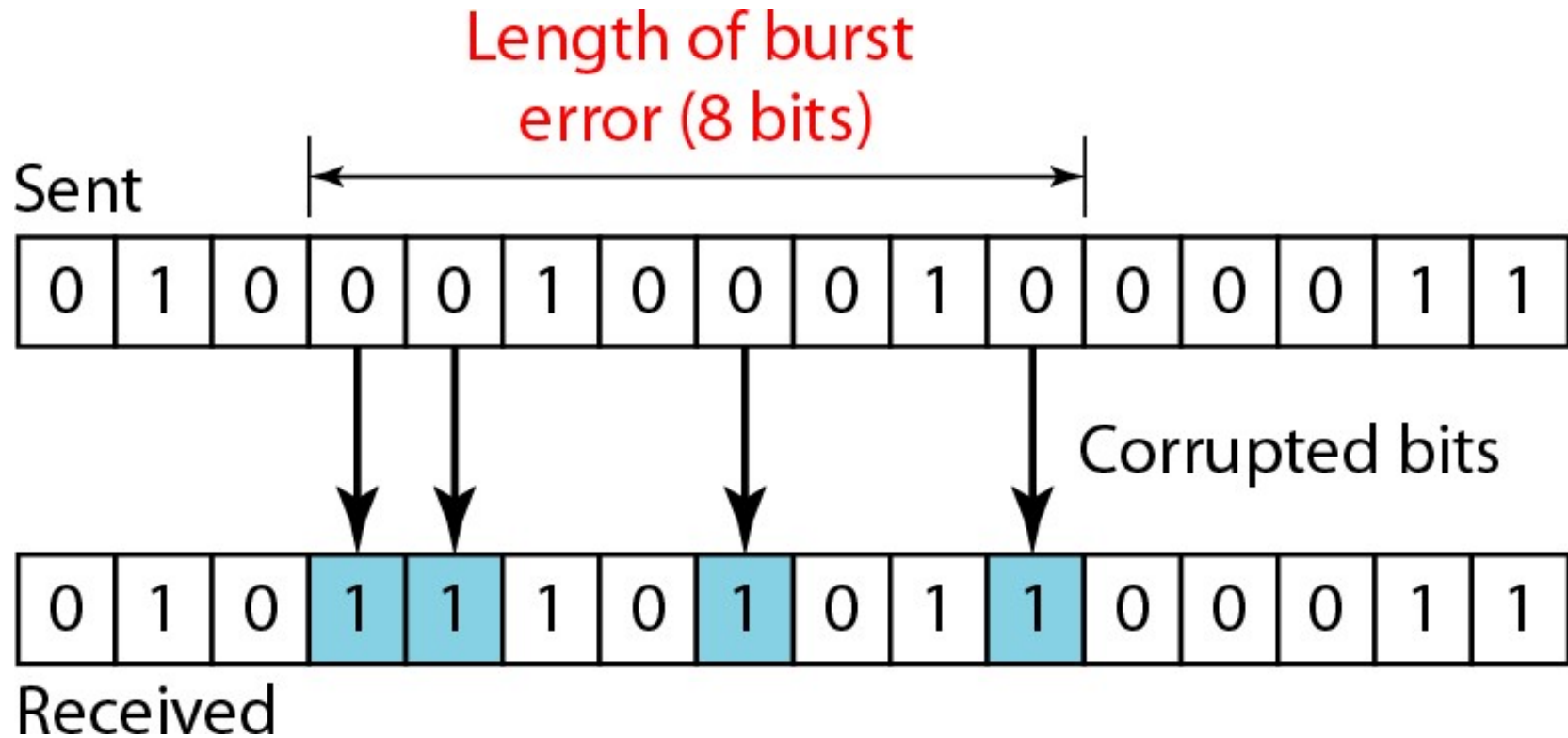
# *Single-bit error*



*Note*

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

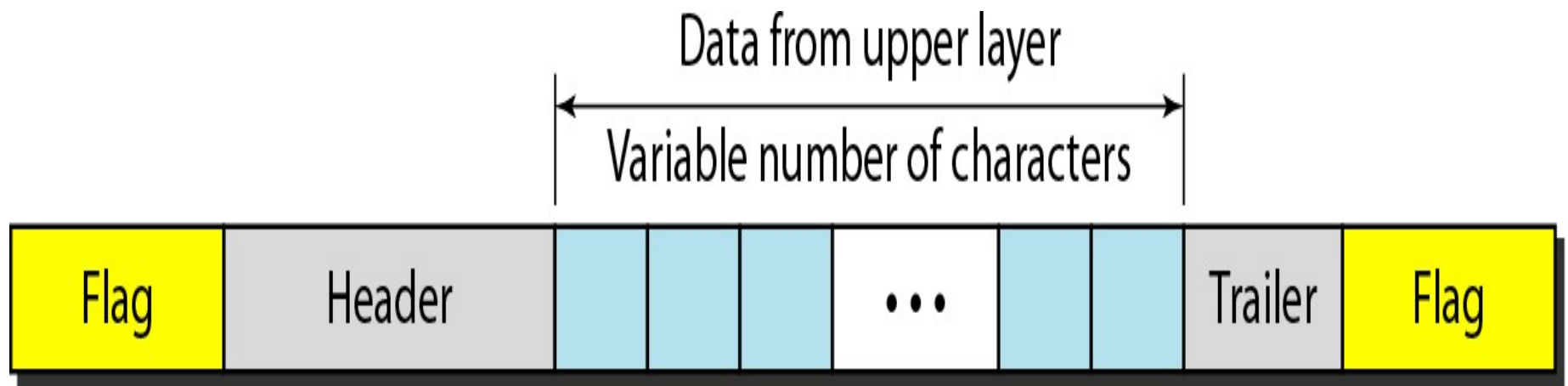
# *Burst error of length 8*



# Framing

-character oriented framing

**Figure** *A frame in a character-oriented protocol*



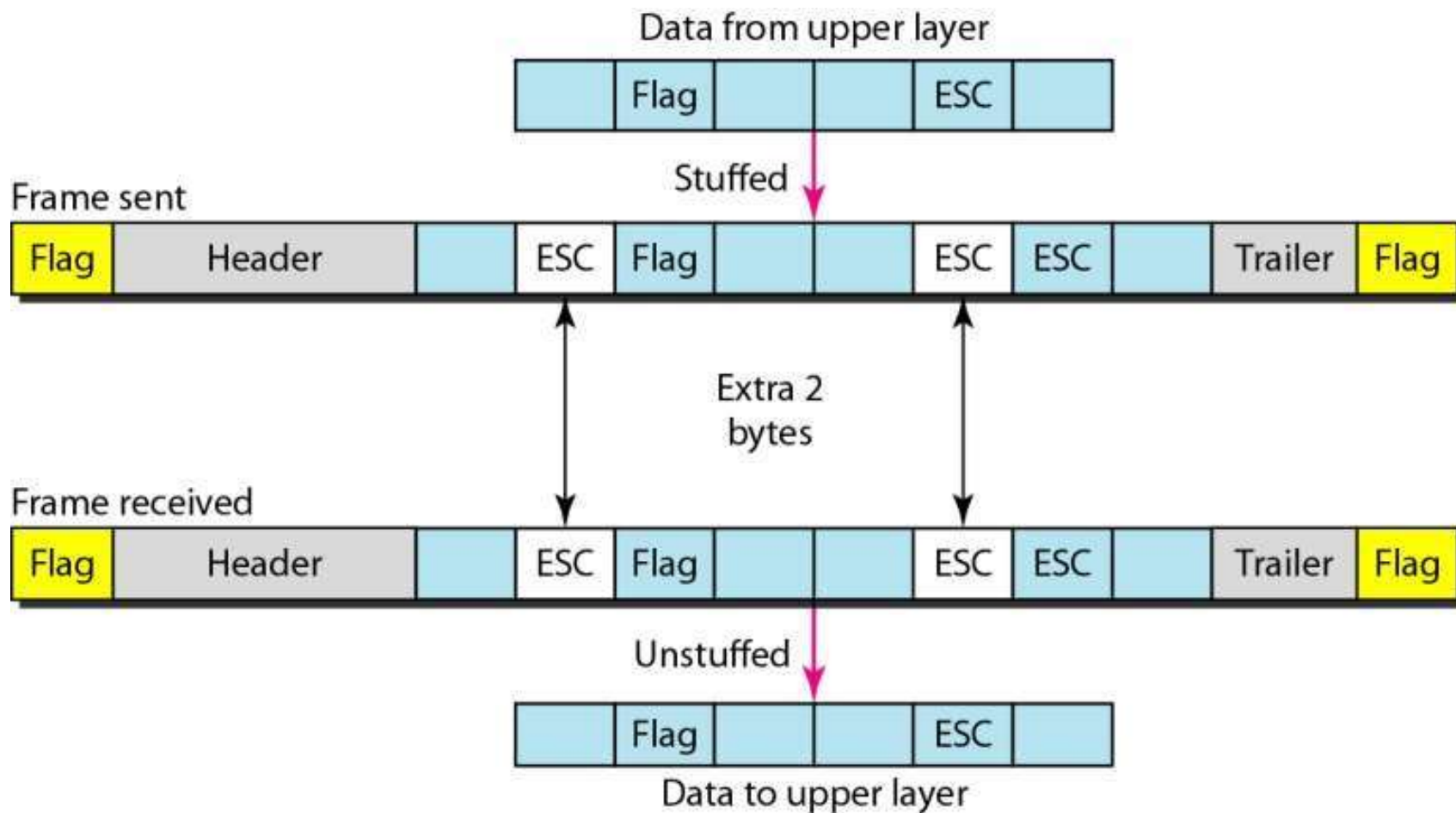


---

*Note*

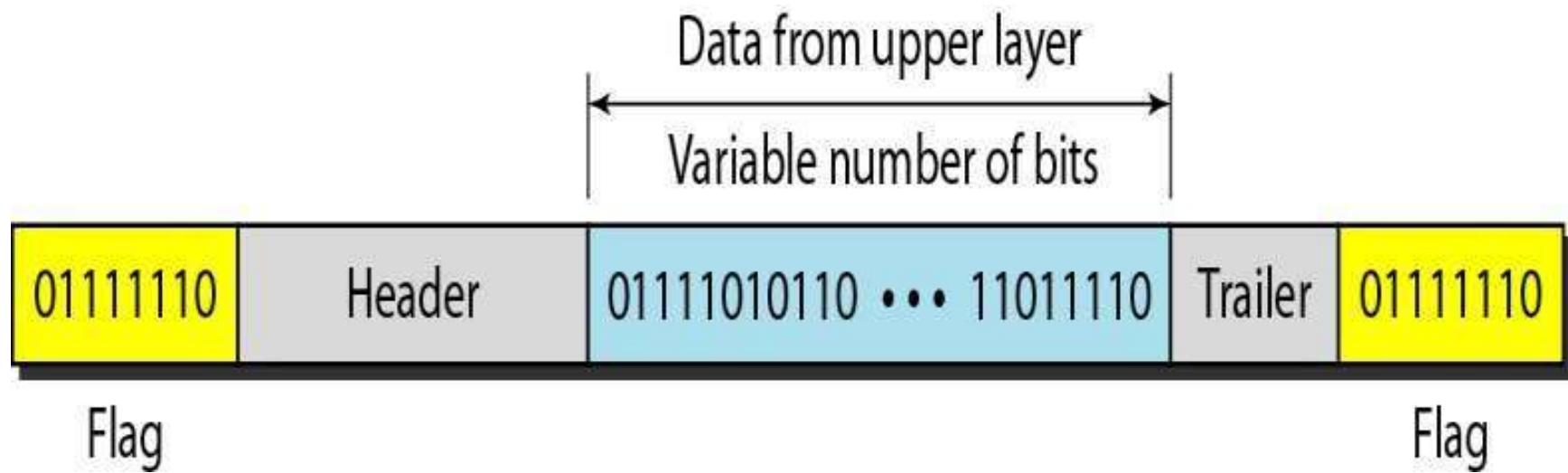
**Byte stuffing is the process of adding 1 extra byte whenever there is a flag or escape character in the text.**

# Figure *Character stuffing or Byte stuffing*



**Figure** *A frame in a bit-oriented protocol*

---

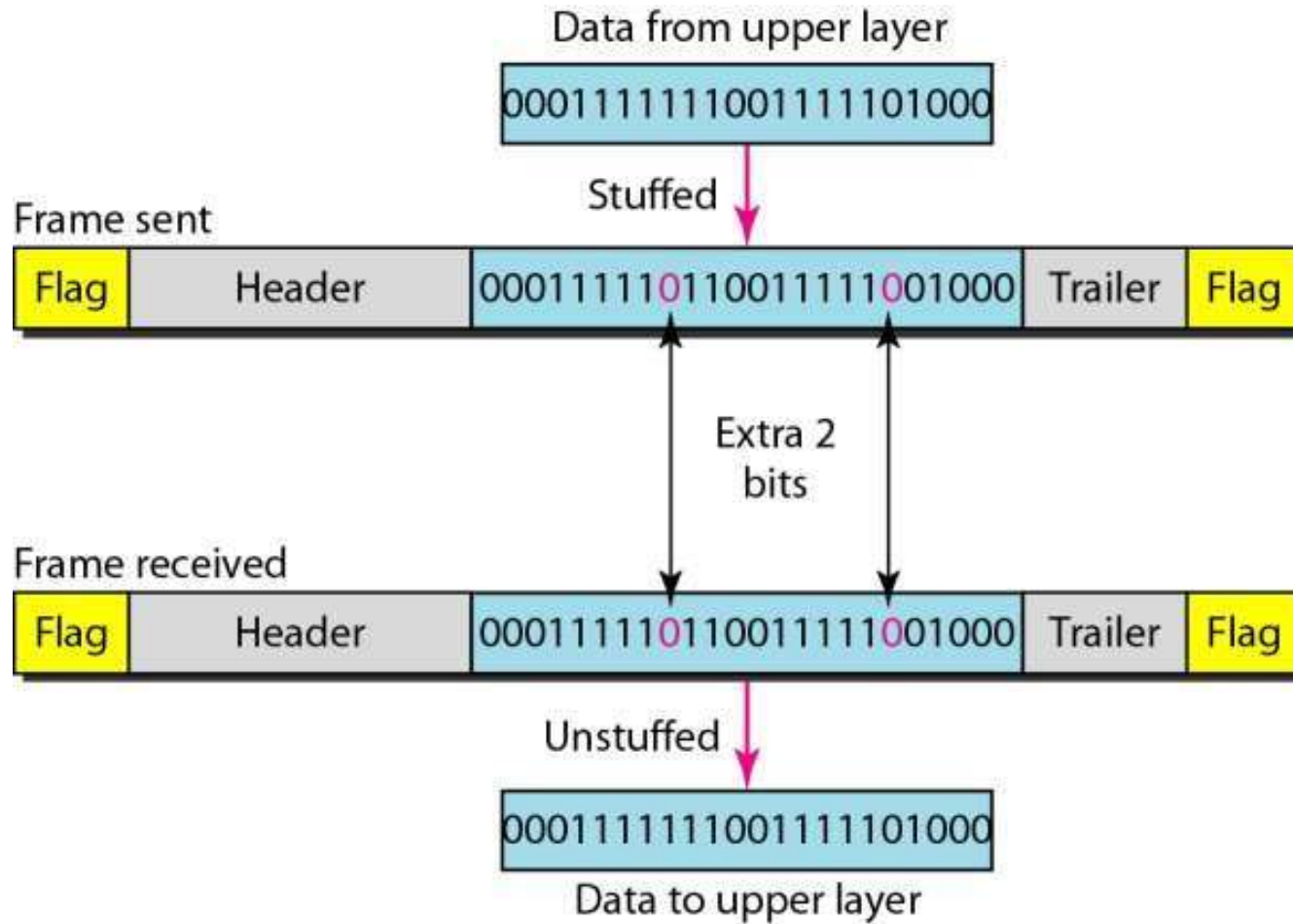




*Note*

**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 011110 for a flag.**

**Figure** *Bit stuffing and unstuffing*



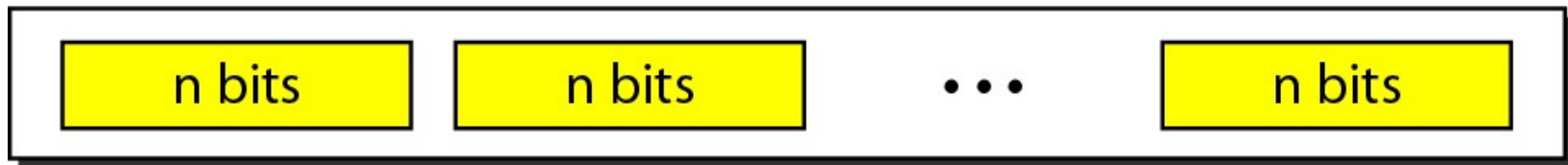
*Note*

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

**Figure** *Datawords and codewords in block coding*

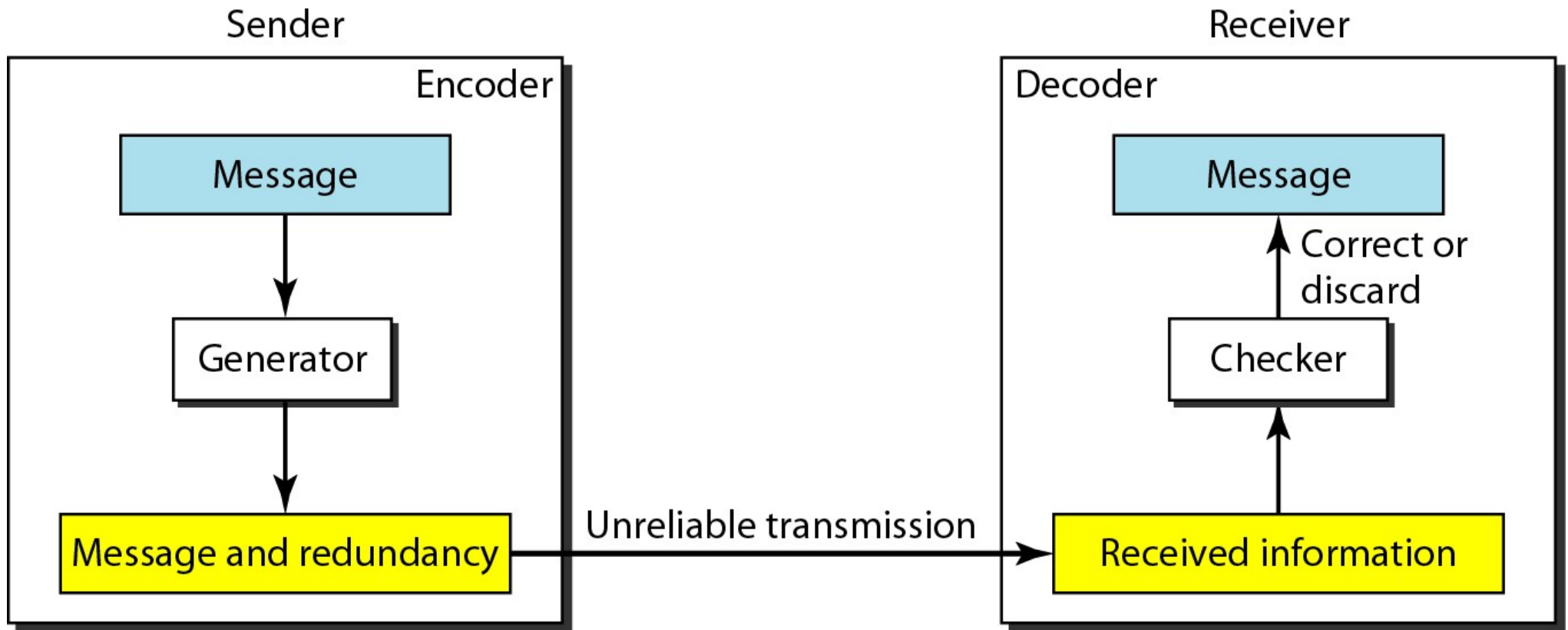


$2^k$  Datawords, each of  $k$  bits



$2^n$  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*

$$0 \oplus 0 = 0 \qquad 1 \oplus 1 = 0$$

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

$$0 \oplus 1 = 1 \qquad 1 \oplus 0 = 1$$

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

	1	0	1	1	0
$\oplus$	1	1	1	0	0
	0	1	0	1	0

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.

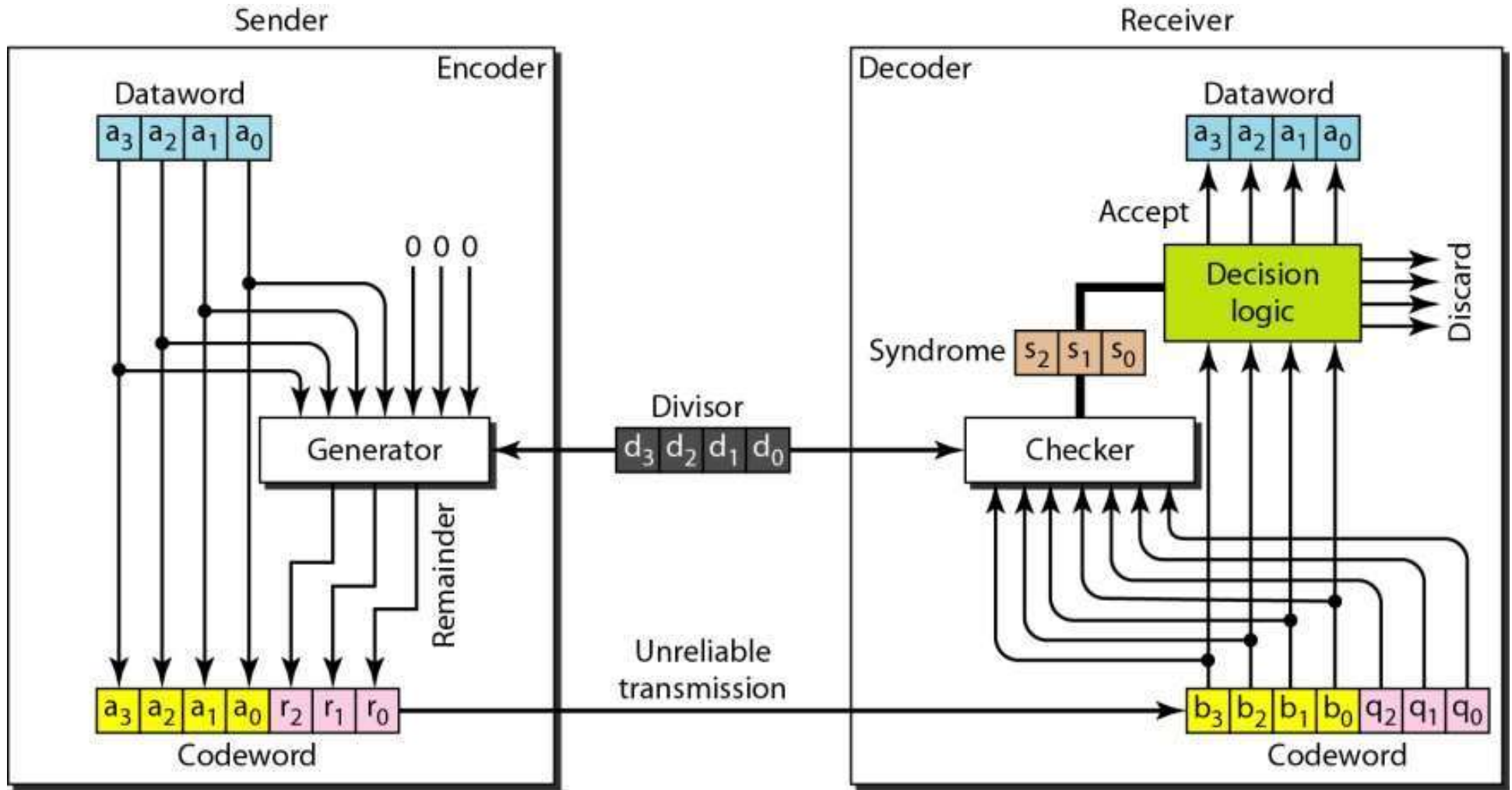
# CYCLIC CODES

***Cyclic codes*** are special linear block codes with one extra property. In a cyclic code, if a codeword is cyclically shifted (rotated), the result is another codeword.

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

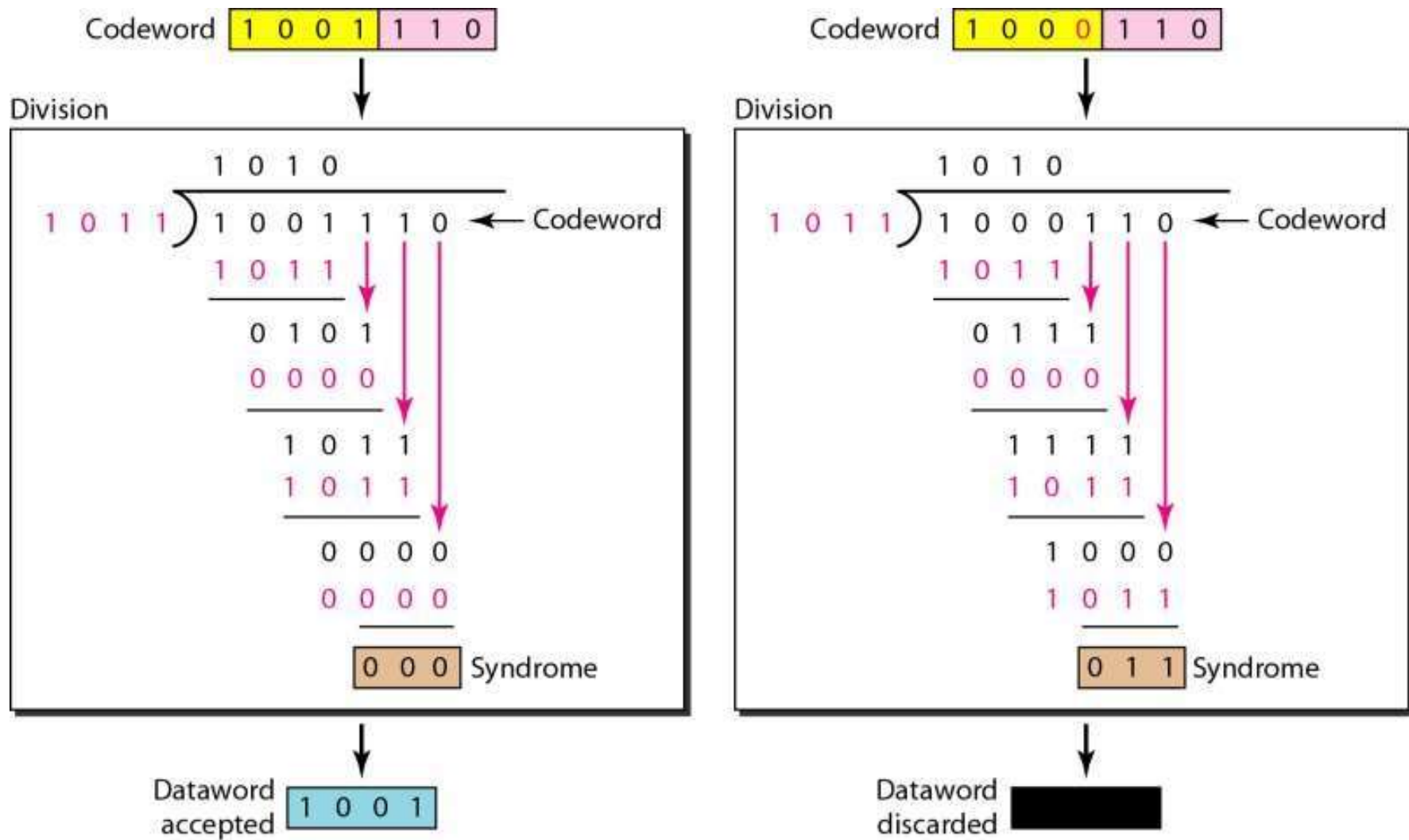
“Cyclic Redundancy check”

Figure **CRC encoder and decoder**





**Figure** *Division in the CRC decoder for two cases*



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

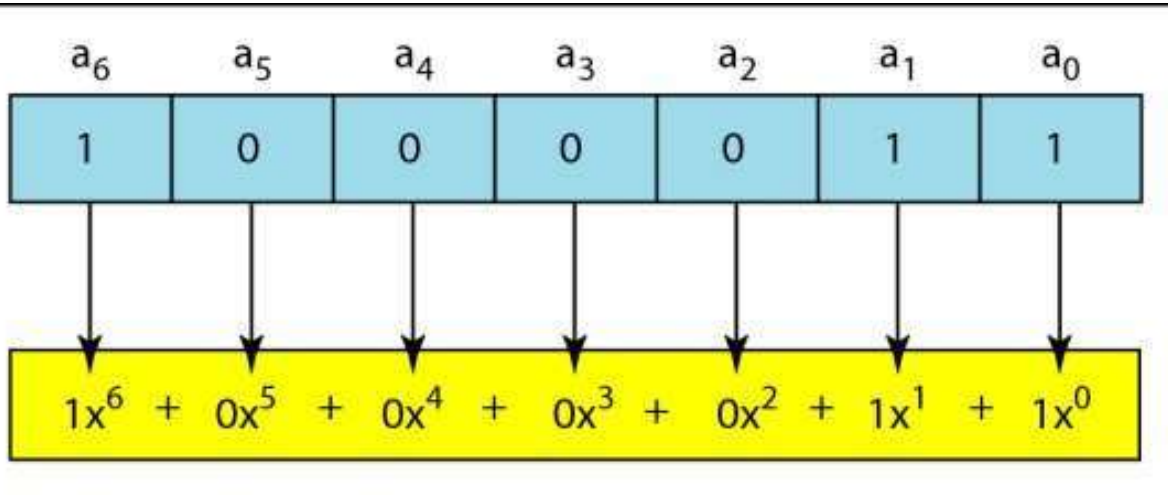
<i>Dataword</i>	<i>Codeword</i>	<i>Dataword</i>	<i>Codeword</i>
0000	0000000	1000	1000101
0001	0001011	1001	1001110
0010	0010110	1010	1010011
0011	0011101	1011	1011000
0100	0100111	1100	1100010
0101	0101100	1101	1101001
0110	0110001	1110	1110100
0111	0111010	1111	1111111



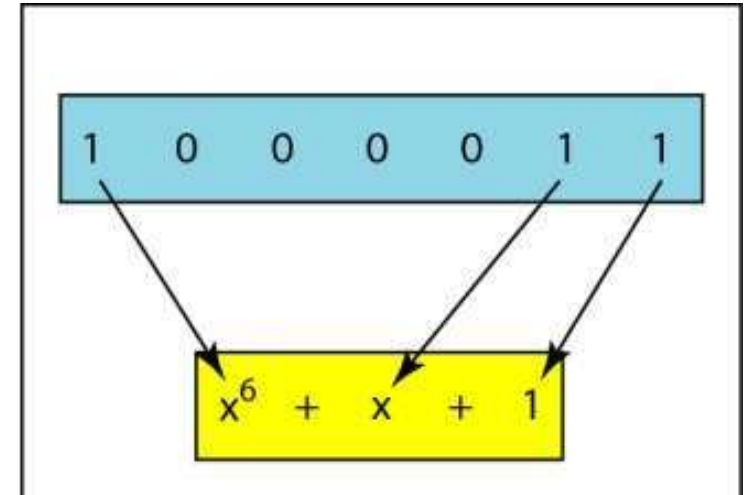
# Using Polynomials

- 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*

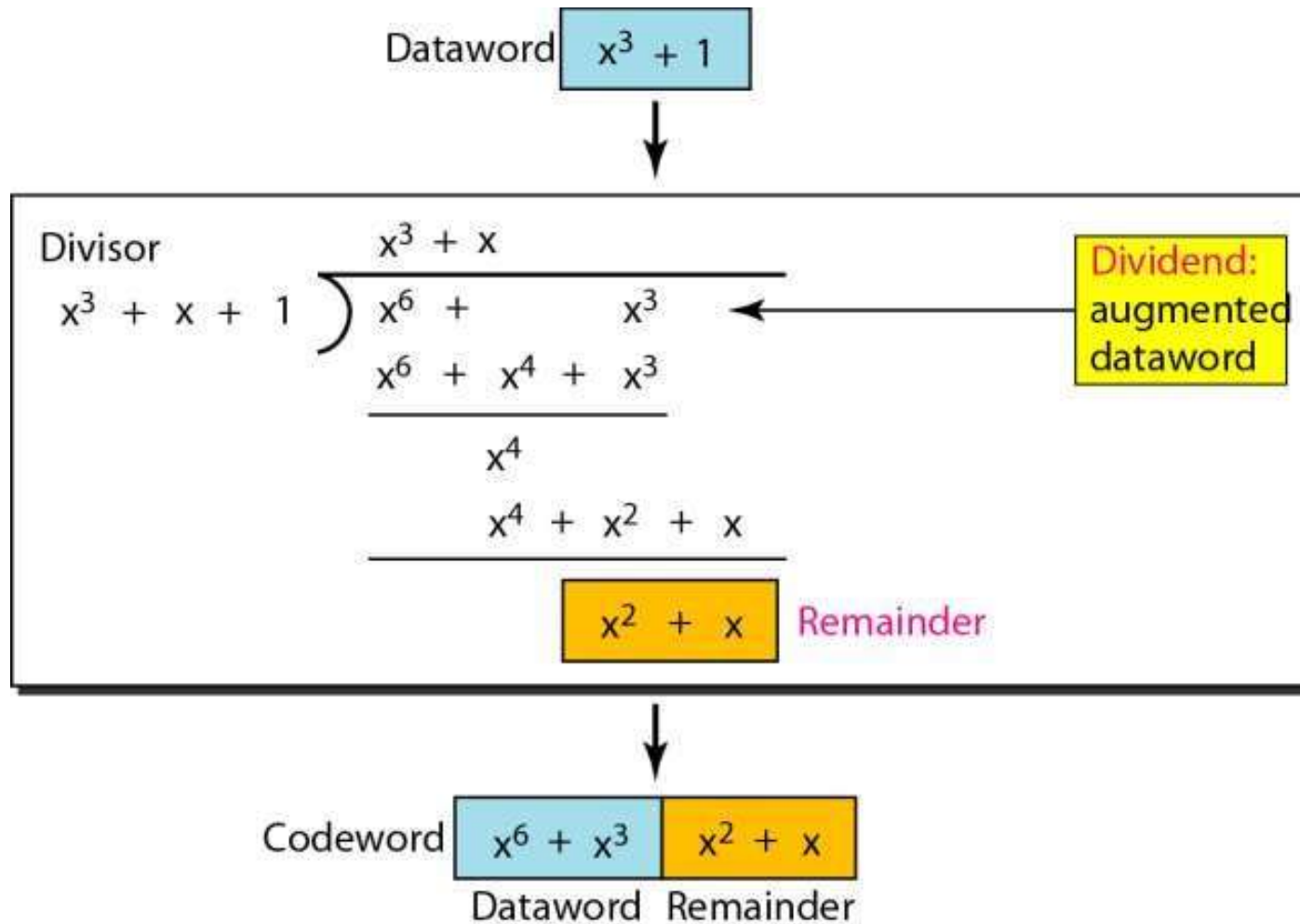


a. Binary pattern and polynomial



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*

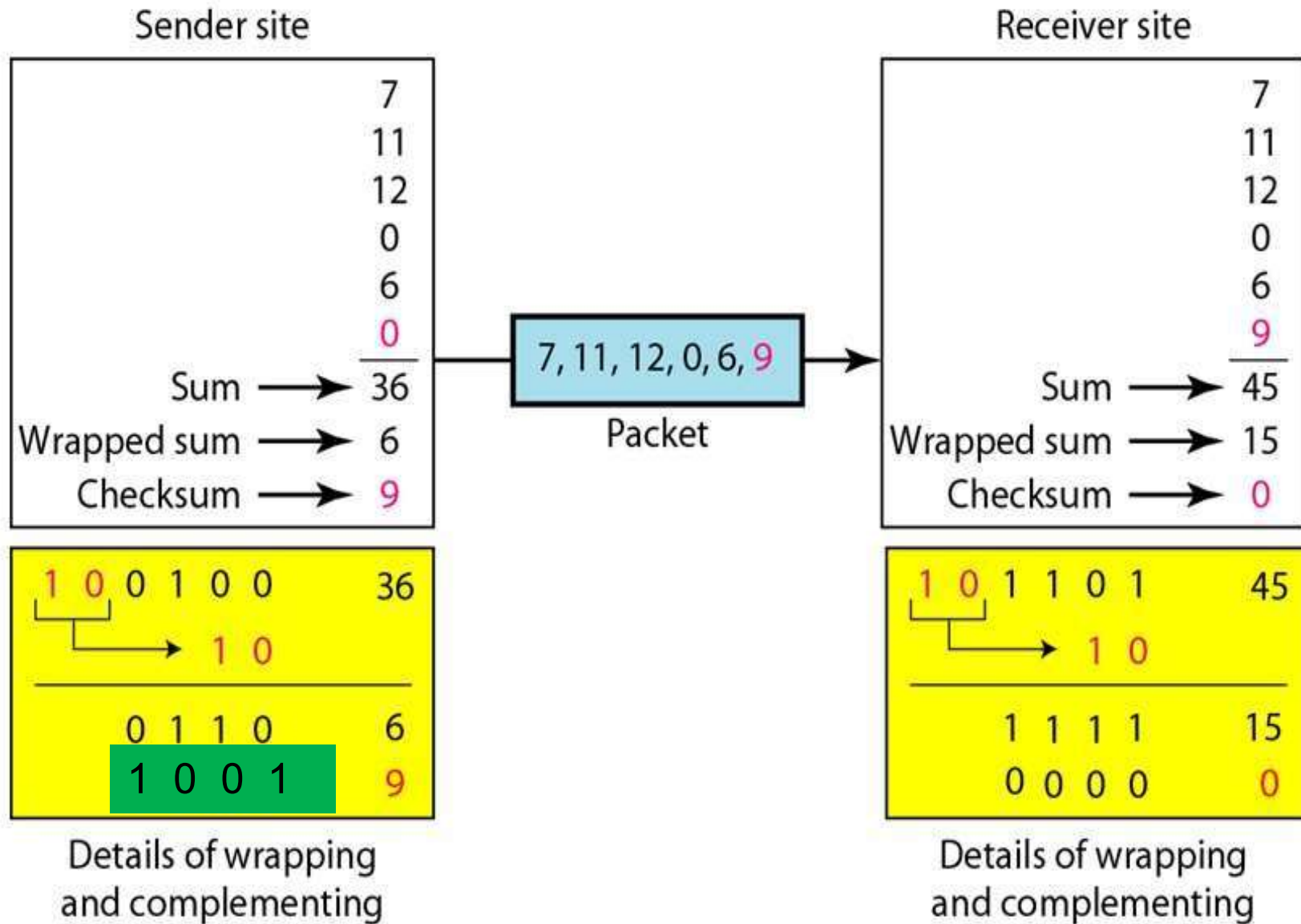
<i>Name</i>	<i>Polynomial</i>	<i>Application</i>
CRC-8	$x^8 + x^2 + x + 1$	ATM header
CRC-10	$x^{10} + x^9 + x^5 + x^4 + x^2 + 1$	ATM AAL
CRC-16	$x^{16} + x^{12} + x^5 + 1$	HDLC
CRC-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	LANs



# CHECKSUM

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

# checksum



# 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)
<hr/>					
8	F	C	6		Sum (partial)
<hr/>					
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)
<hr/>					
F	F	F	E		Sum (partial)
<hr/>					
F	F	F	F		Sum
0	0	0	0		Checksum (new)

a. Checksum at the receiver site

# FLOW AND 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.**

---



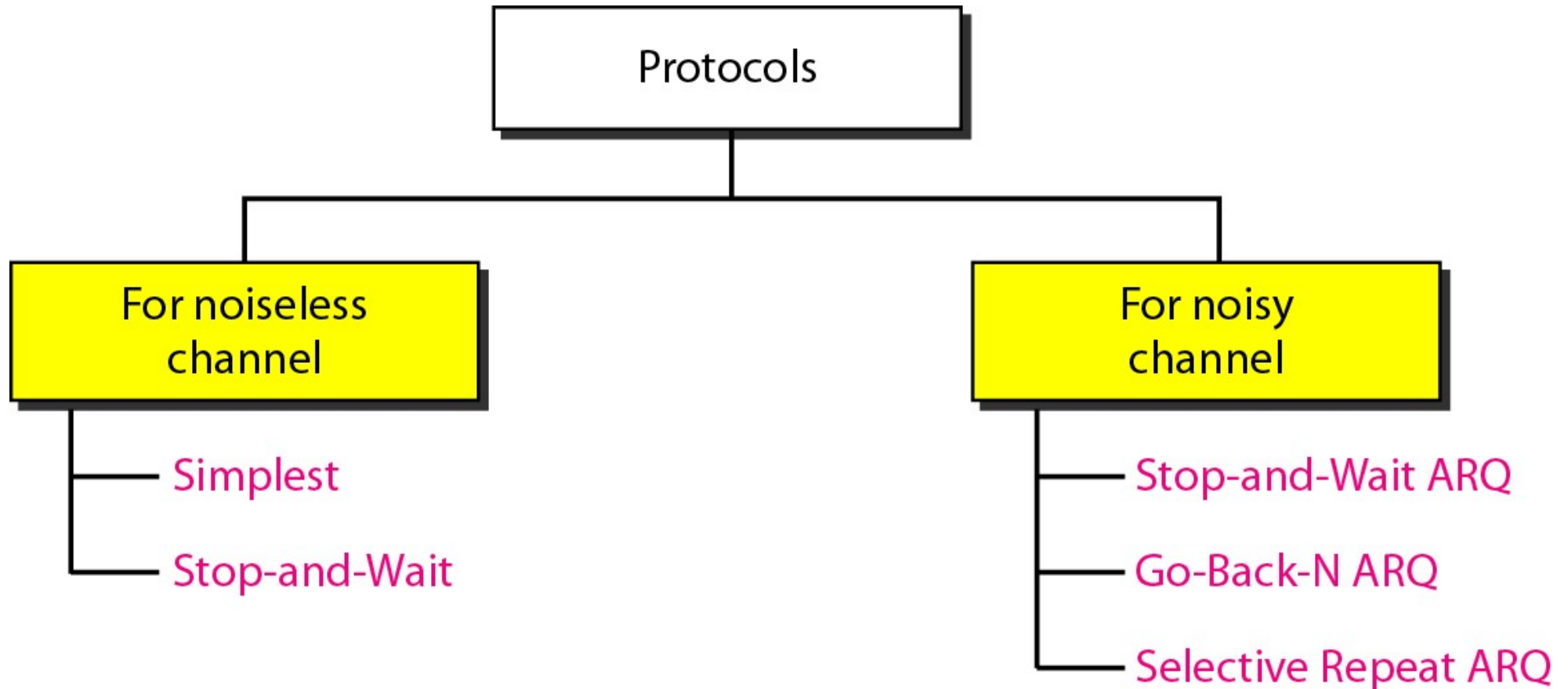
---

*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

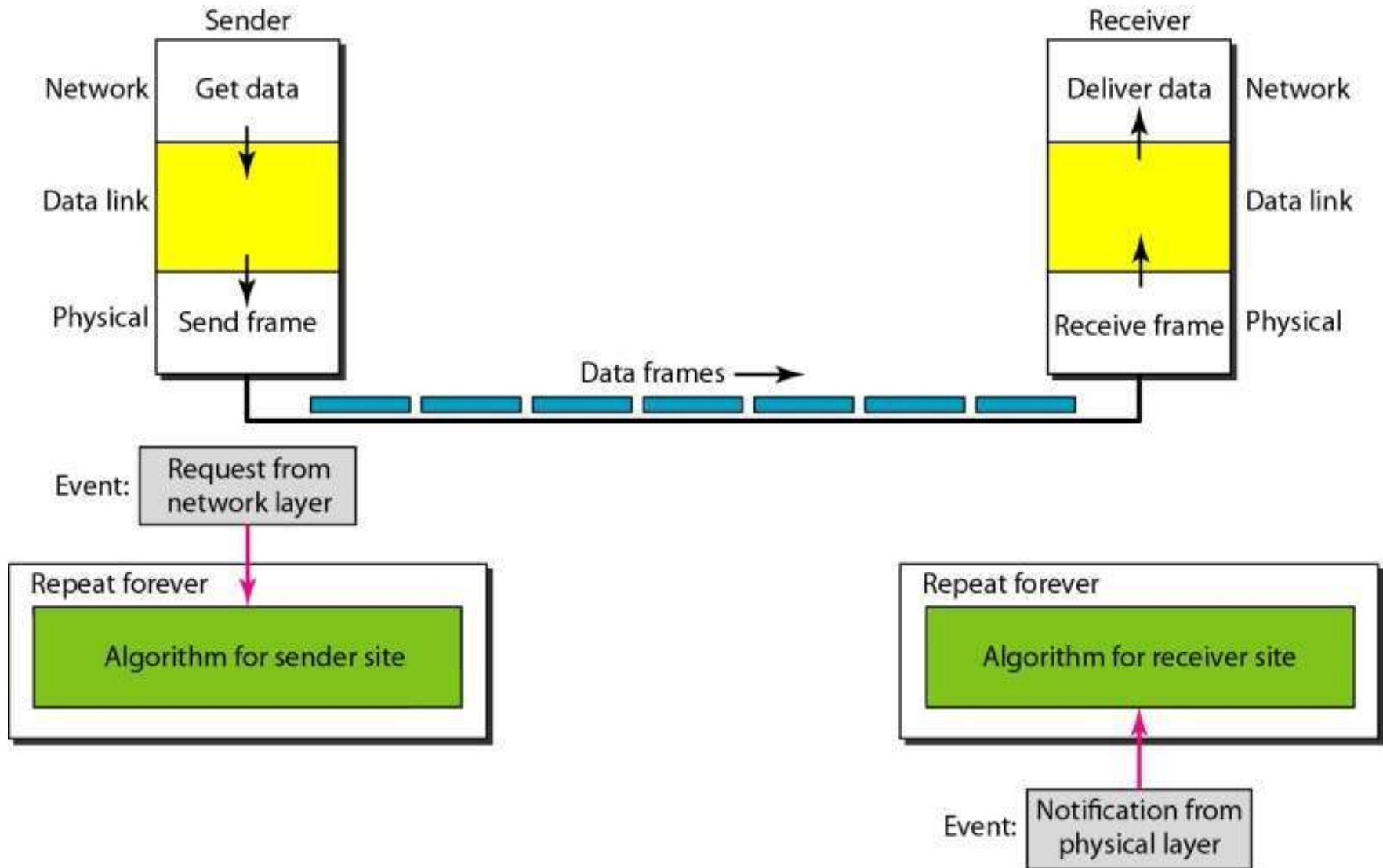
*Let us first assume we have an ideal channel in which no frames are lost, duplicated, or corrupted. We introduce two protocols for this type of channel.*

**Topics discussed in this section:**

**Simplest Protocol**

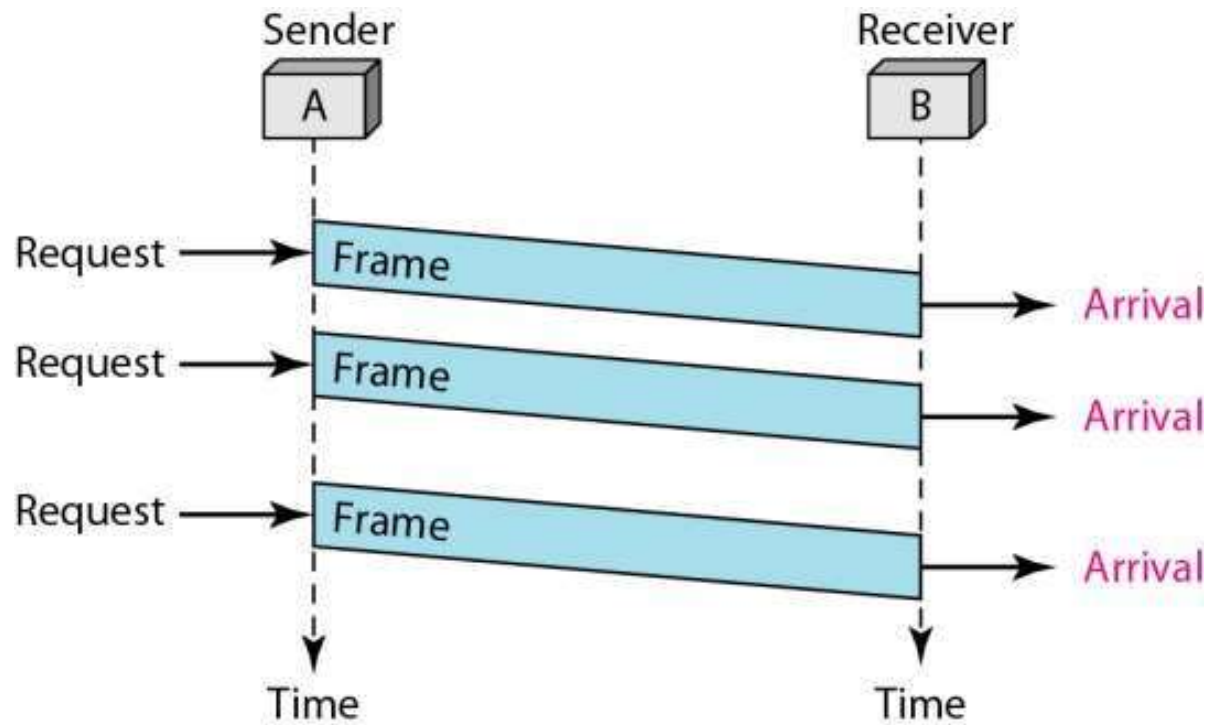
**Stop-and-Wait Protocol**

# *simplest protocol* with no flow or error control

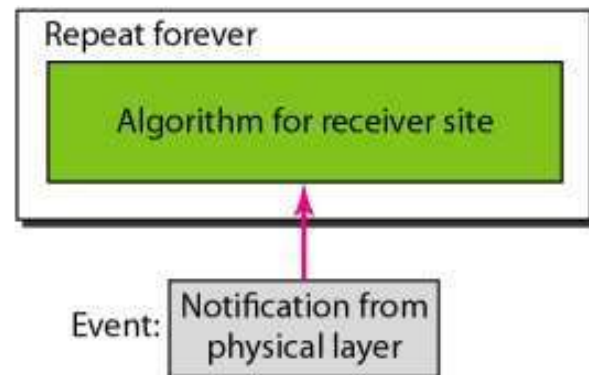
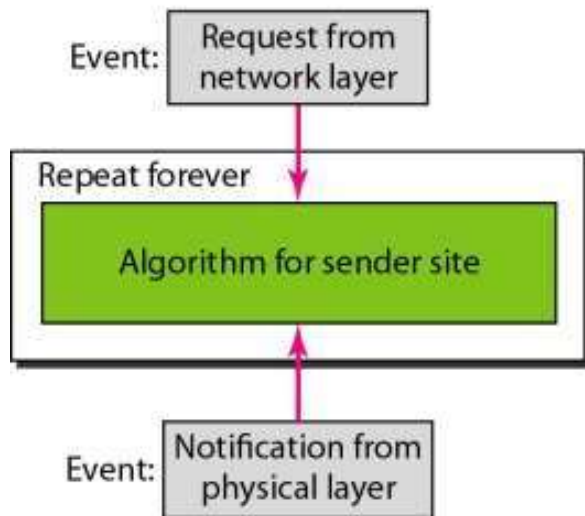
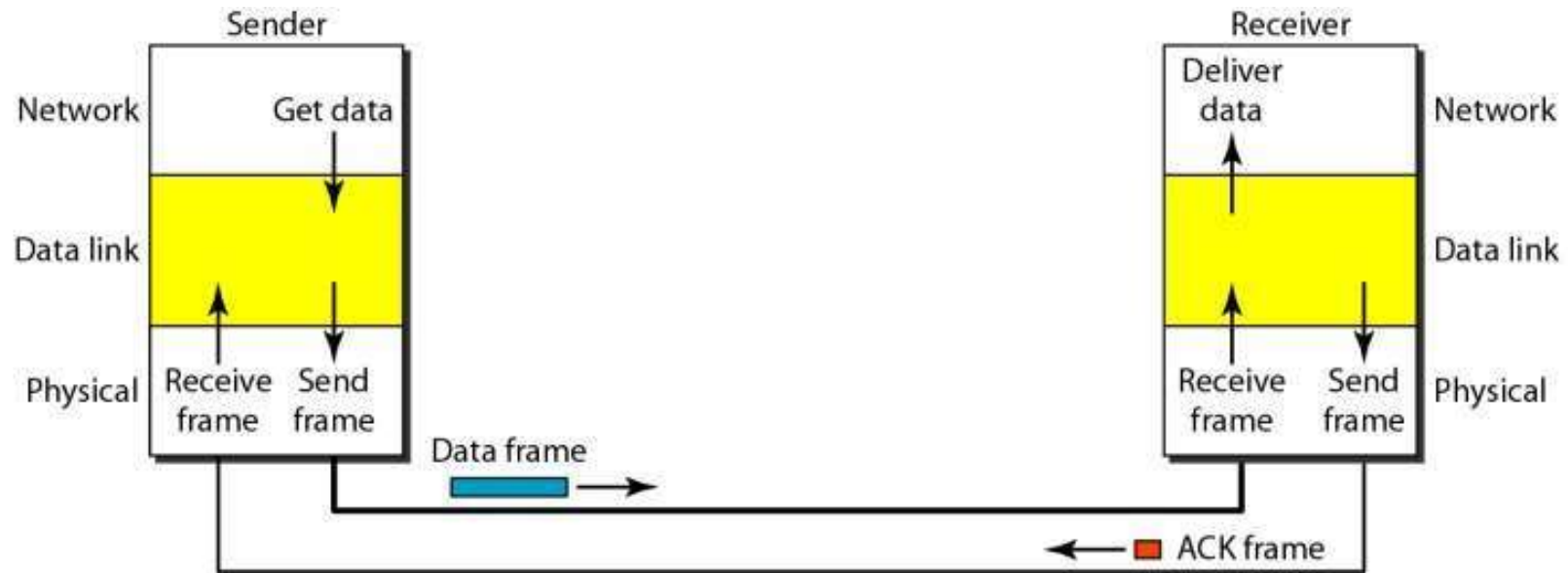


# Figure *Flow diagram*

---

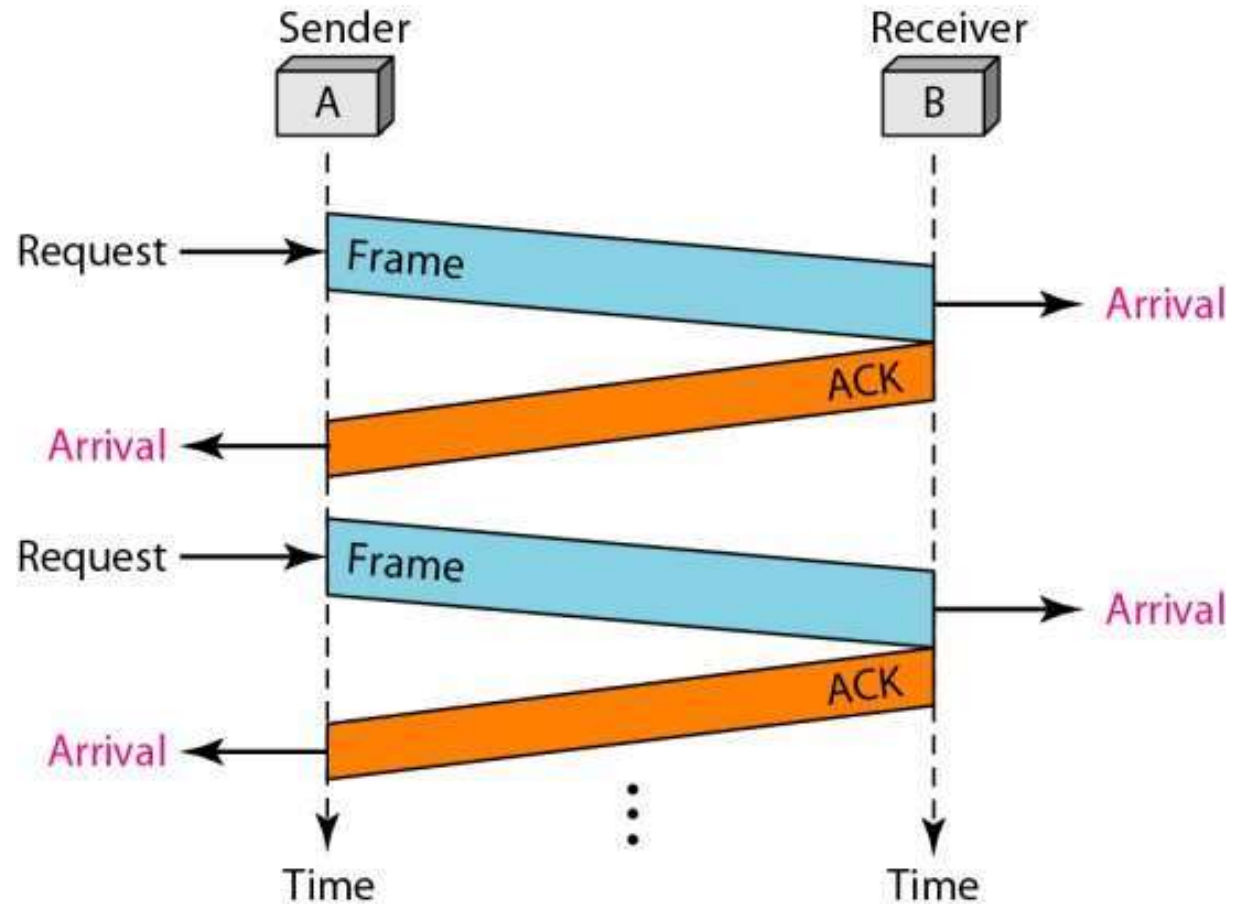


# Figure *Design of Stop-and-Wait Protocol*



**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:*

**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.**



---

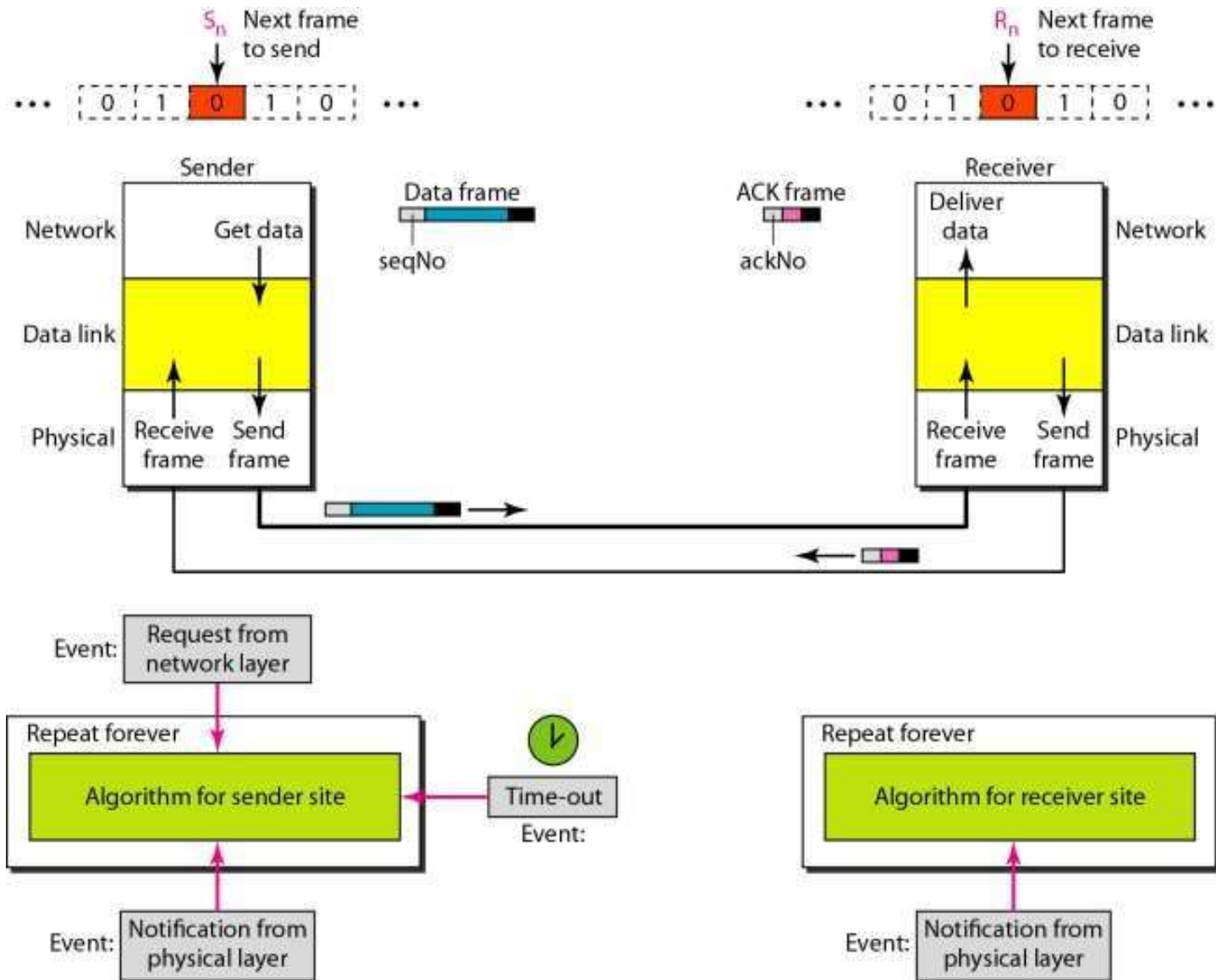
*Note*

---

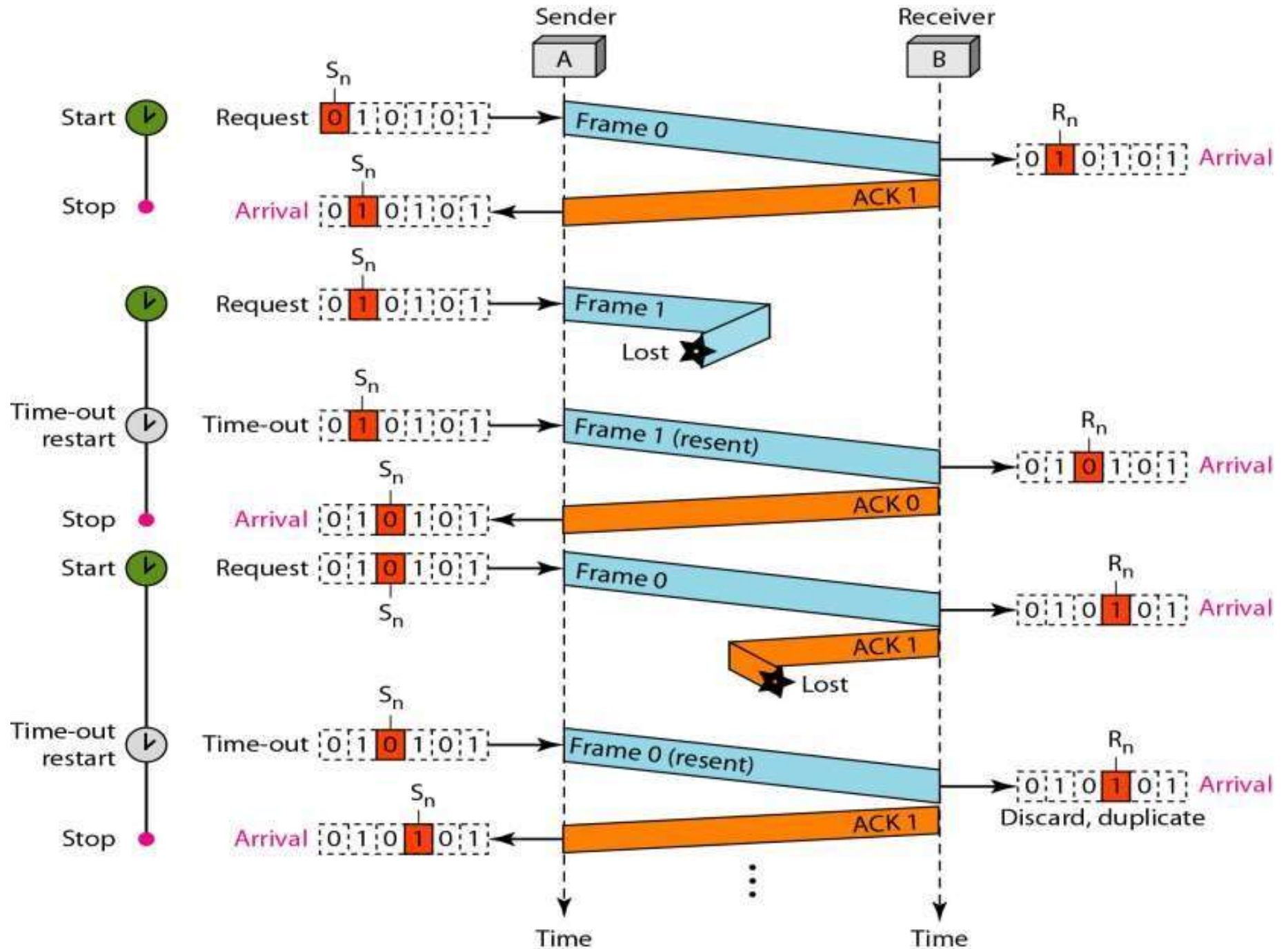
**In Stop-and-Wait ARQ, we use sequence numbers to number the frames.**

---

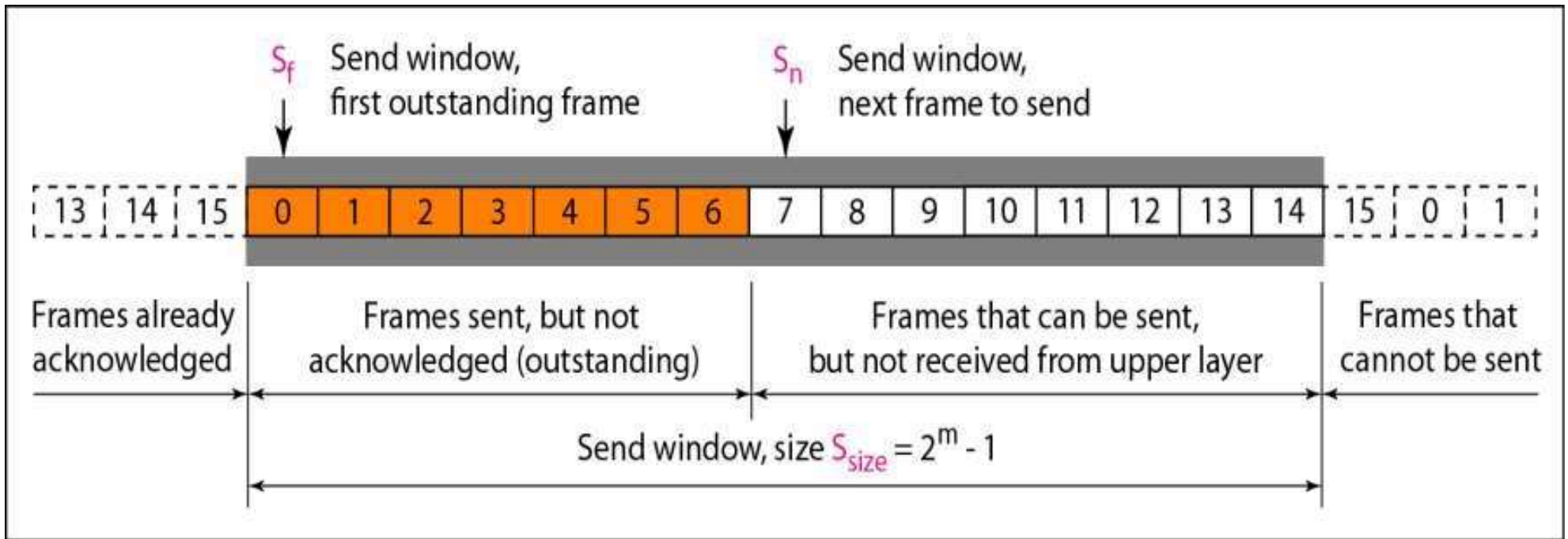
# Figure *Design of the Stop-and-Wait ARQ Protocol*



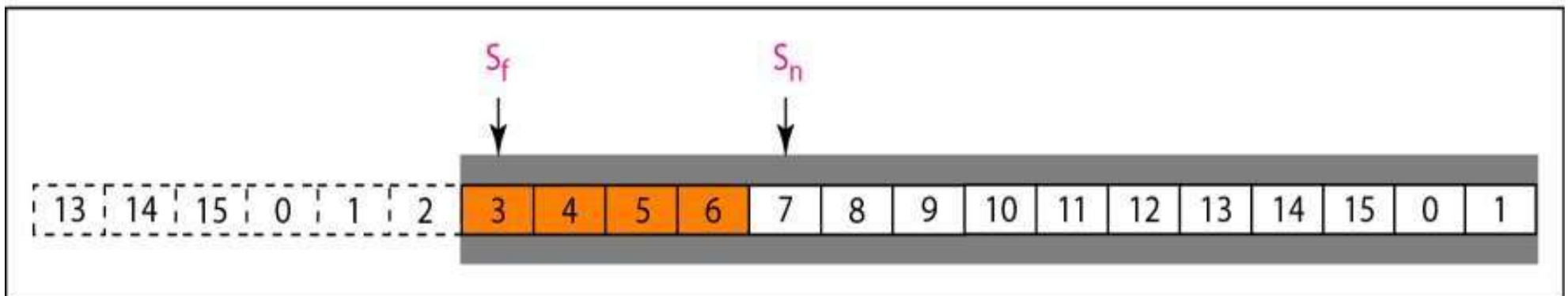
**Figure** Flow diagram *Stop-and-Wait ARQ Protocol*



# Figure Send window for *Go-Back-N ARQ*



a. Send window before sliding



b. Send window after sliding

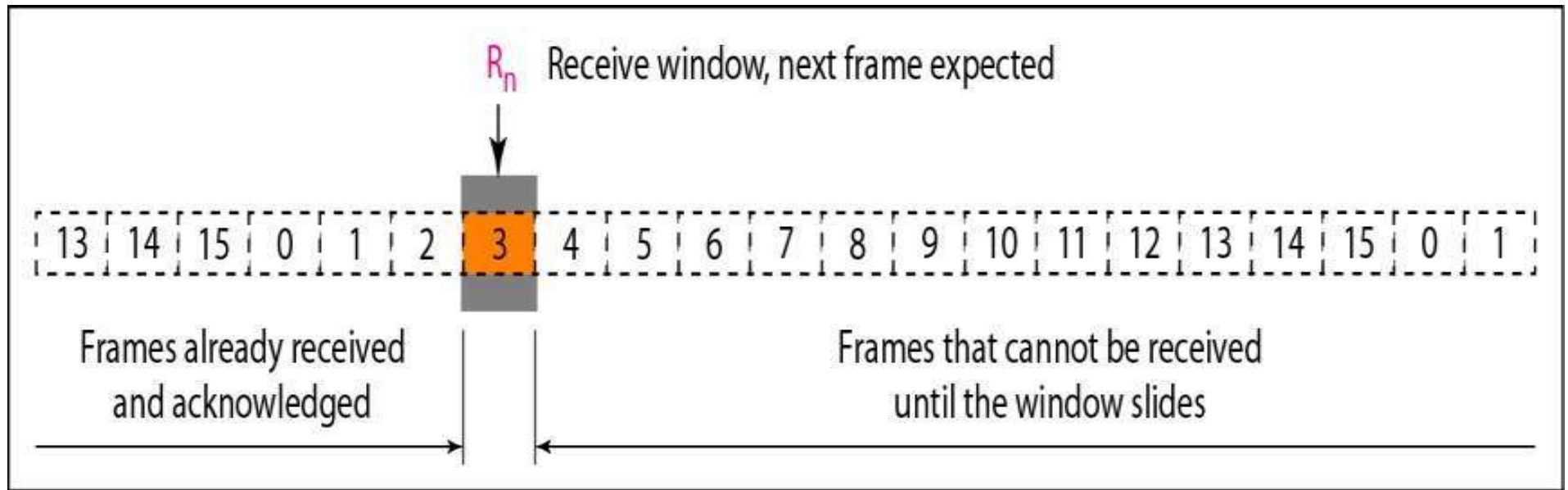


---

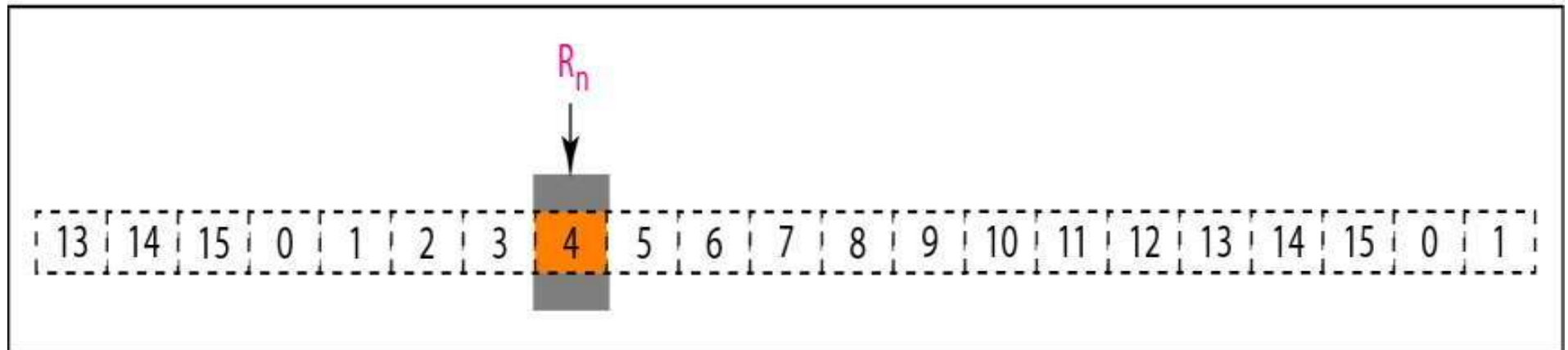
*Note*

**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*



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*

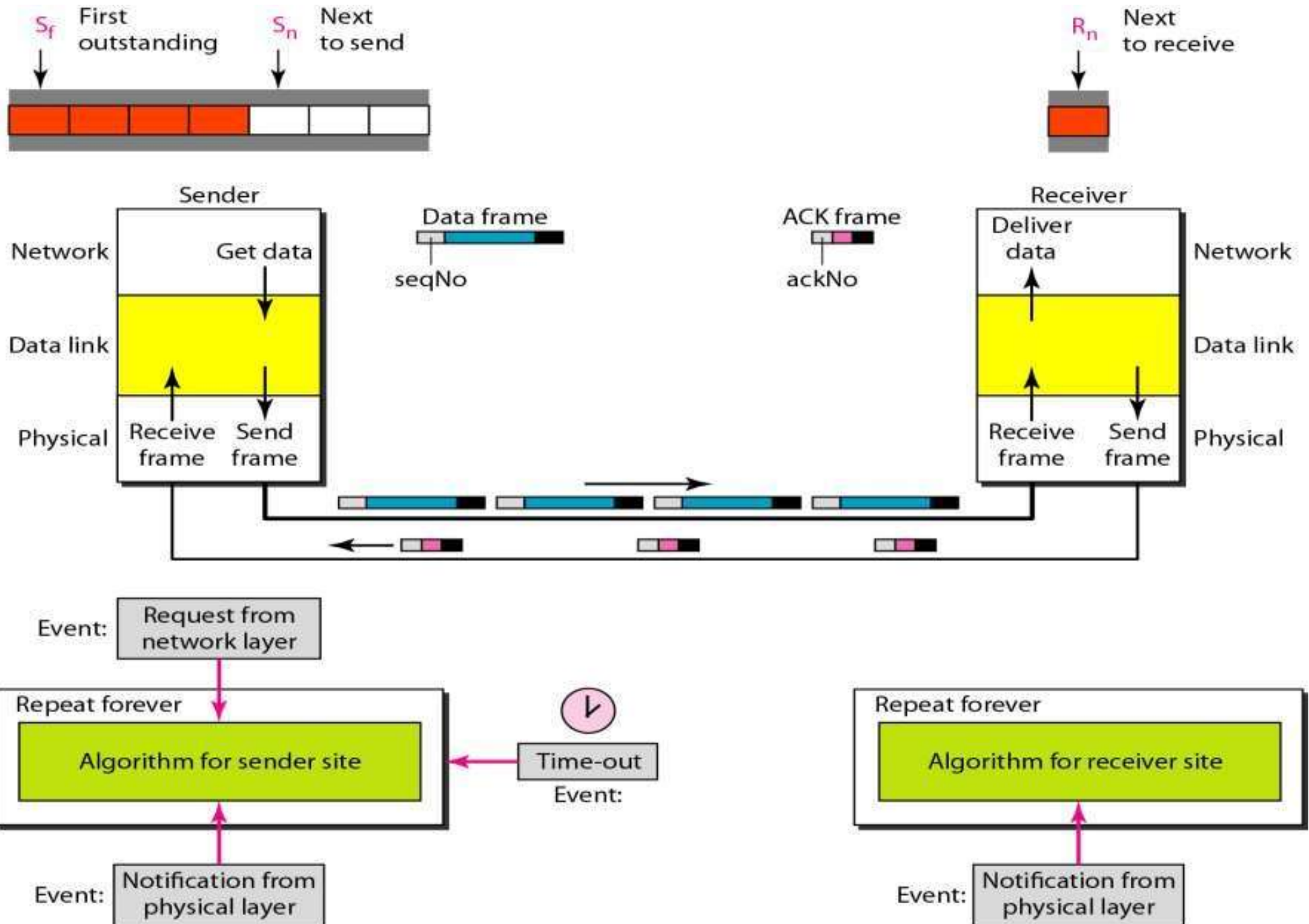
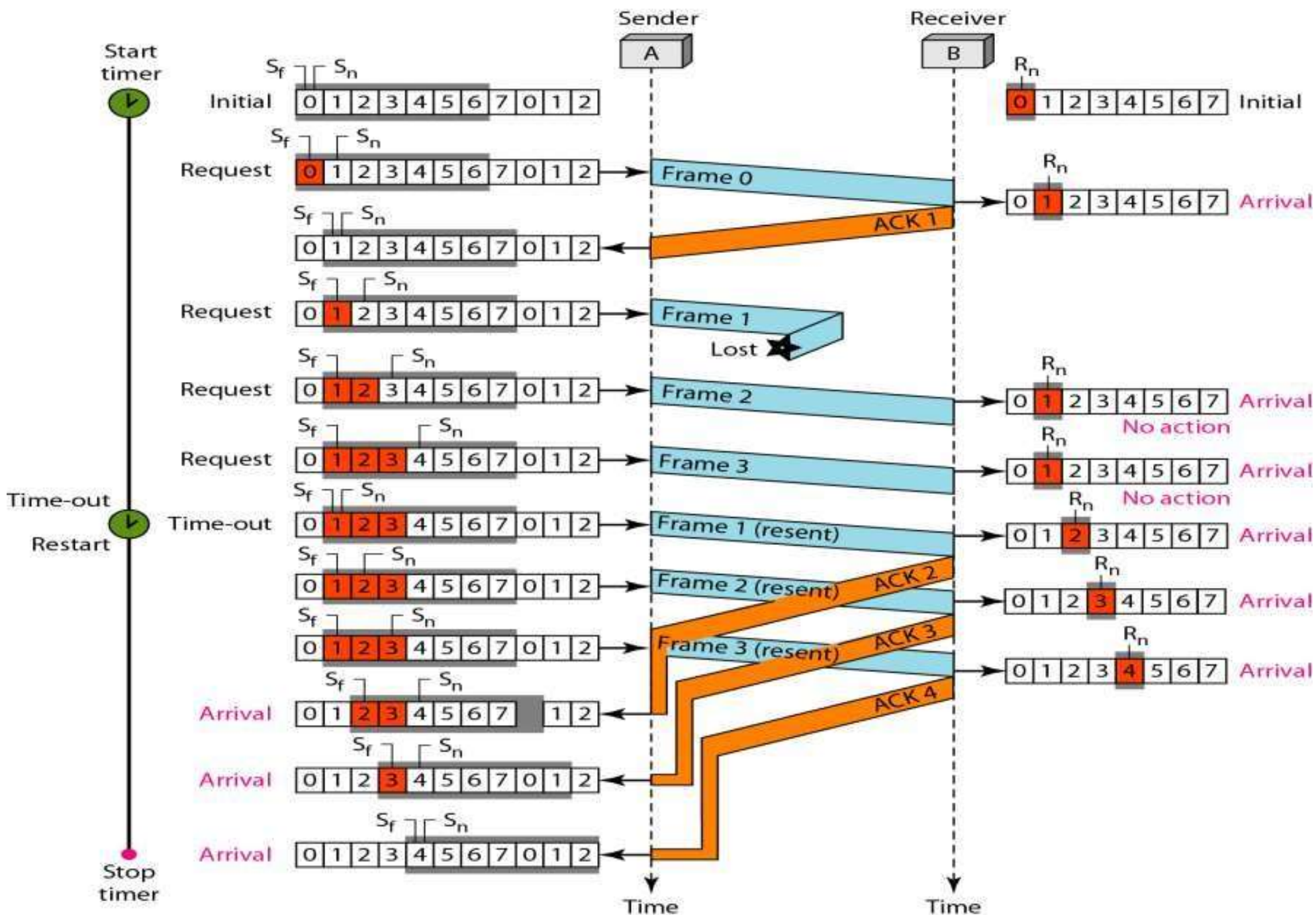


Figure 11.17 Flow diagram **GO-BACK-n Protocol**



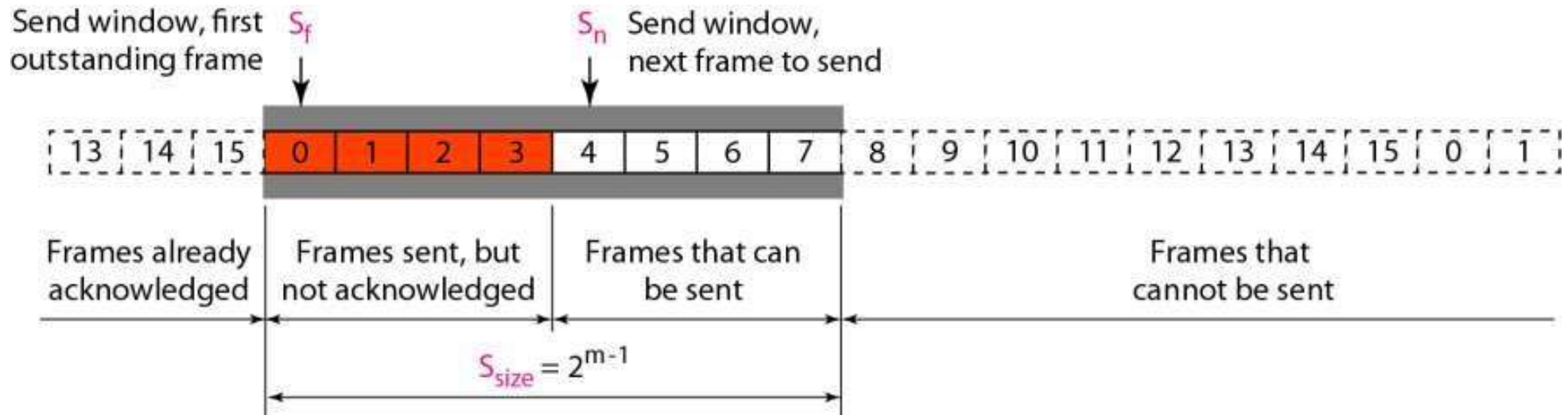


---

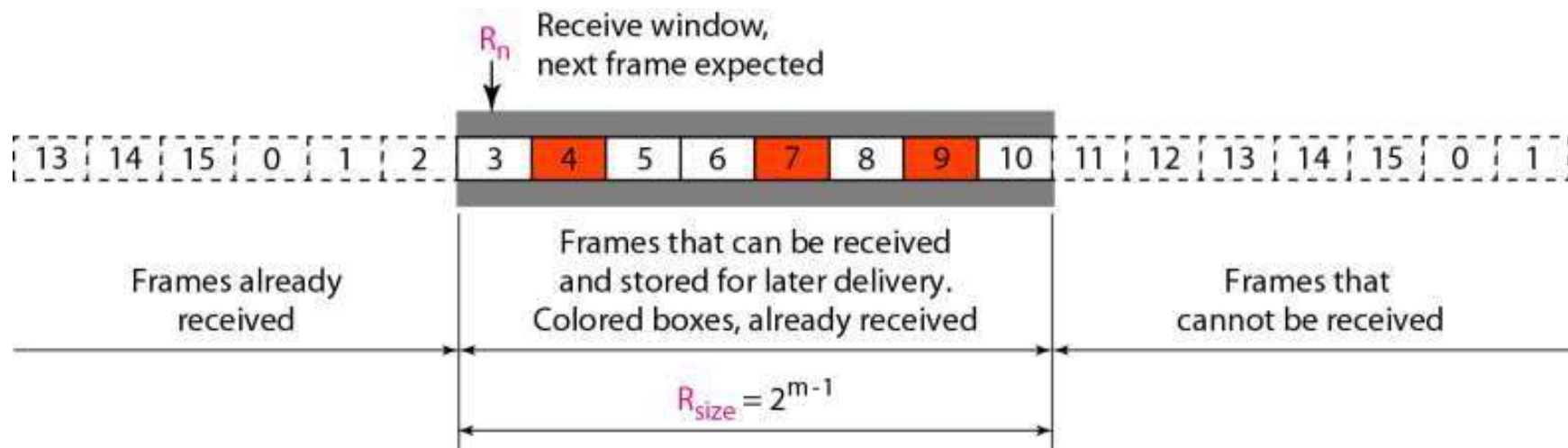
*Note*

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

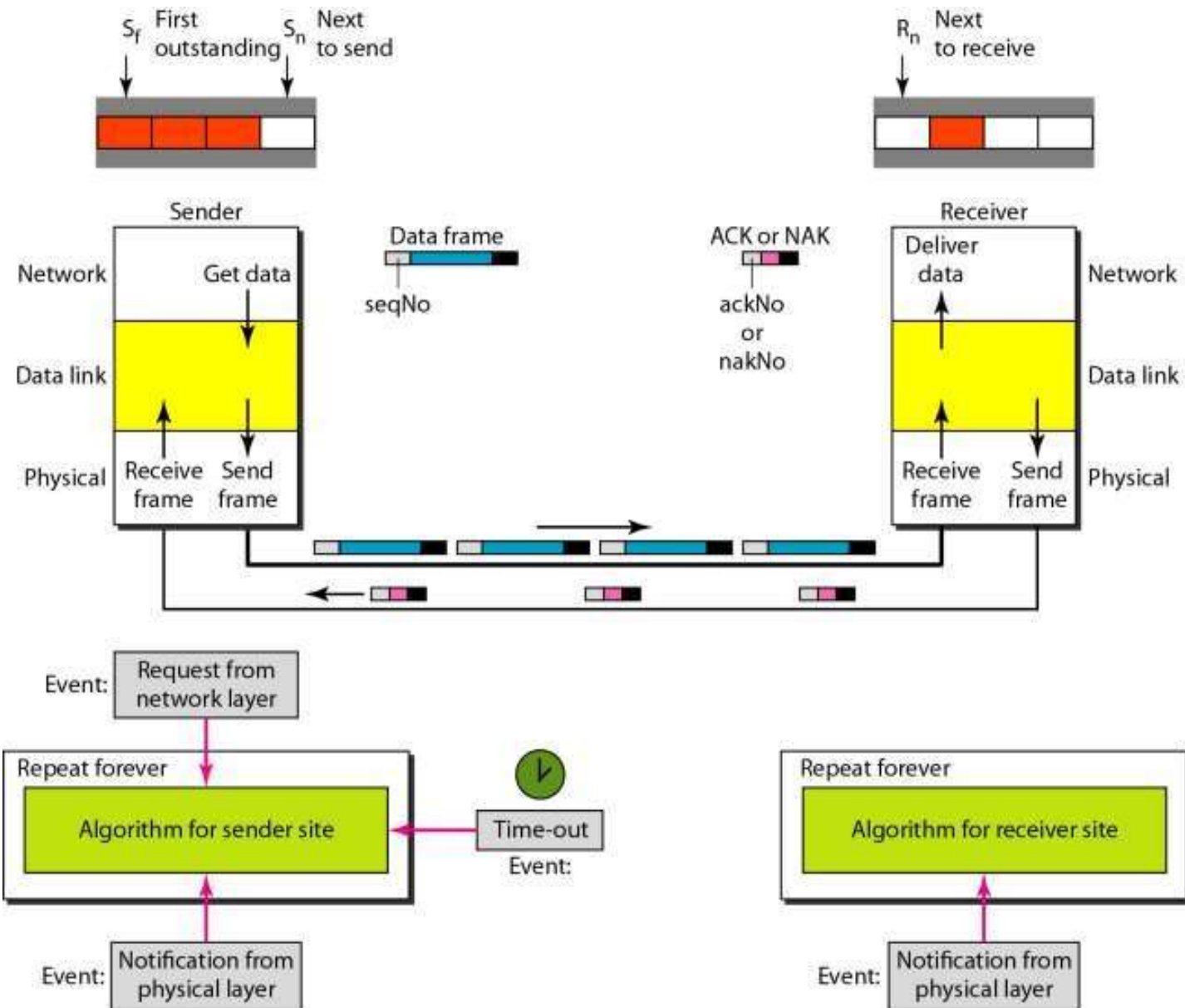
# Selective Repeat ARQ



# Receive window for *Selective Repeat ARQ*



**Figure 11.20** *Design of Selective Repeat ARQ*



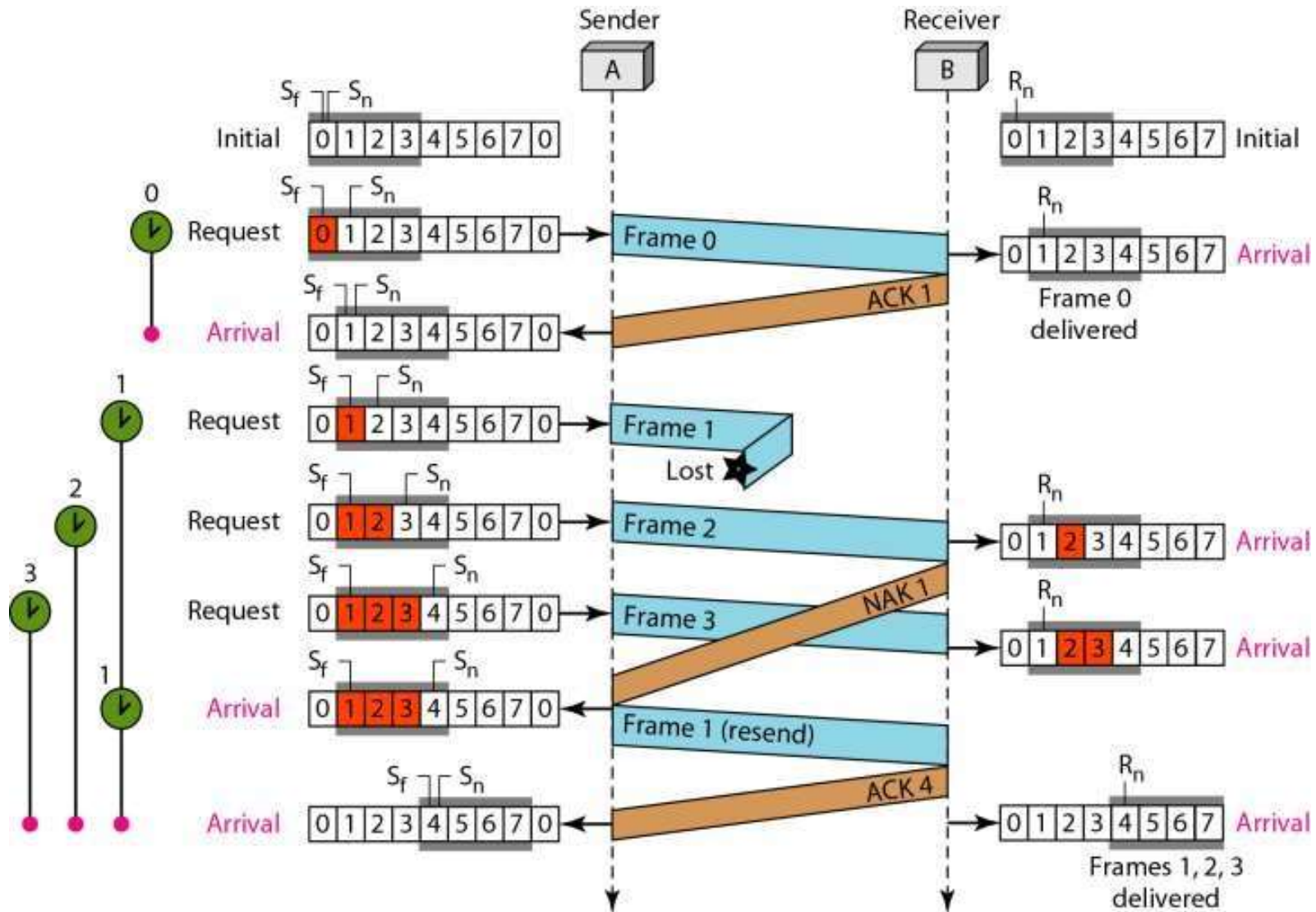


---

*Note*

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

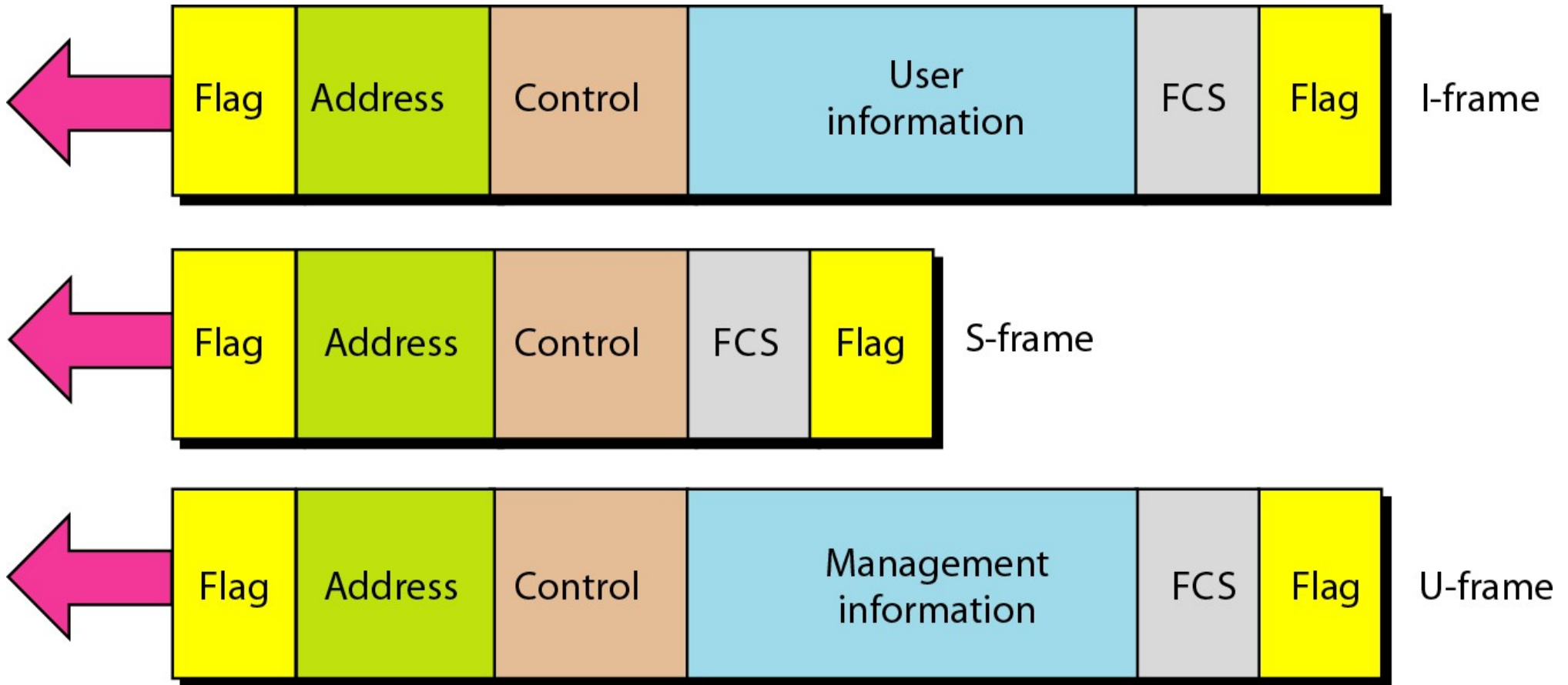
Figure *Flow diagram for Selective Repeat ARQ*



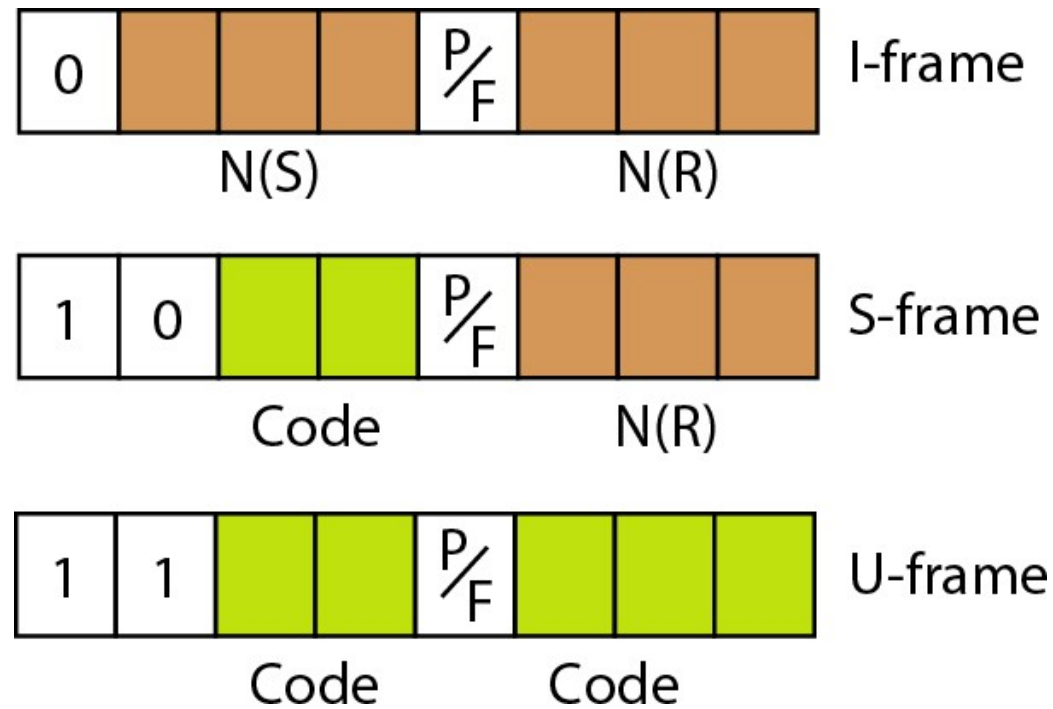
## 11-6 HDLC

*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*



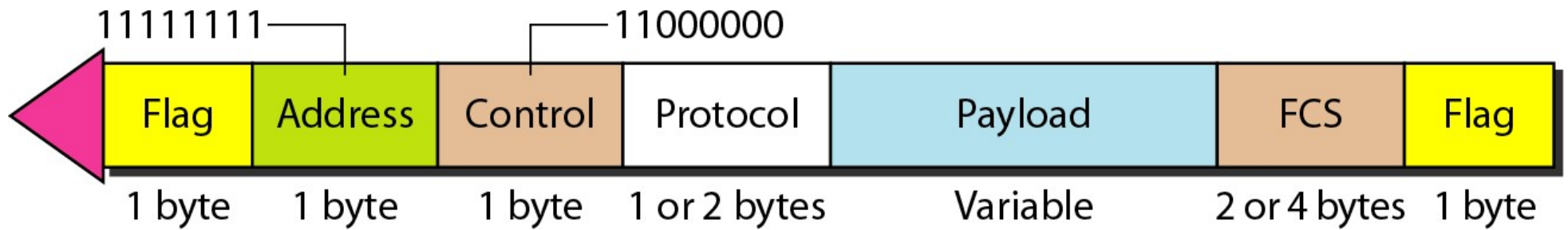
**Figure 11.28** *Control field format for the different frame types*



# POINT-TO-POINT PROTOCOL

*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*



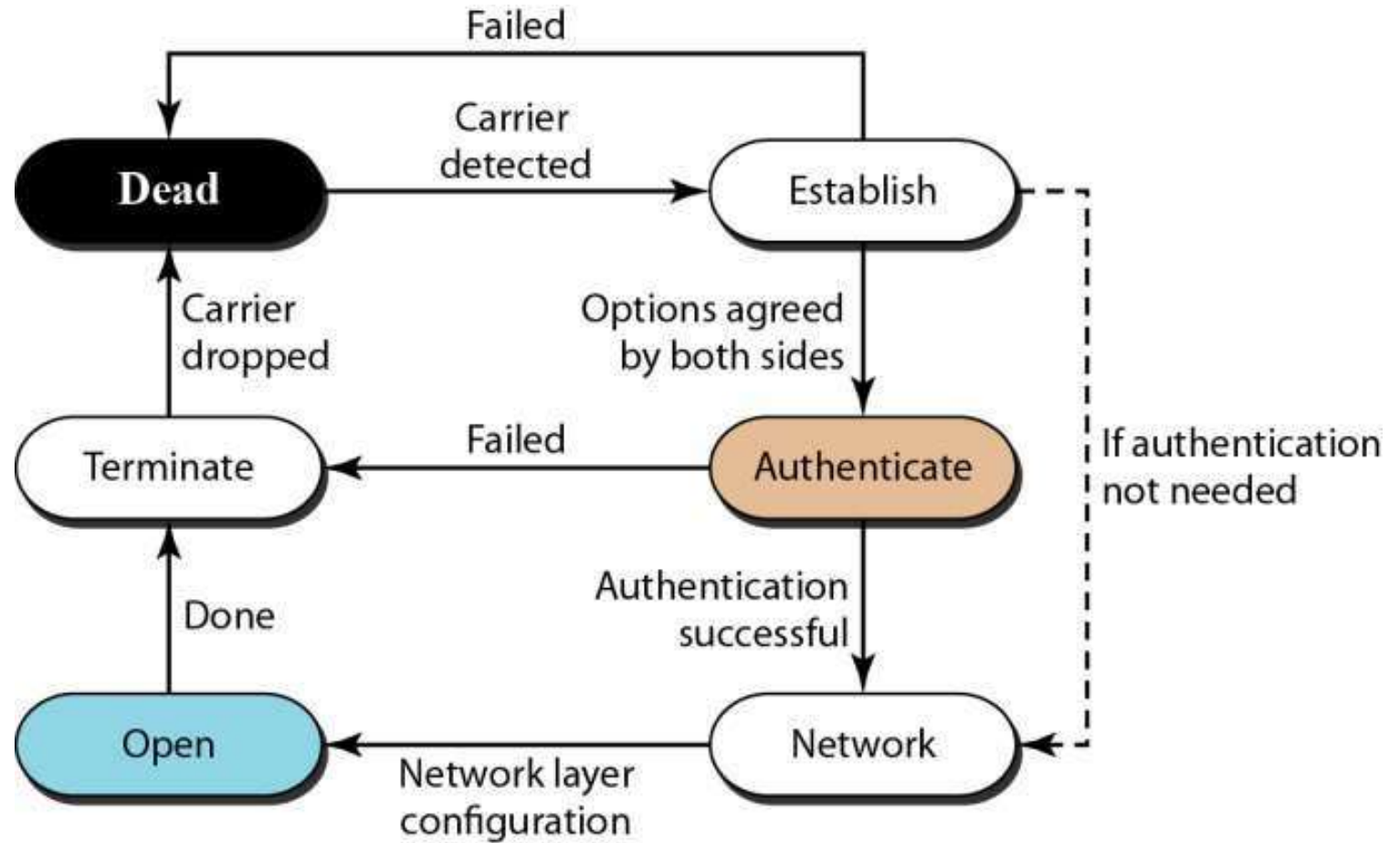


---

*Note*

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

## *Transition phases*





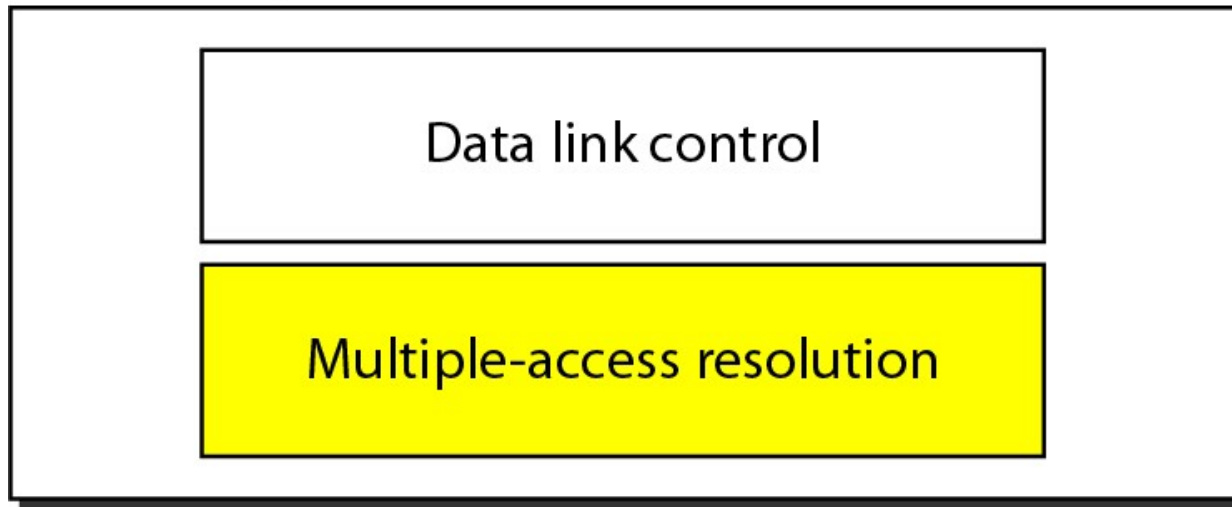
# Medium Access Control

---

## Data link layer divided into two functionality-oriented sublayers

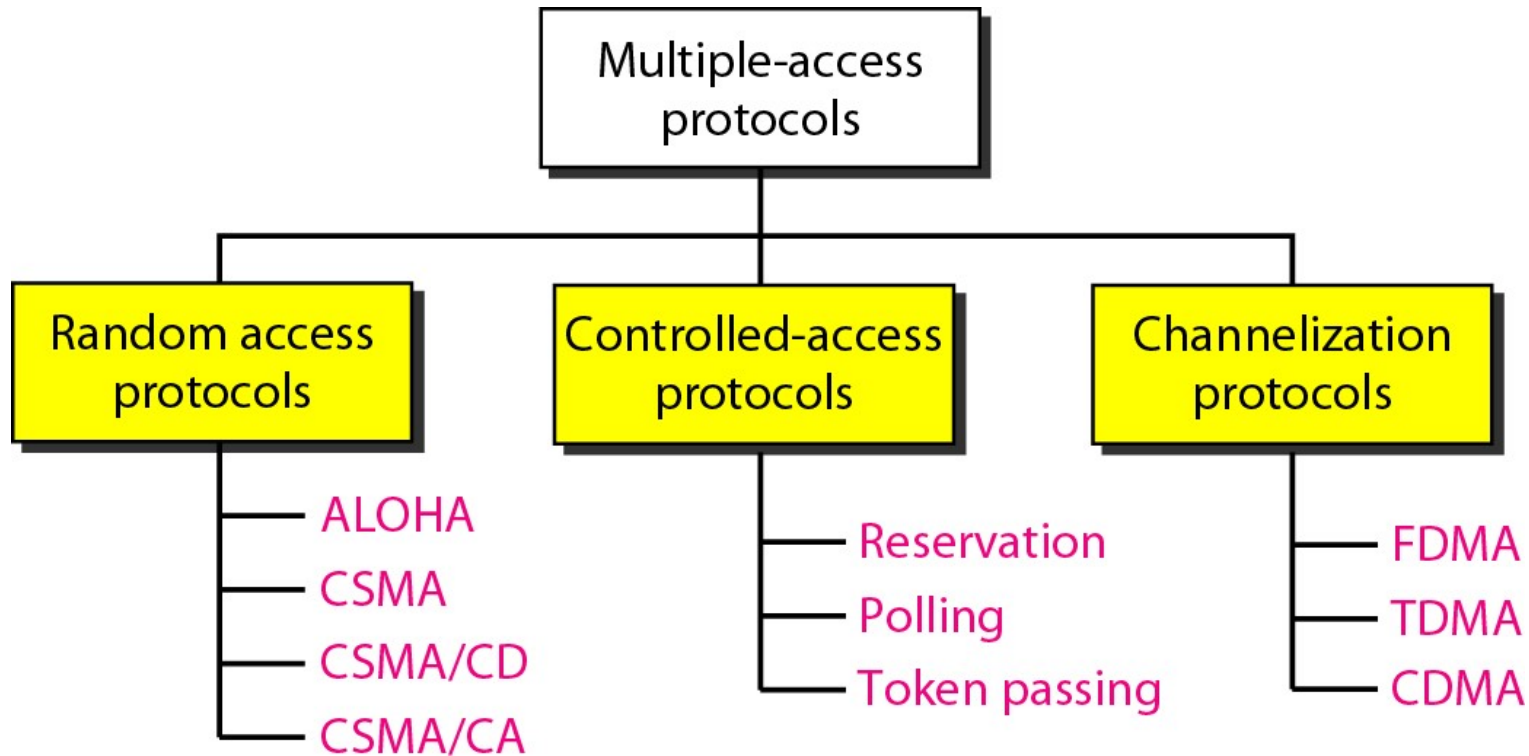
---

Data link layer



# Taxonomy of multiple-access protocols discussed in this chapter

---



# 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.

Topics discussed in this section:

ALOHA

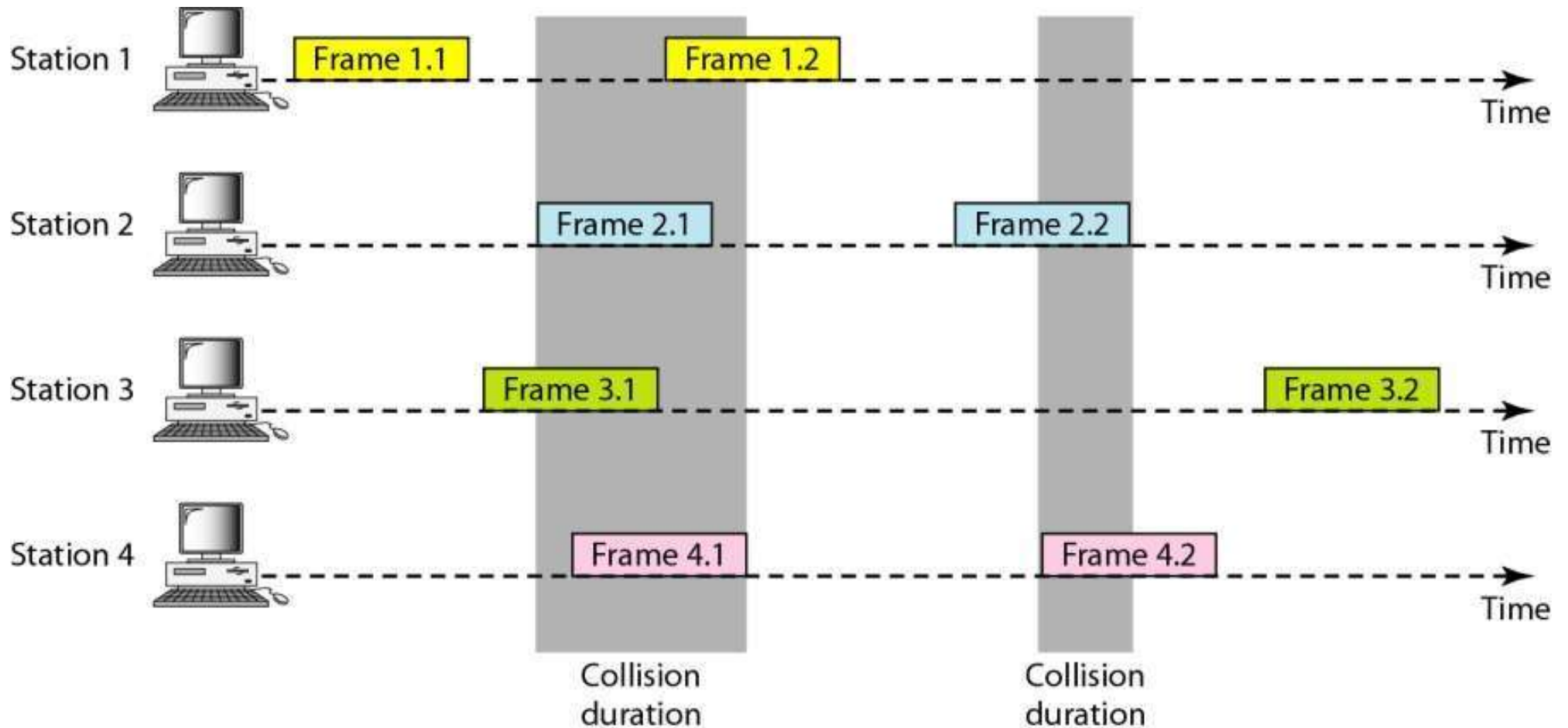
Carrier Sense Multiple Access

Carrier Sense Multiple Access with Collision Detection

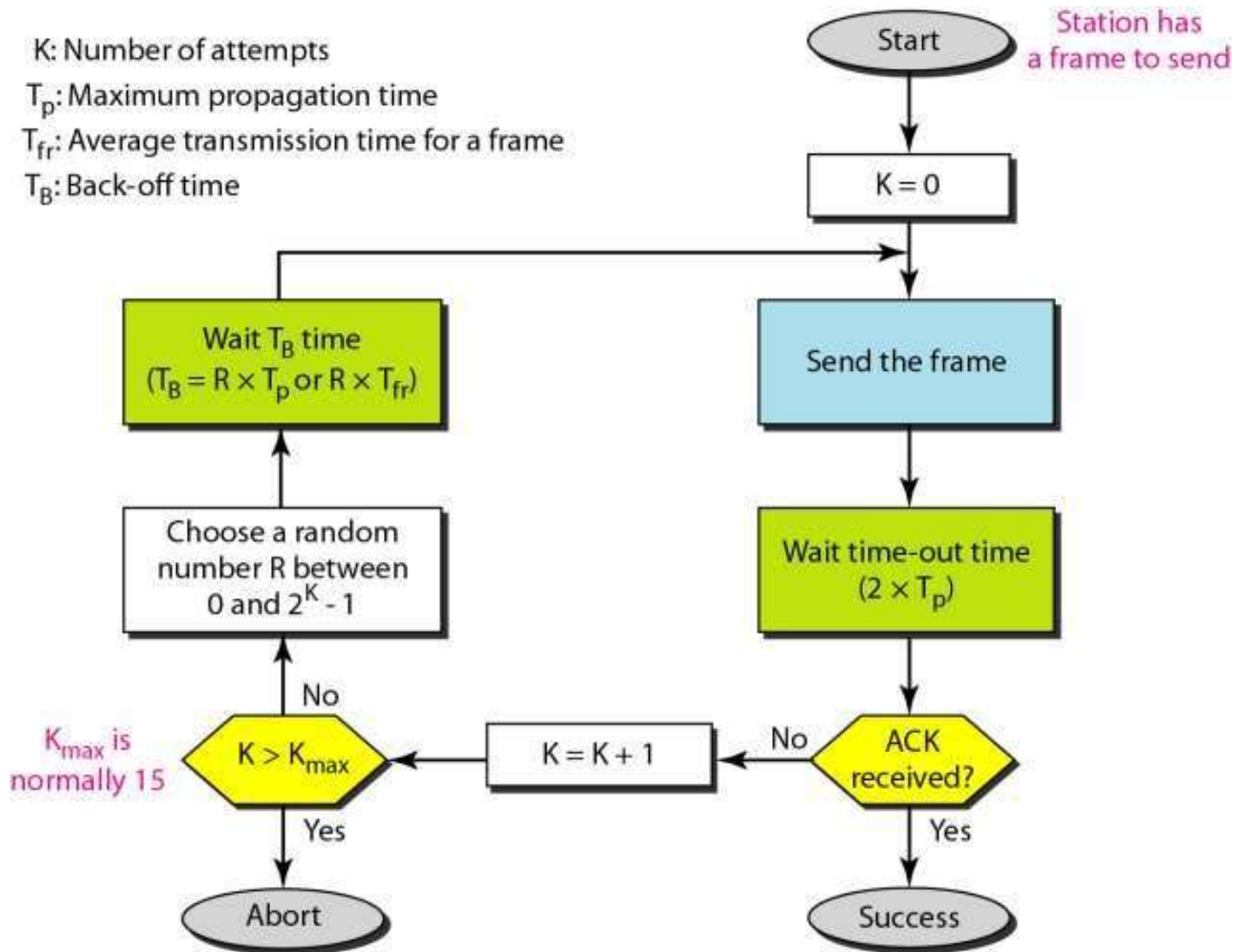
Carrier Sense Multiple Access with Collision Avoidance

# Frames in a pure ALOHA network

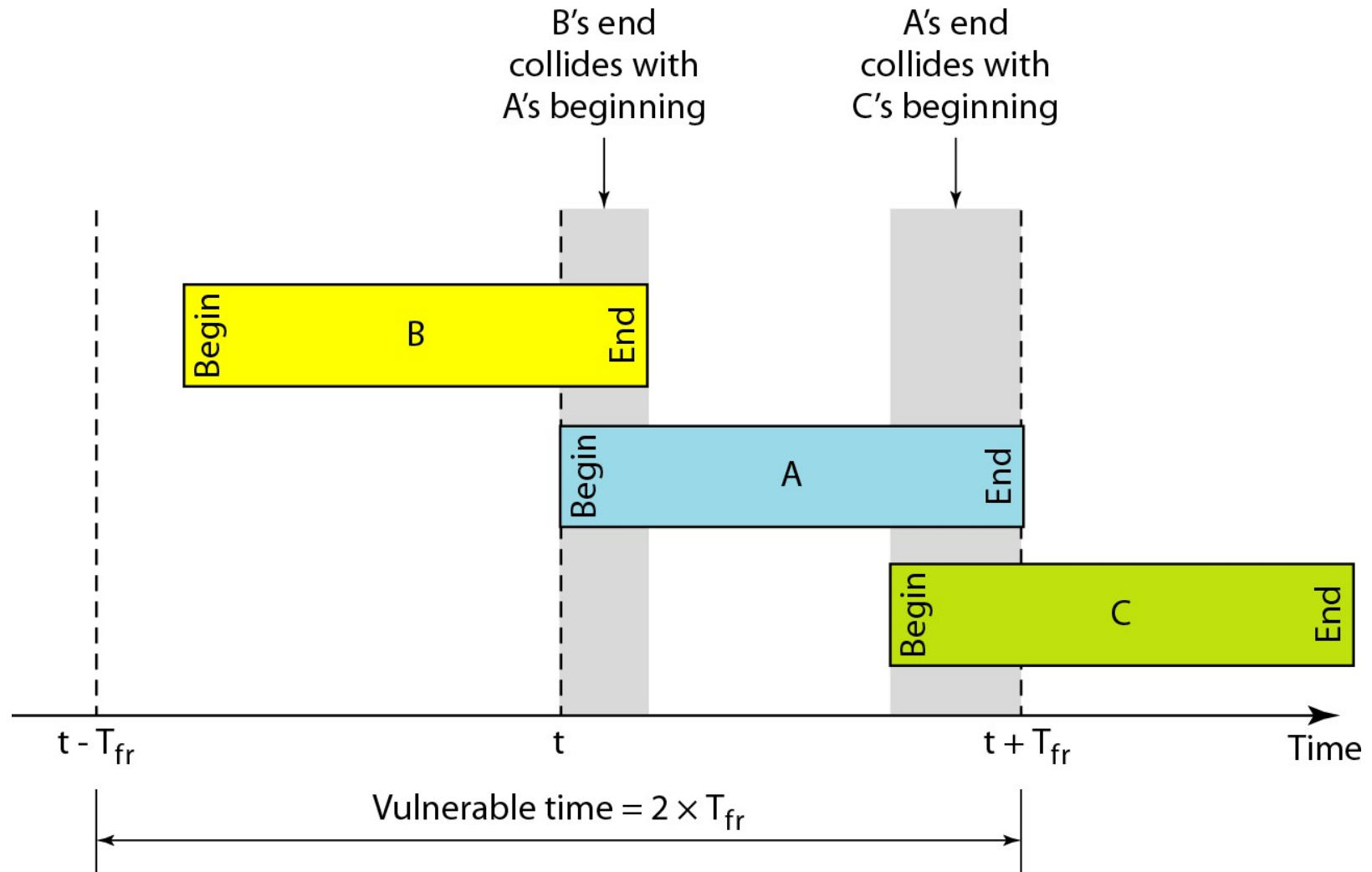
---



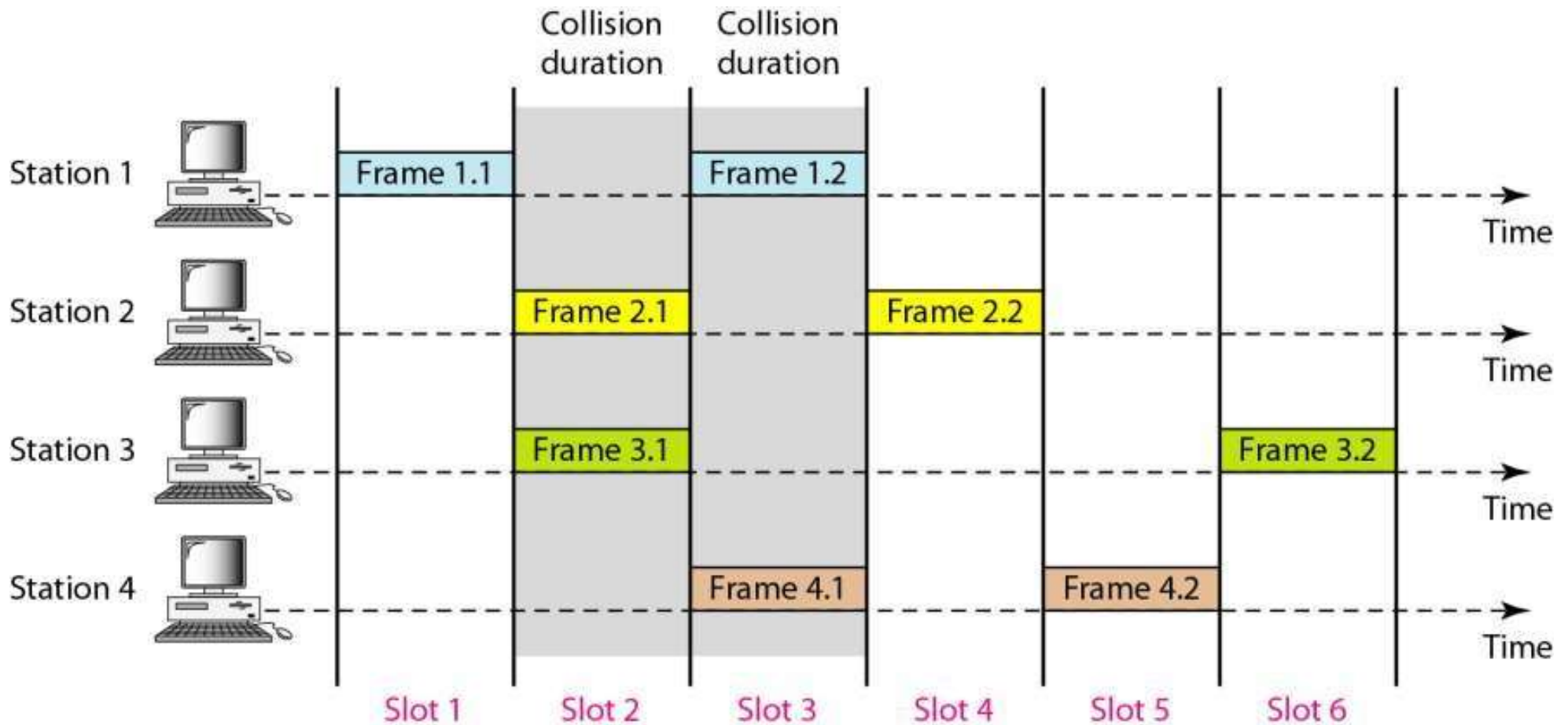
# Procedure for pure ALOHA protocol



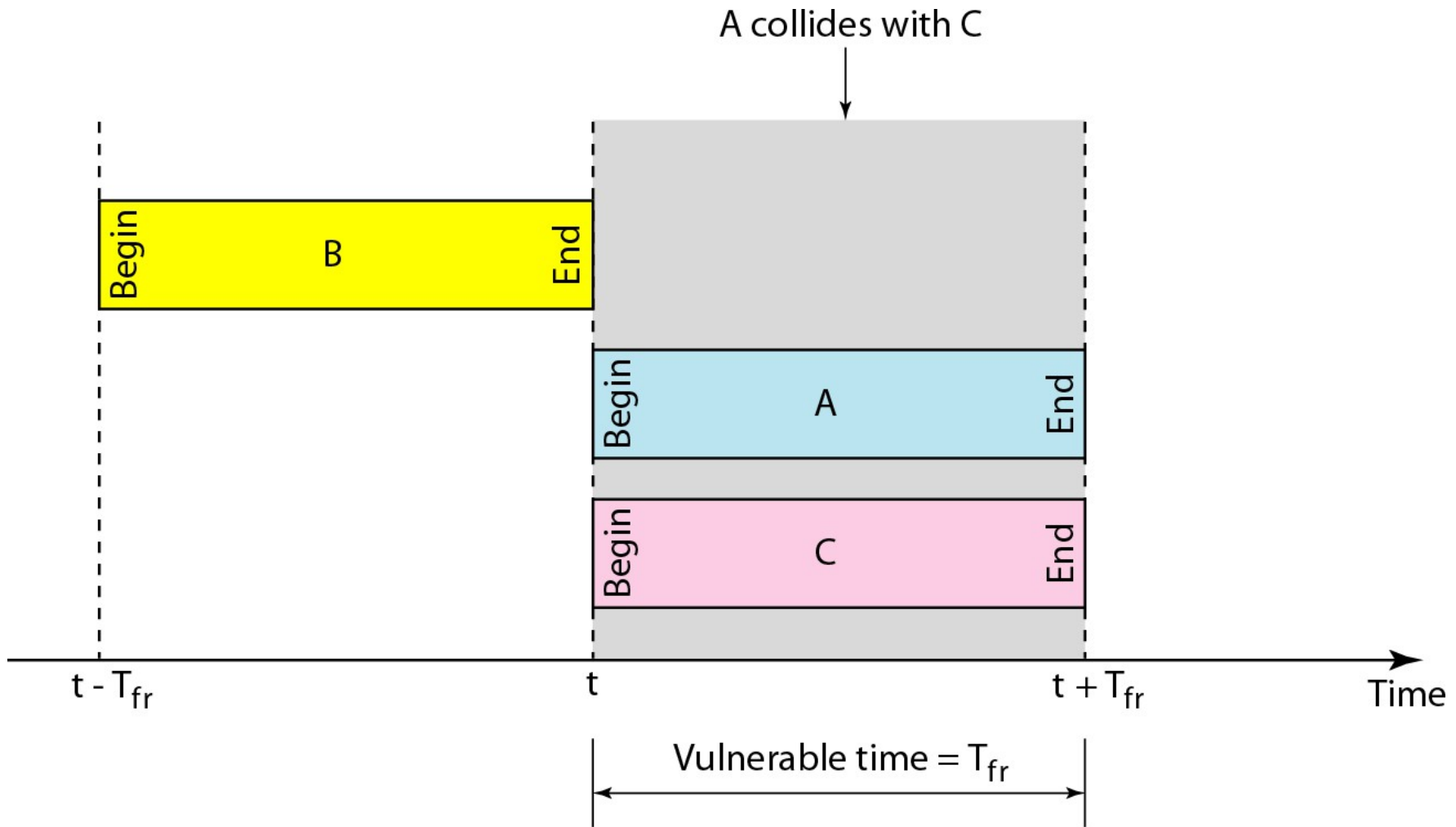
# Vulnerable time for pure ALOHA protocol



# Frames in a slotted ALOHA network



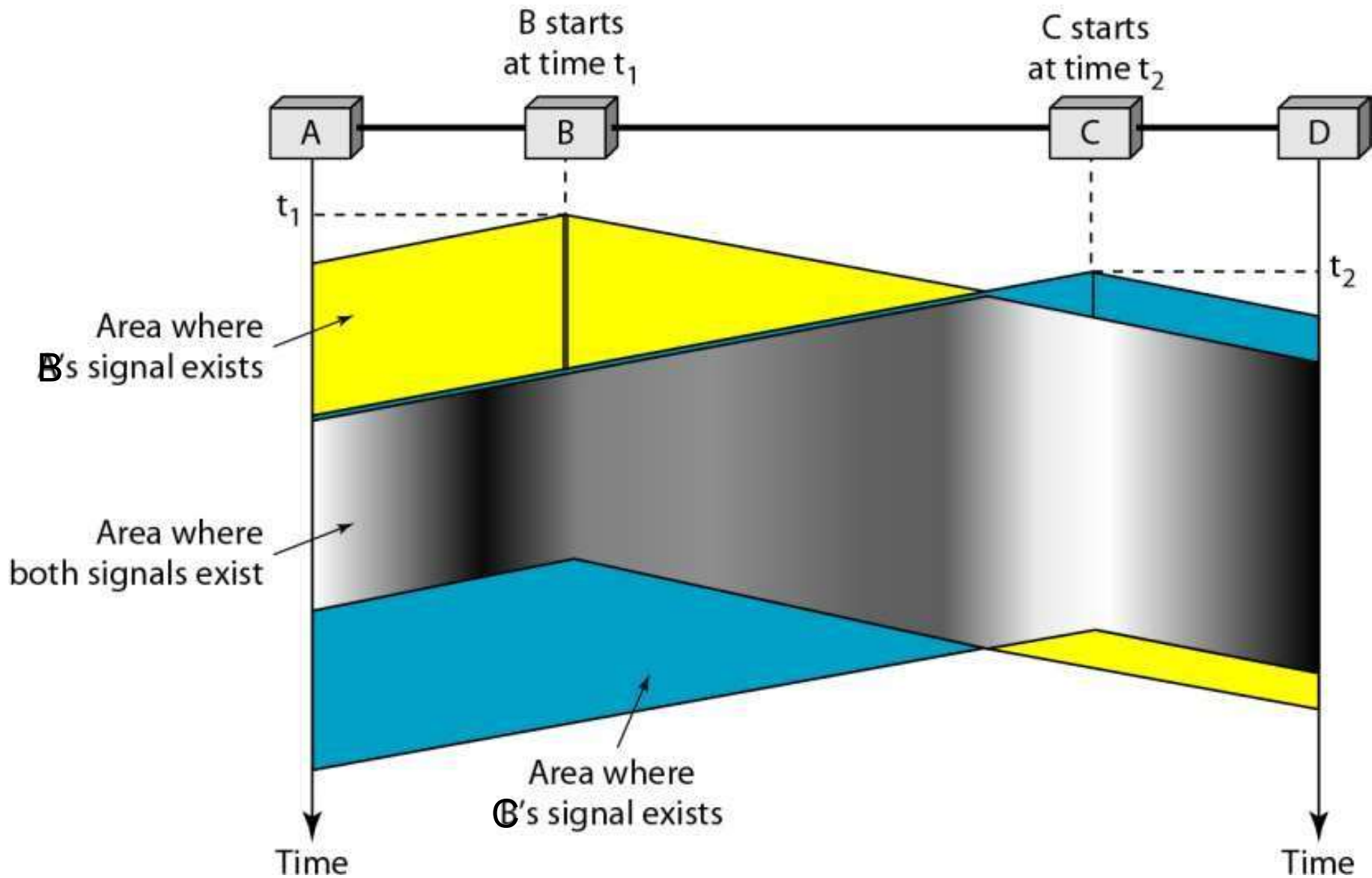
# Vulnerable time for slotted ALOHA protocol



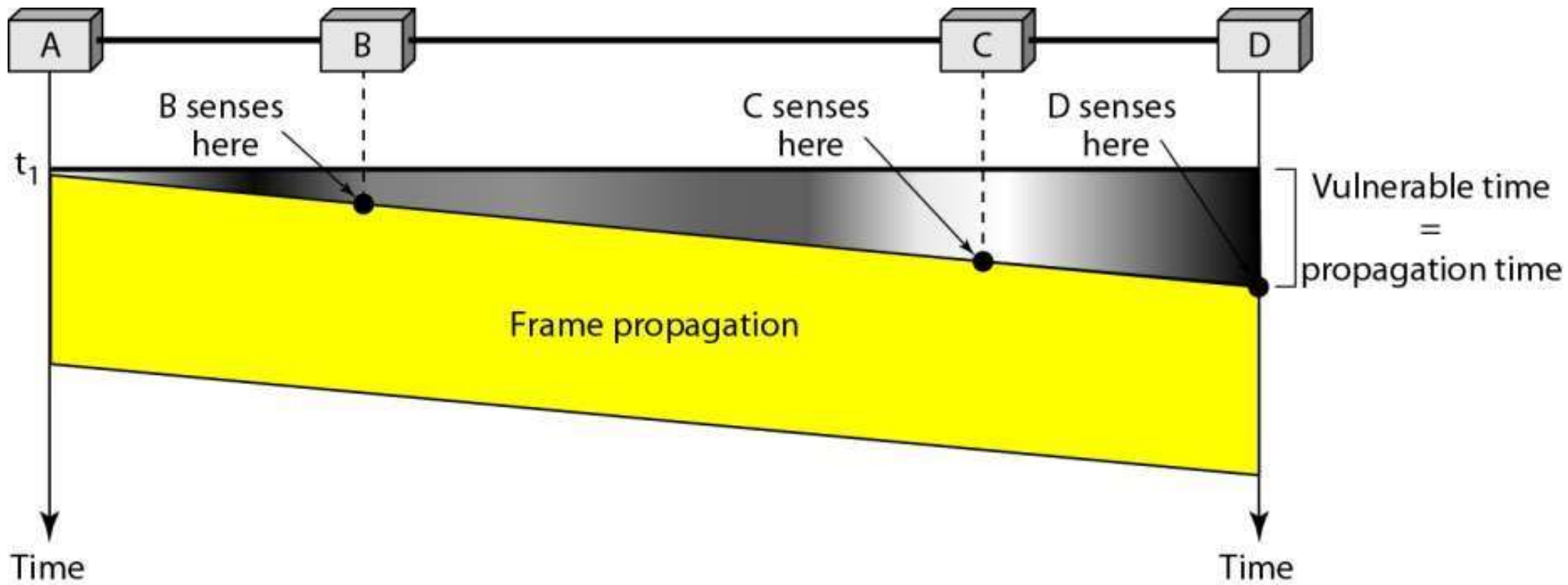
CSMA/CD

Carrier Sense Multiple Access with  
Collision Detection

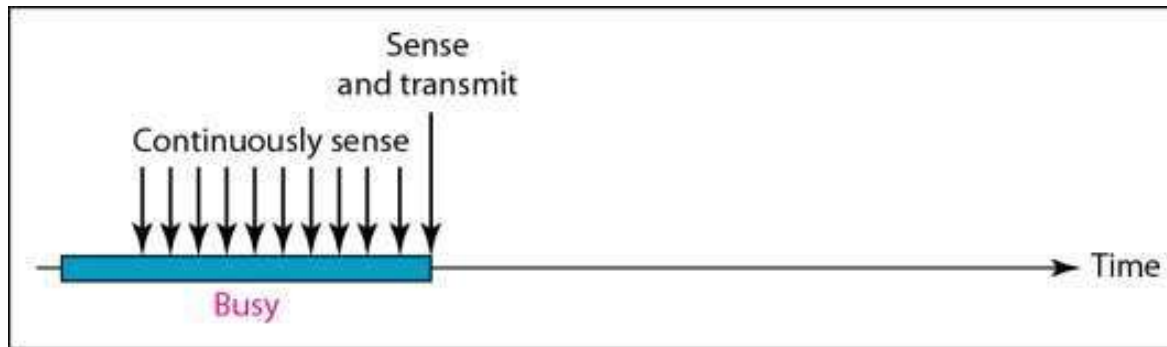
# Space/time model of the collision in CSMA



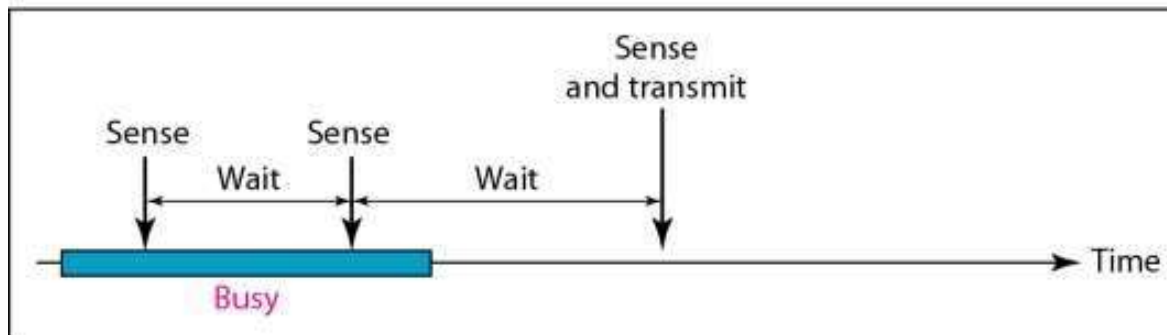
# Vulnerable time in CSMA



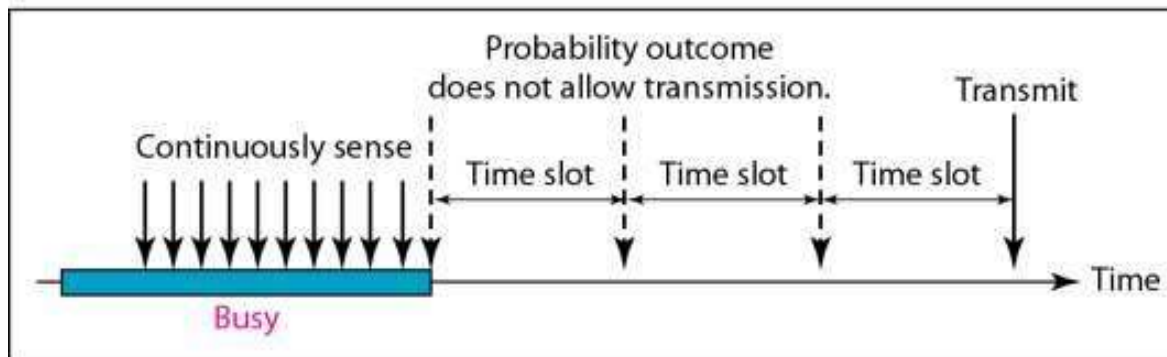
# Behavior of three persistence methods



a. 1-persistent

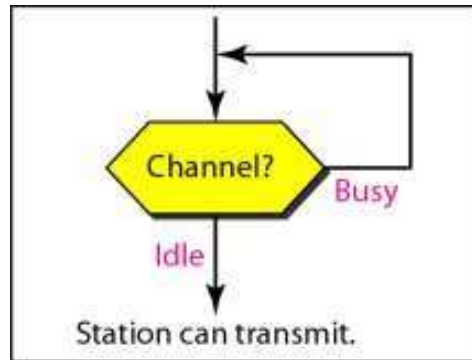


b. Nonpersistent

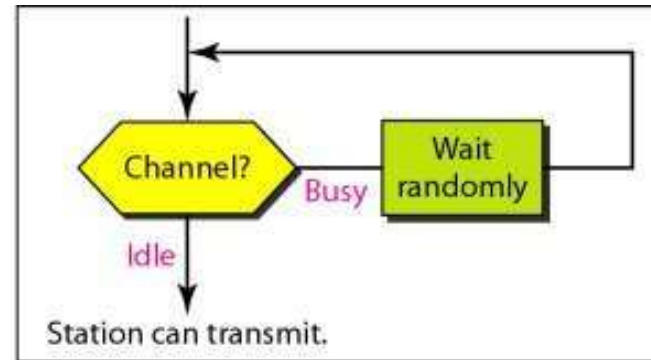


c. p-persistent

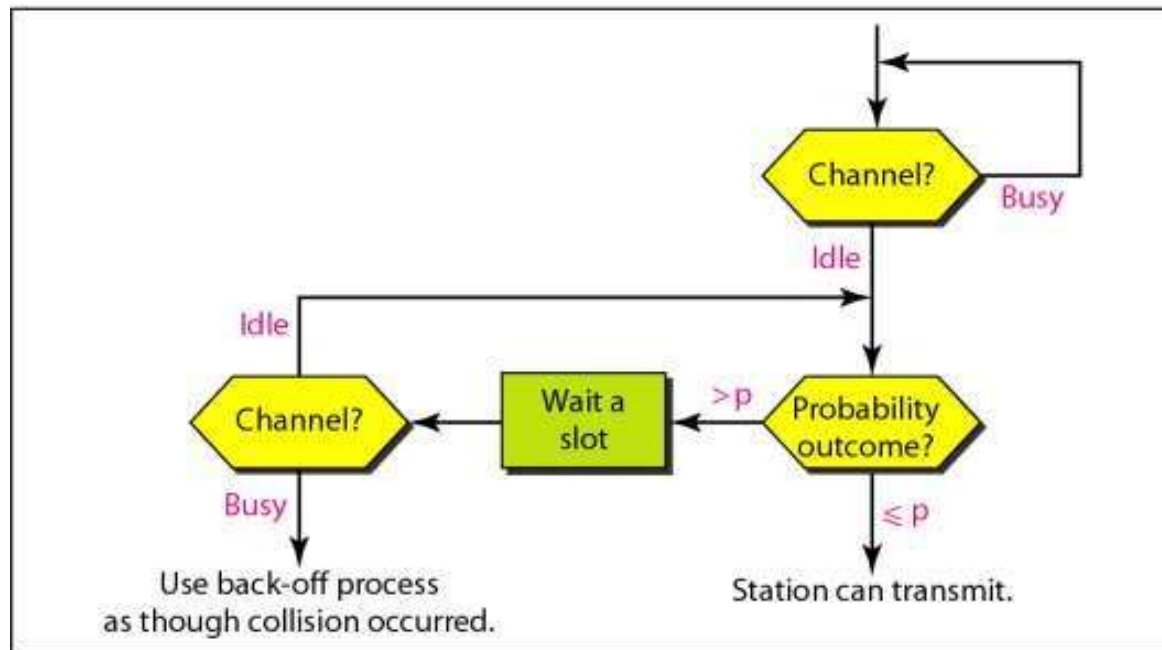
# Flow diagram for three persistence methods



a. 1-persistent

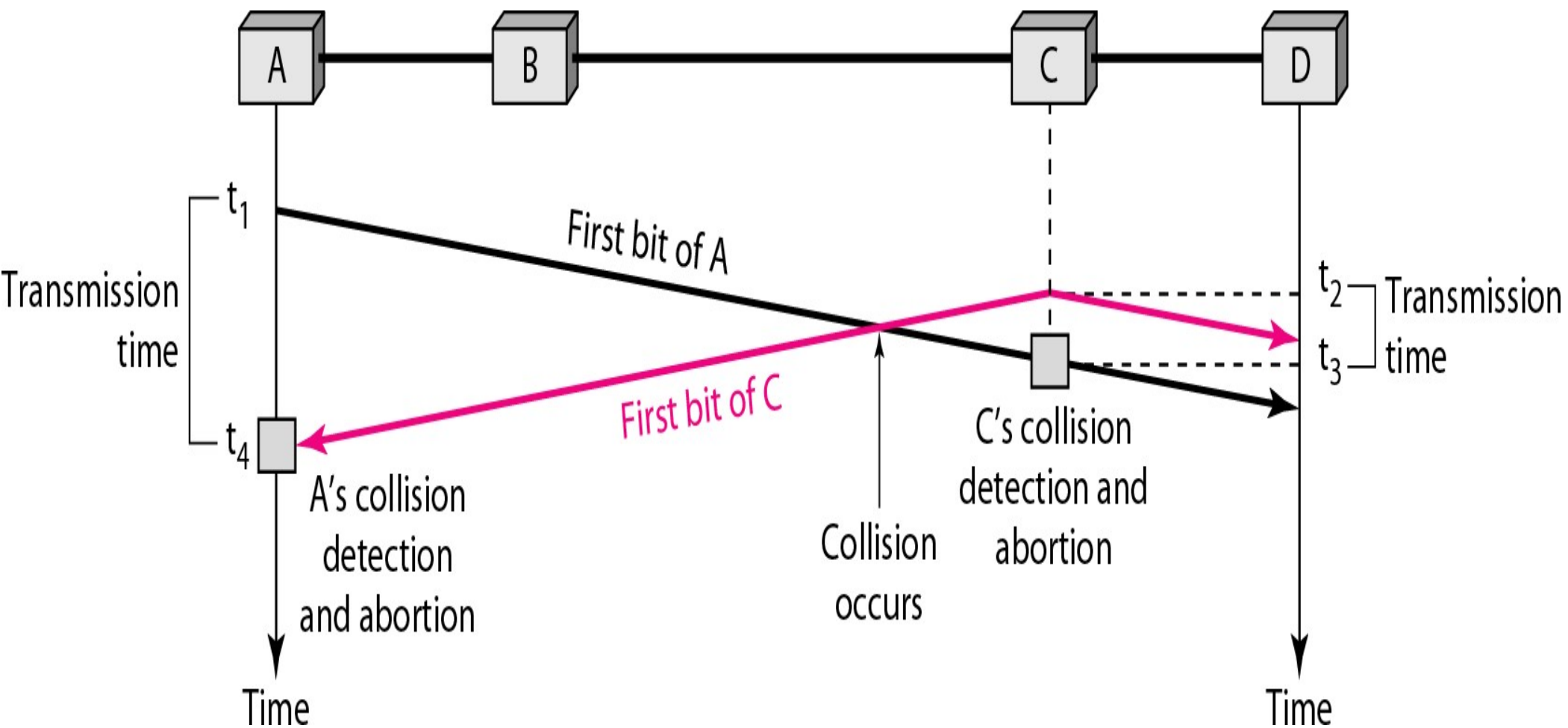


b. Nonpersistent

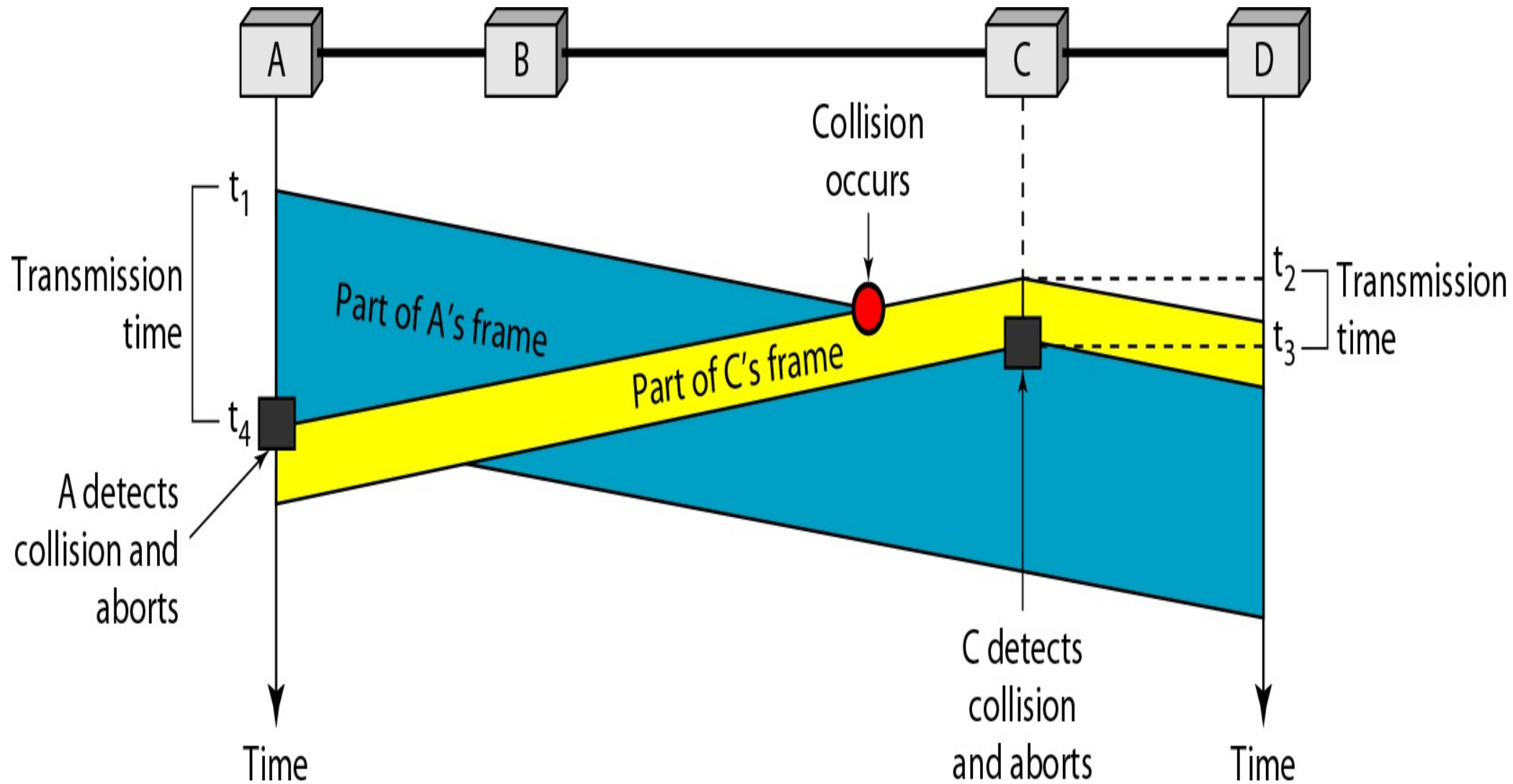


c. p-persistent

# Collision of the first bit in CSMA/CD



# Collision and abortion in CSMA/CD



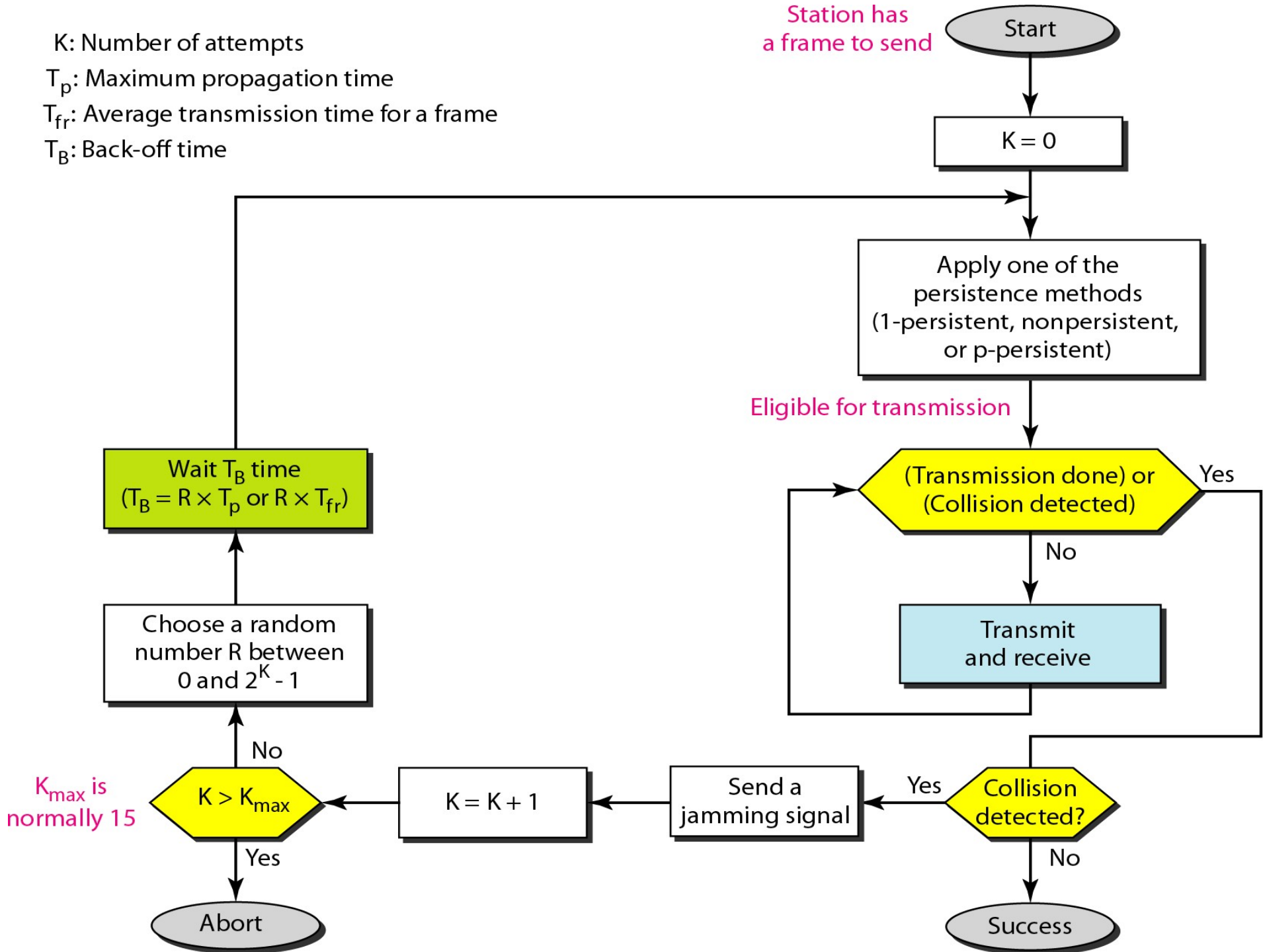
# Flow diagram for the CSMA/CD

K: Number of attempts

$T_p$ : Maximum propagation time

$T_{fr}$ : Average transmission time for a frame

$T_B$ : Back-off time

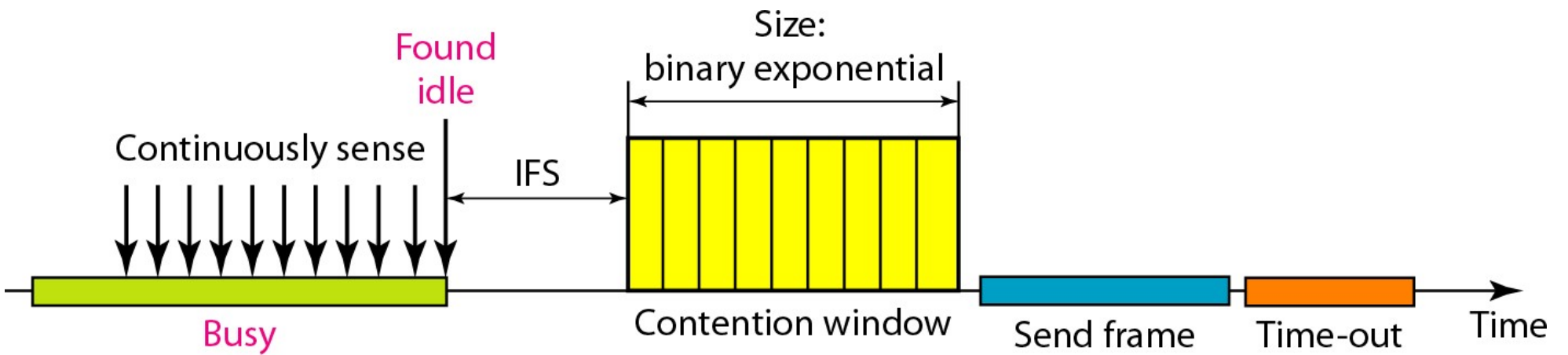


CSMA/CA

Carrier Sense Multiple Access with  
Collision Avoidance

# Timing in CSMA/CA

---



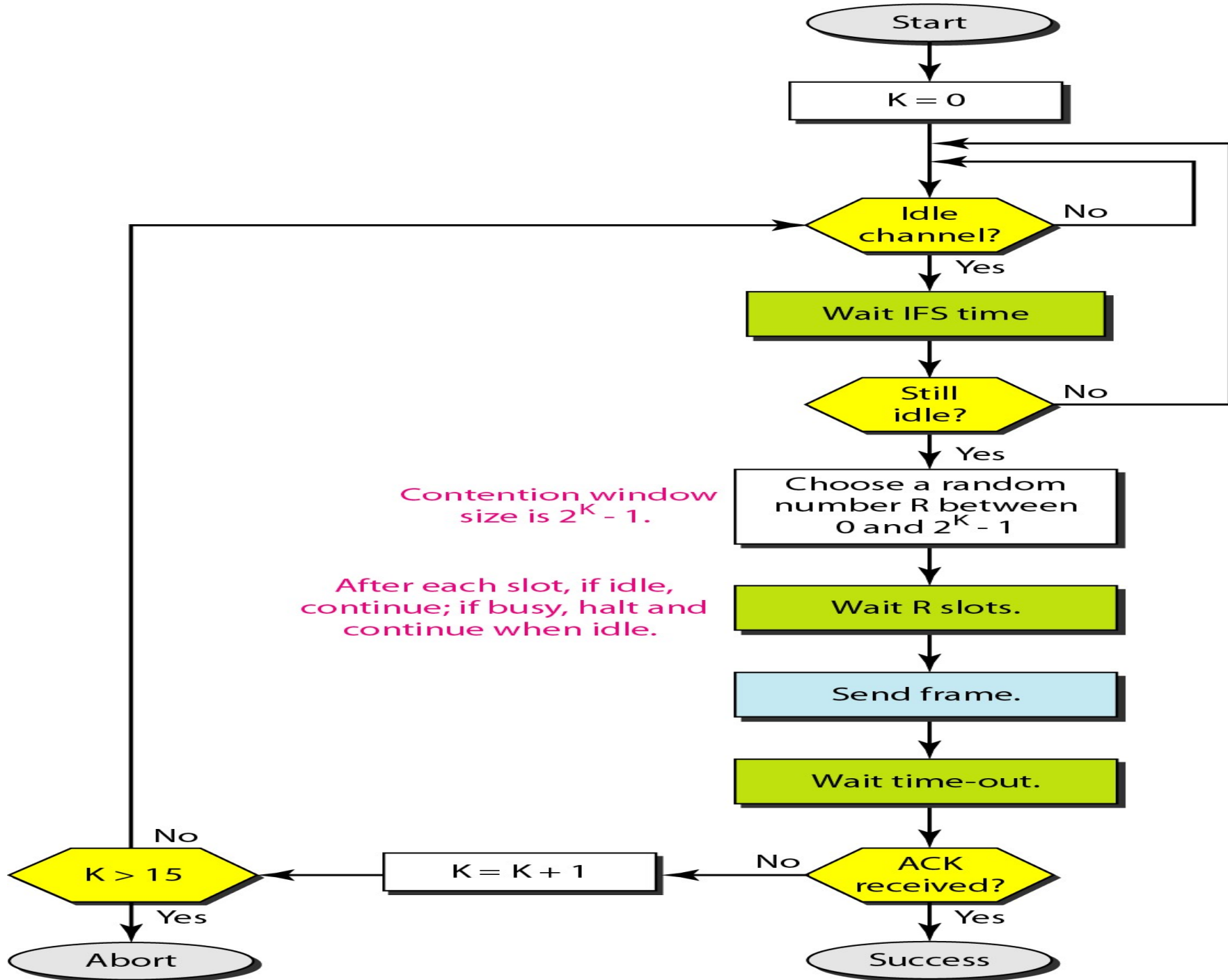


---

Note

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

# Flow diagram for CSMA/CA



# 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.

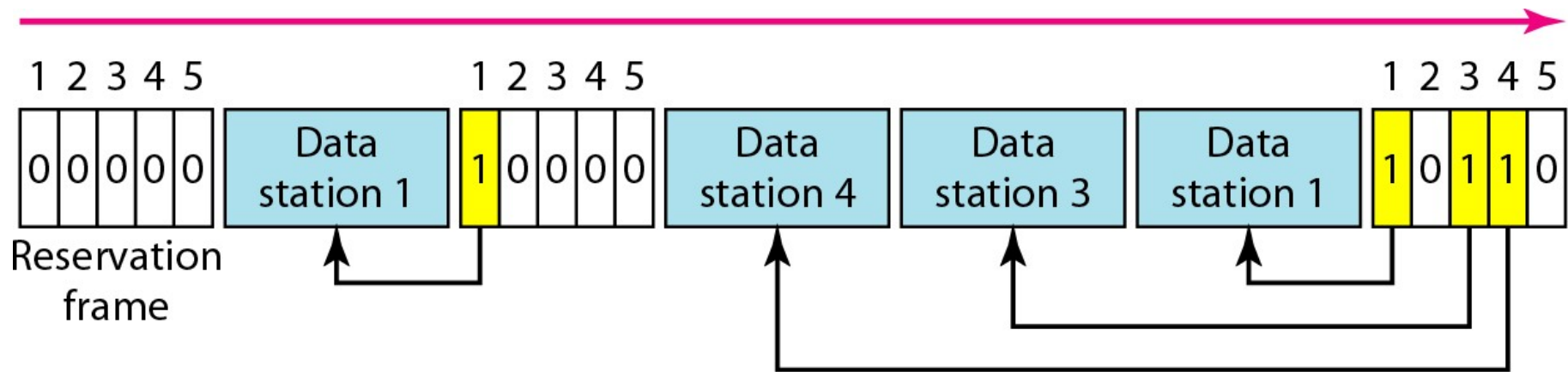
Topics discussed in this section:

Reservation

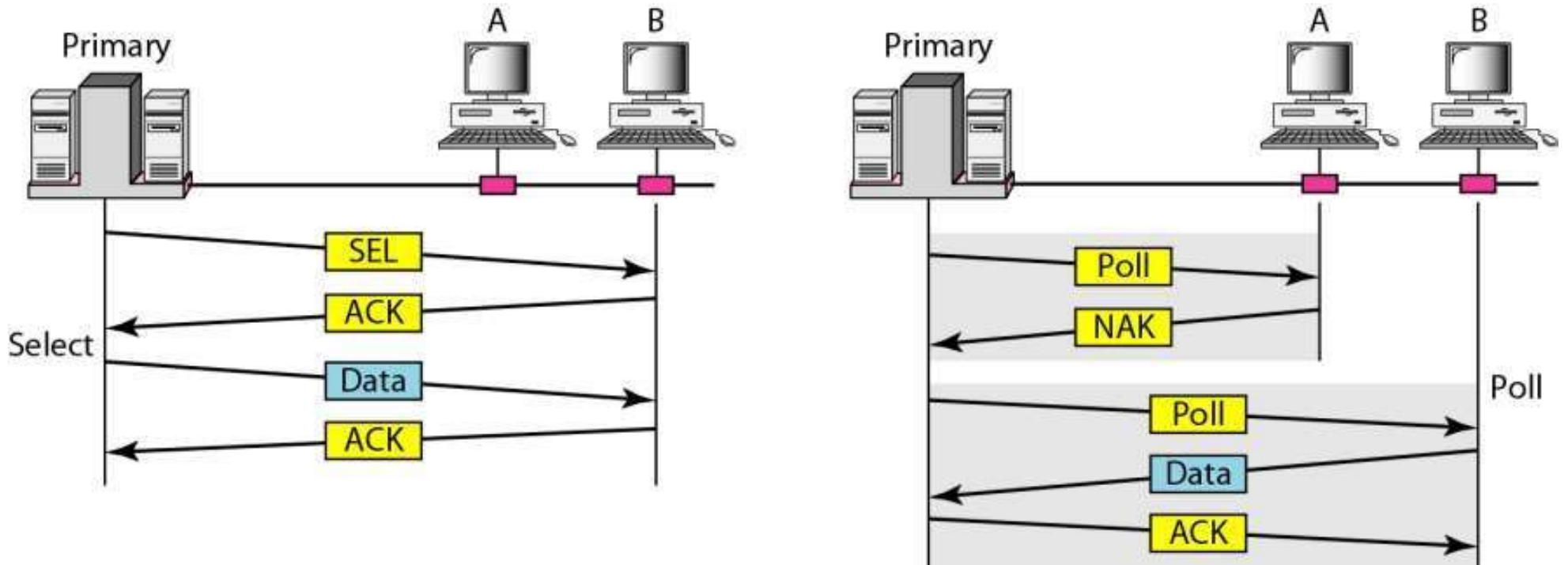
Polling

Token Passing

# Reservation access method



# polling

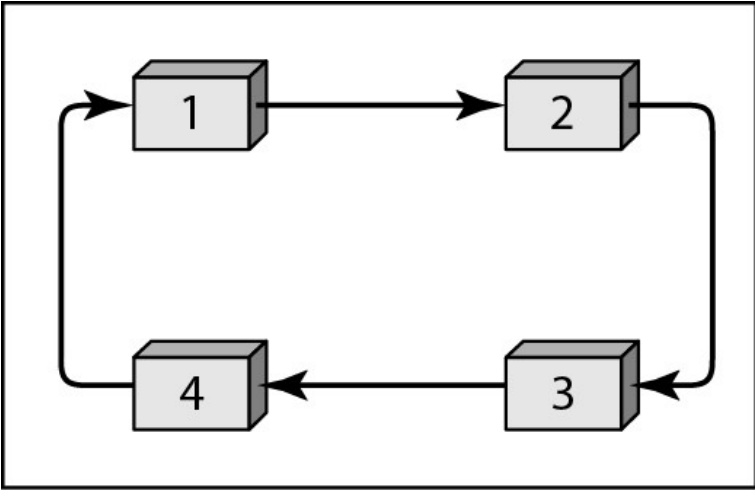


**Polling** – one device as primary station and the other device as secondary station

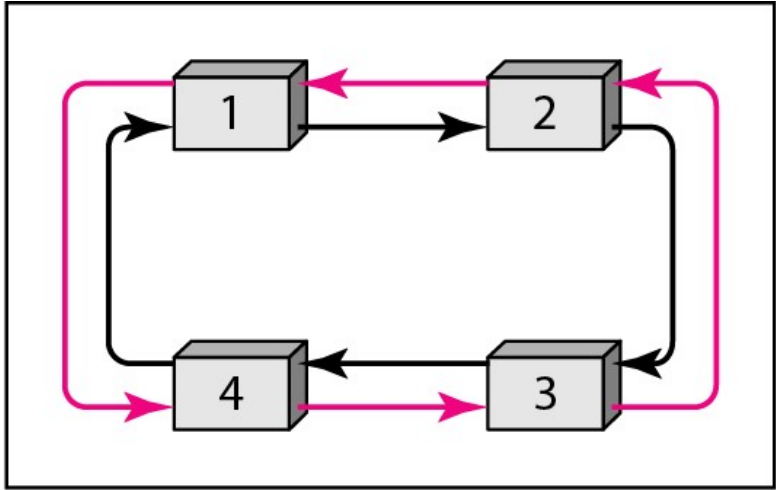
**Select** – primary device wants to send data to secondary device, secondary device gets ready to receive

**Poll** – primary device solicits (ask) transmissions from secondary devices

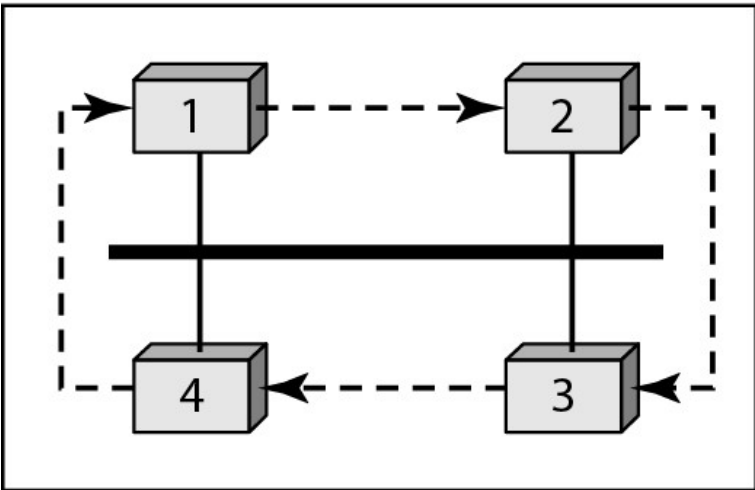
Logical ring and physical topology in **token-passing** access method



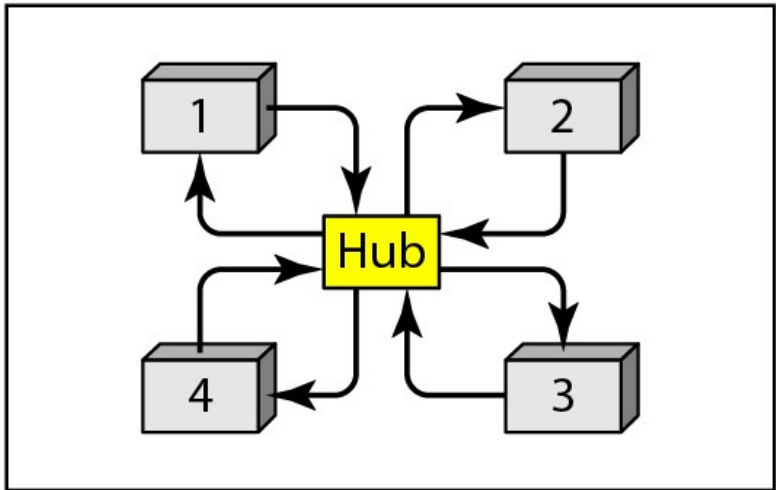
a. Physical ring



b. Dual ring



c. Bus ring



d. Star ring

# CHANNELIZATION

**Channelization** 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:

Frequency-Division Multiple Access (FDMA)

Time-Division Multiple Access (TDMA)

Code-Division Multiple Access (CDMA)

# 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



---

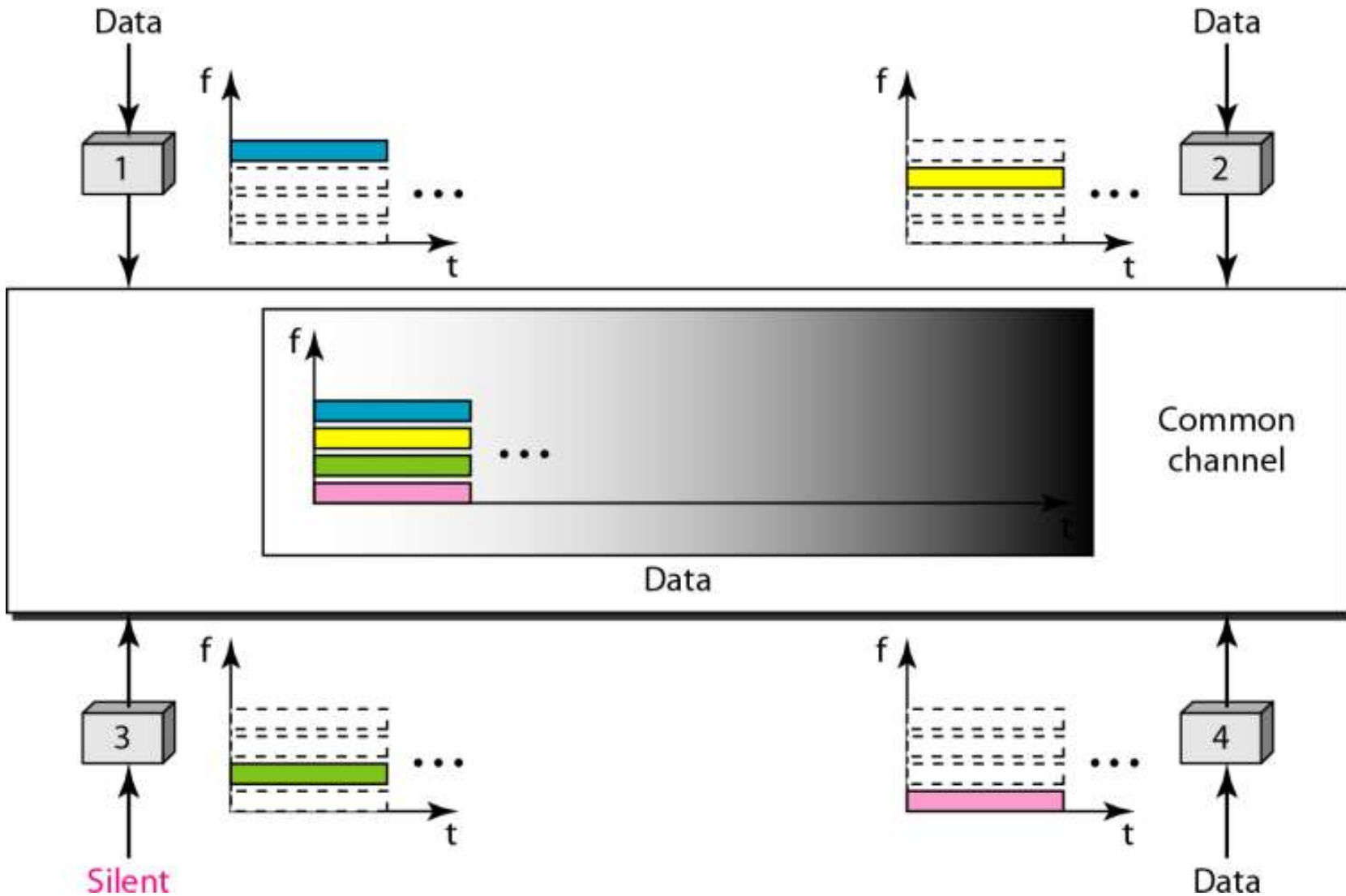
Note

---

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

- **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

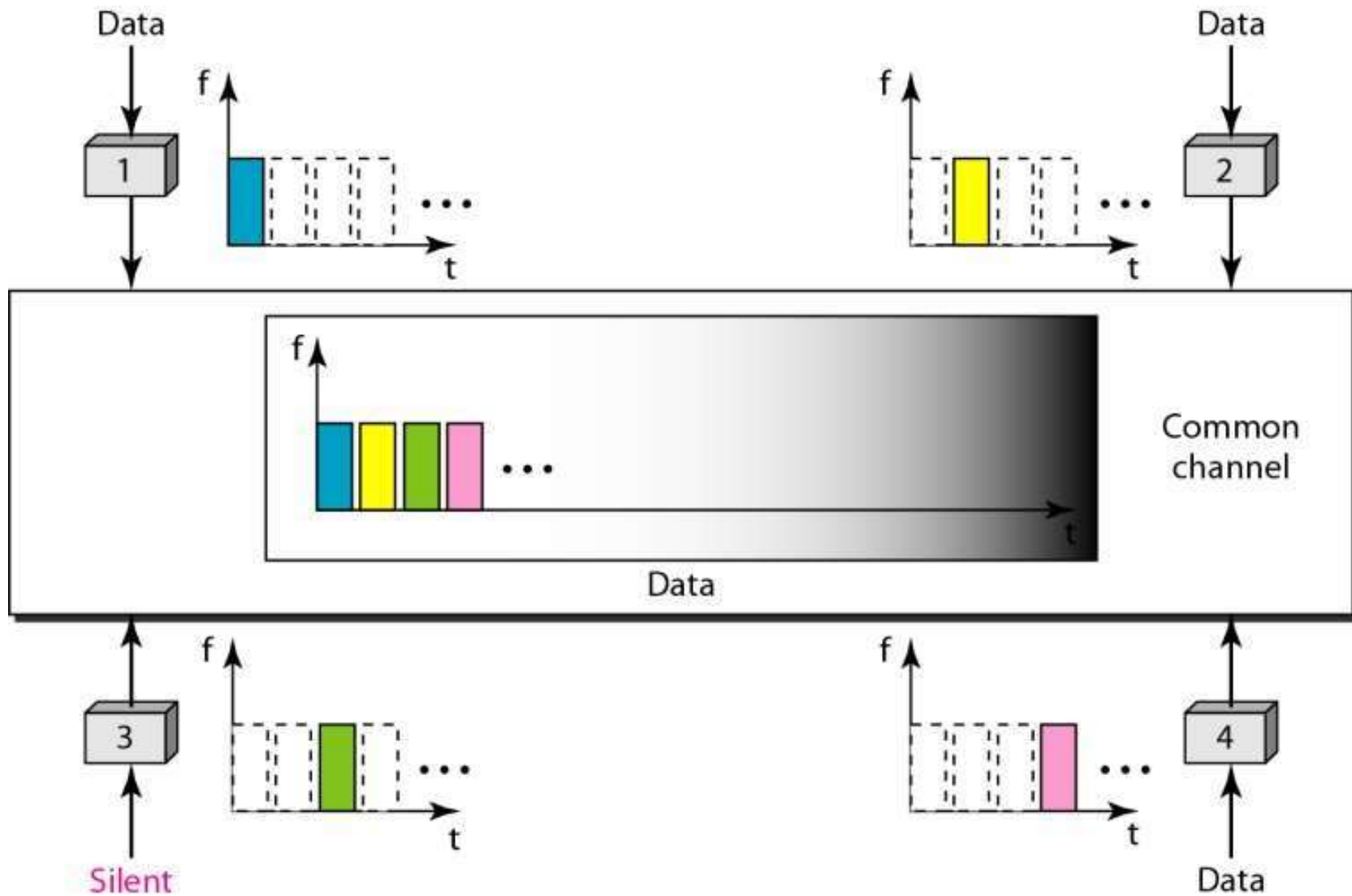
---

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, one channel 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

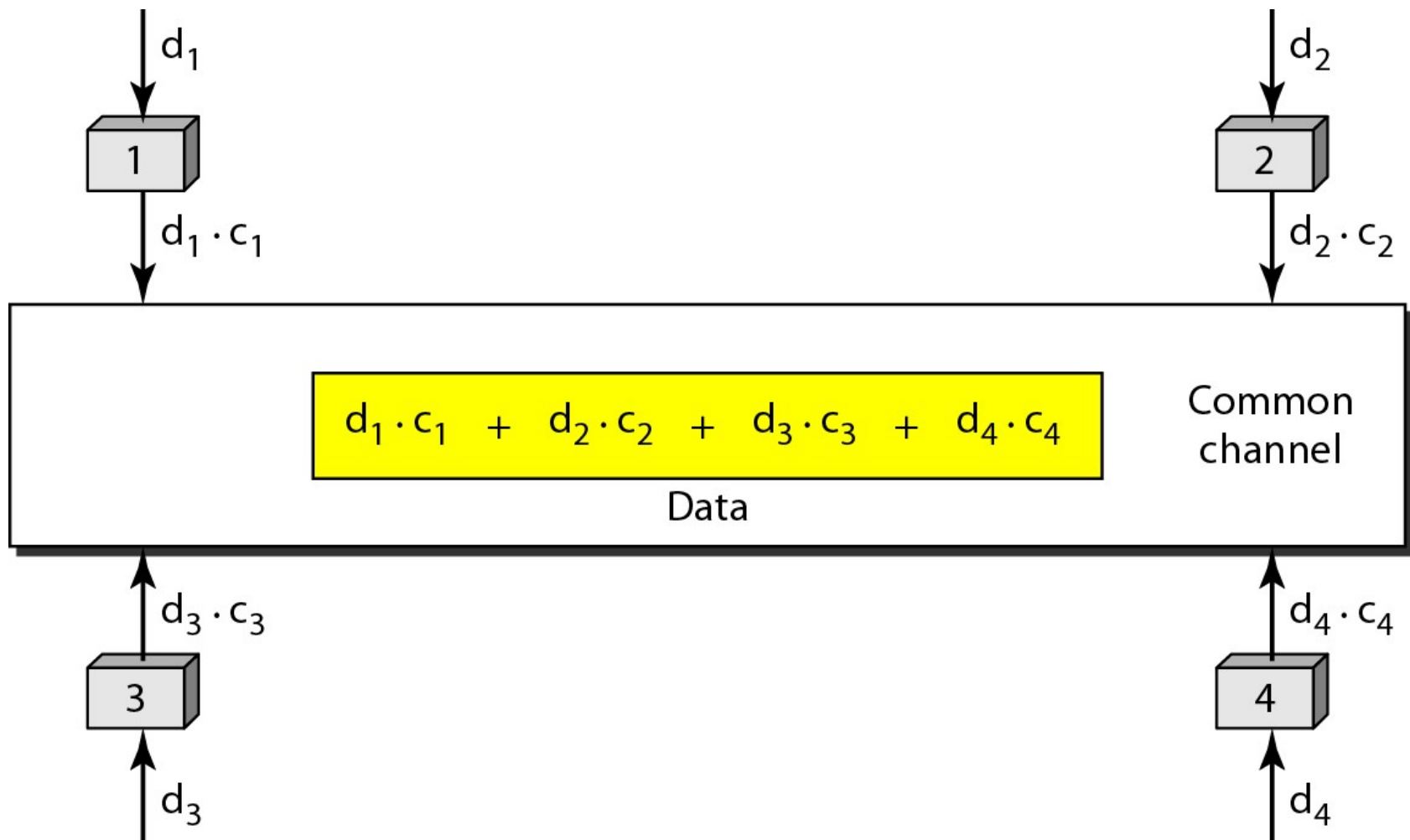
---

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]

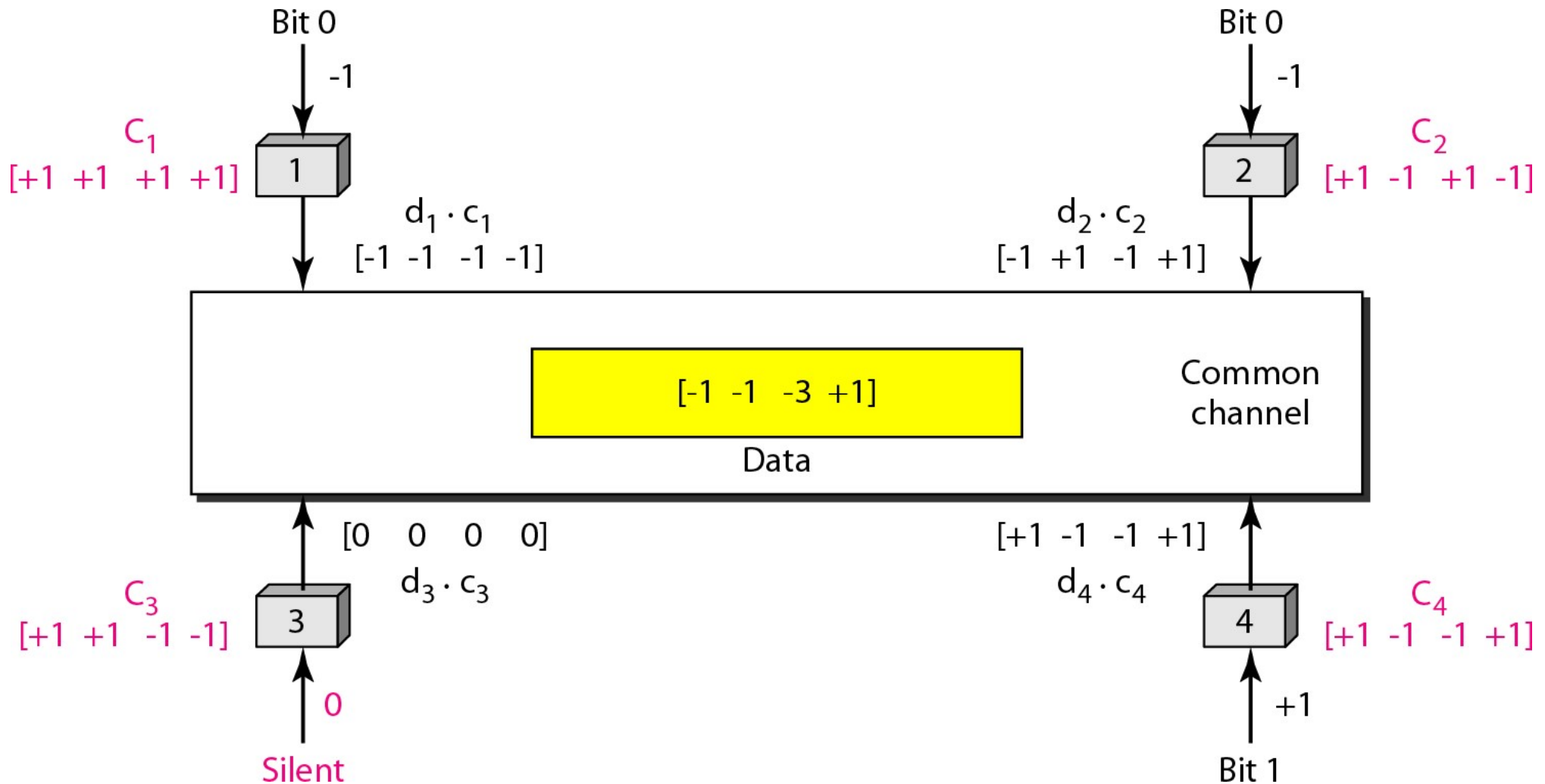
$C_3$

[+1 +1 -1 -1]

$C_4$

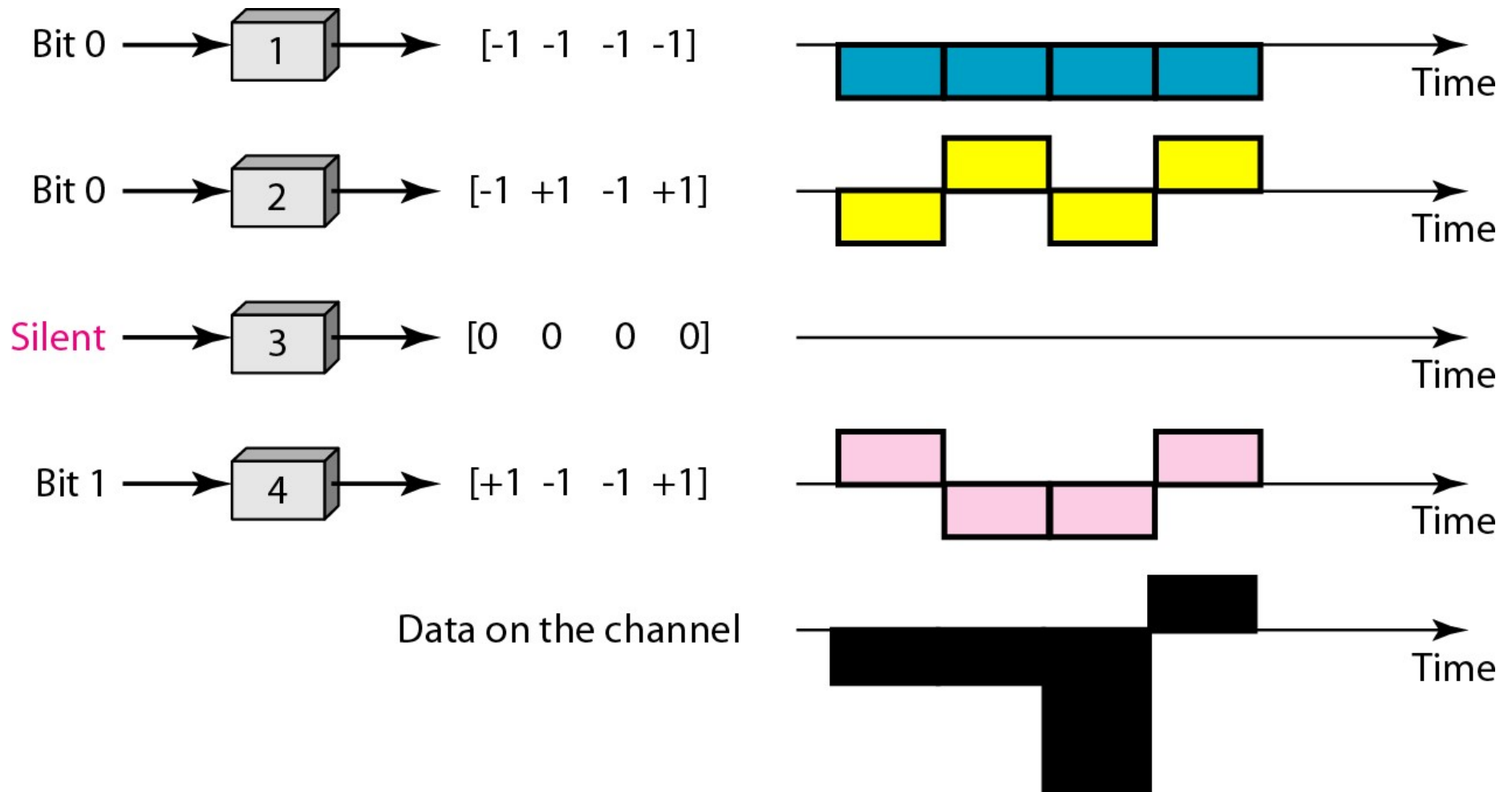
[+1 -1 -1 +1]

# Sharing channel in CDMA



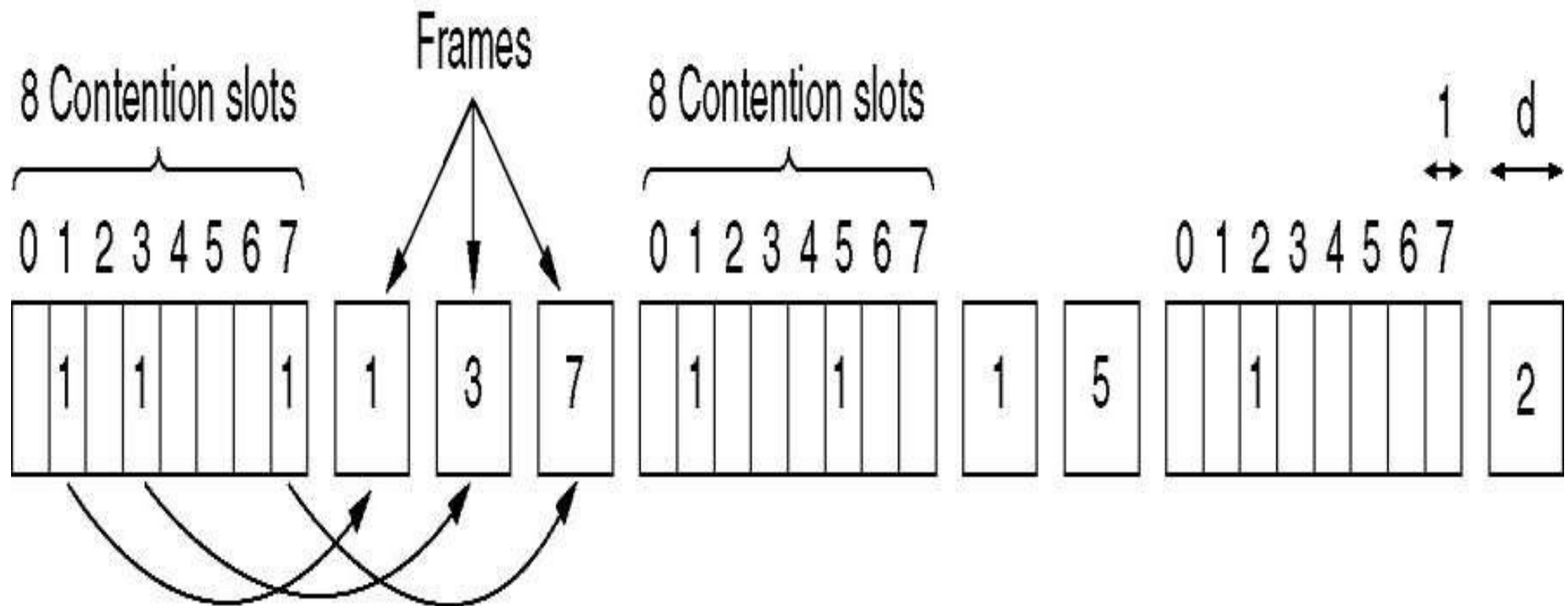
# Digital signal created by four stations in CDMA

---

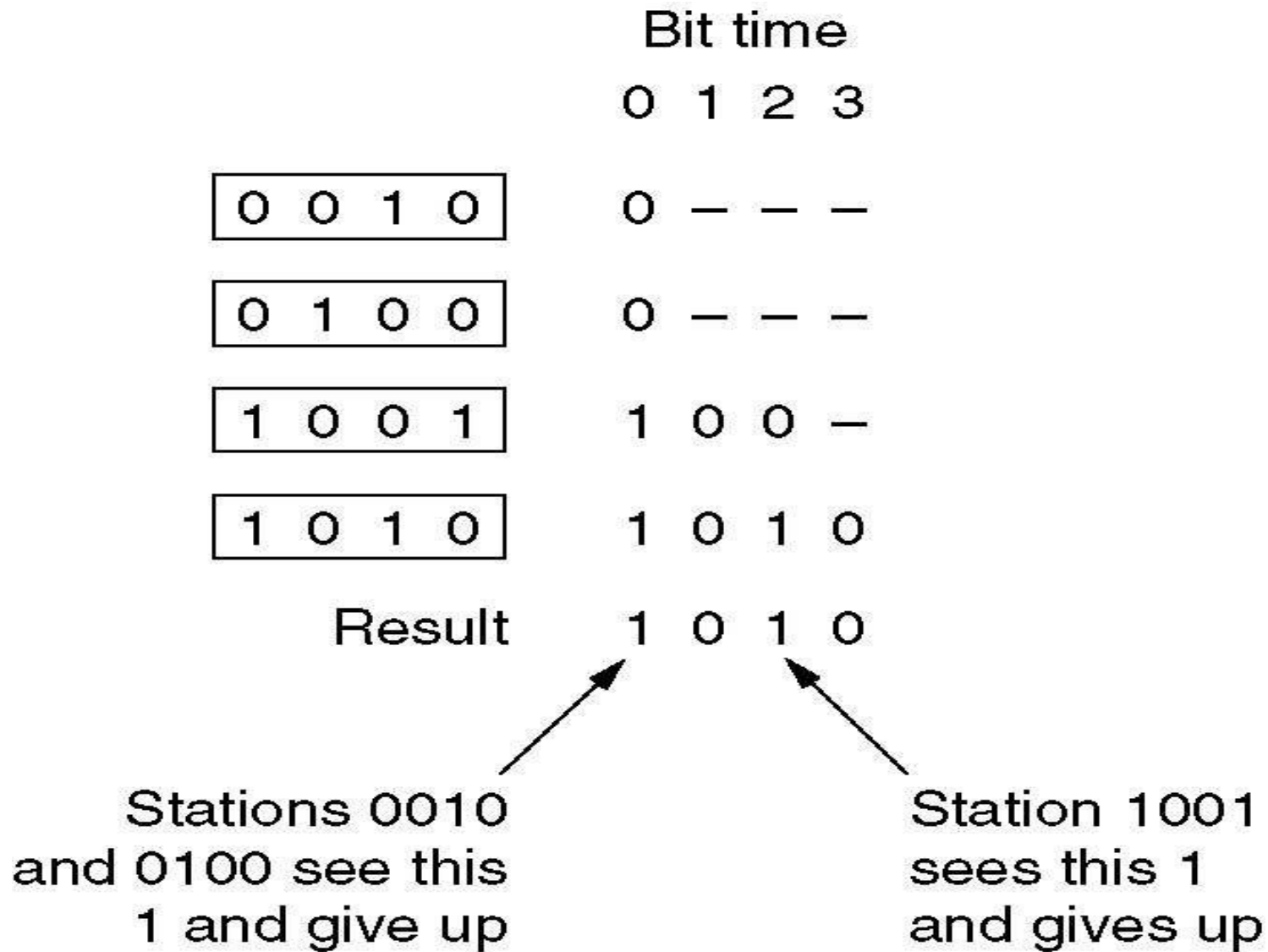


# Collision-Free Protocols

## *A Bit-Map Protocol*



# binary countdown protocol



# 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 Area 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.*

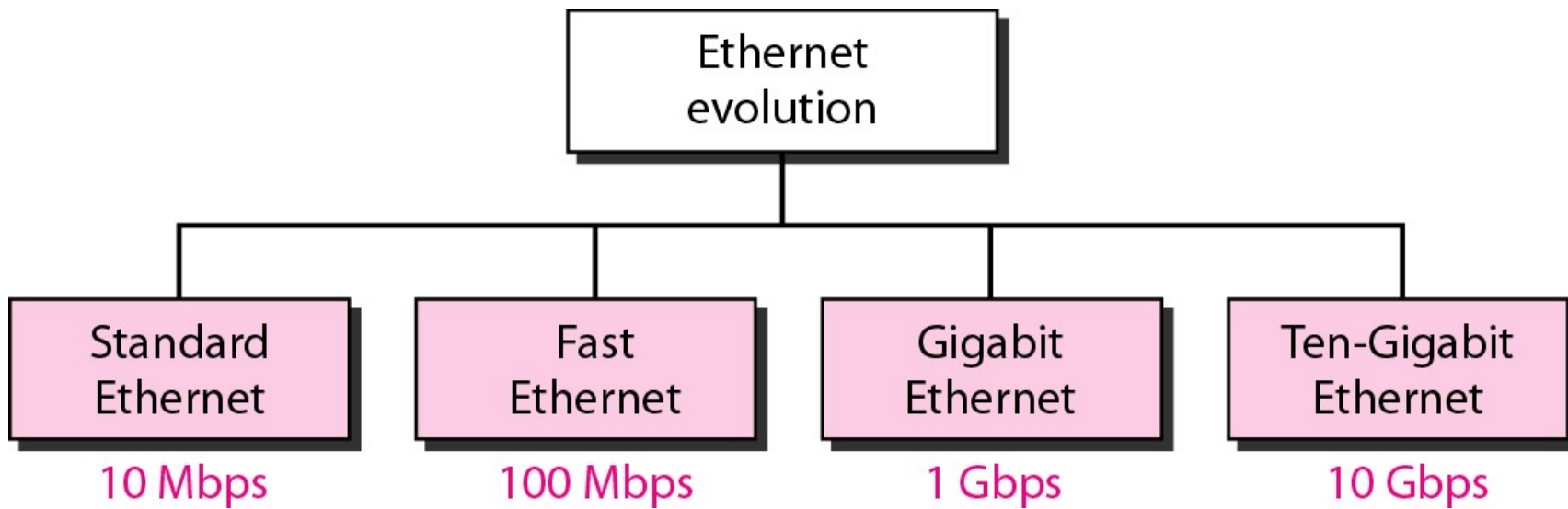
*Topics discussed in this section:*

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

---

**Preamble:** 56 bits of alternating 1s and 0s.

**SFD:** Start frame delimiter, flag (10101011)

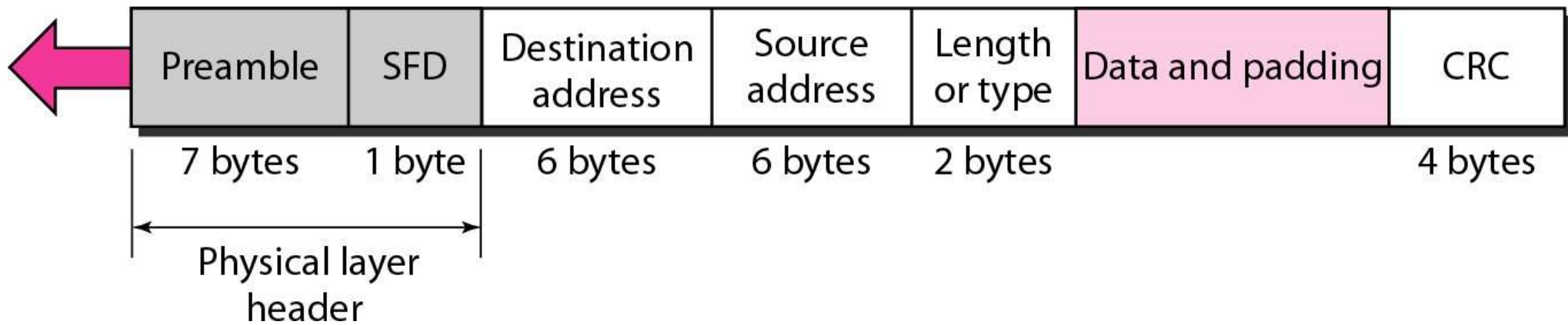
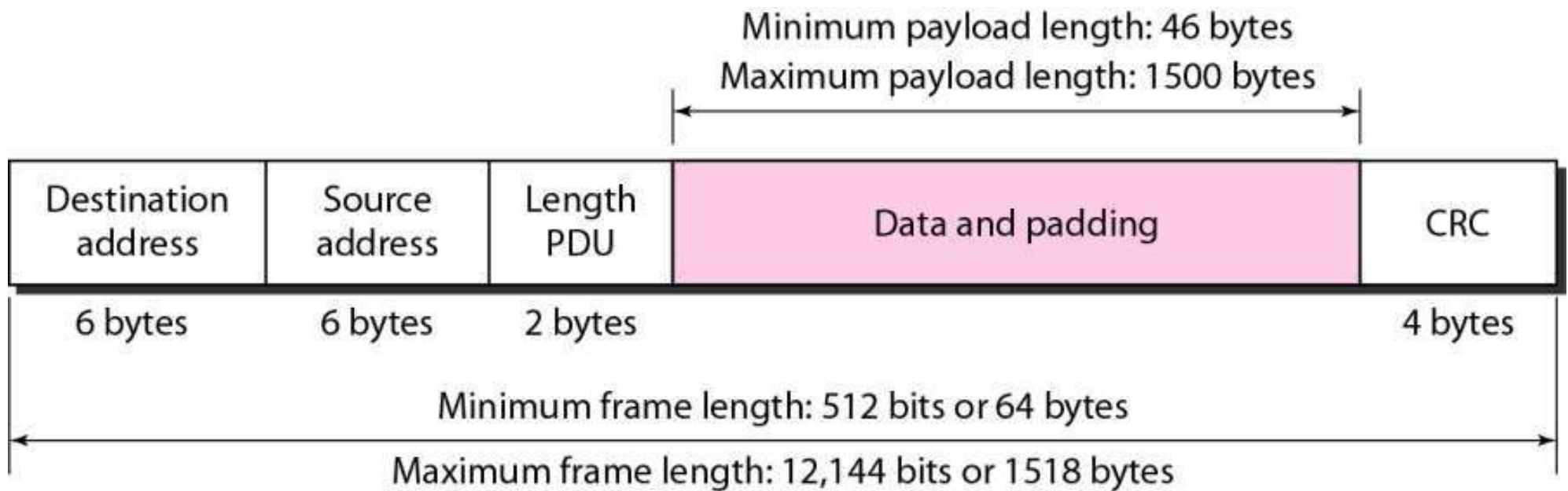


Figure 13.5 *Minimum and maximum lengths*



---

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

---

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

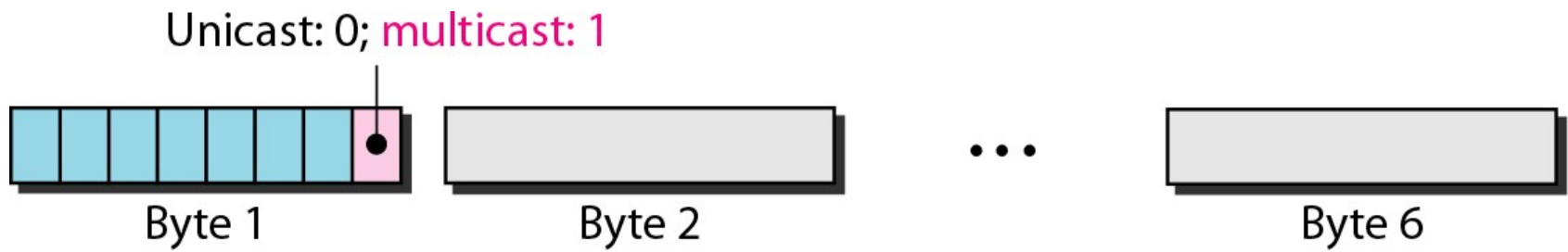


6 bytes = 12 hex digits = 48 bits

---

Figure 13.7 *Unicast and multicast addresses*

---

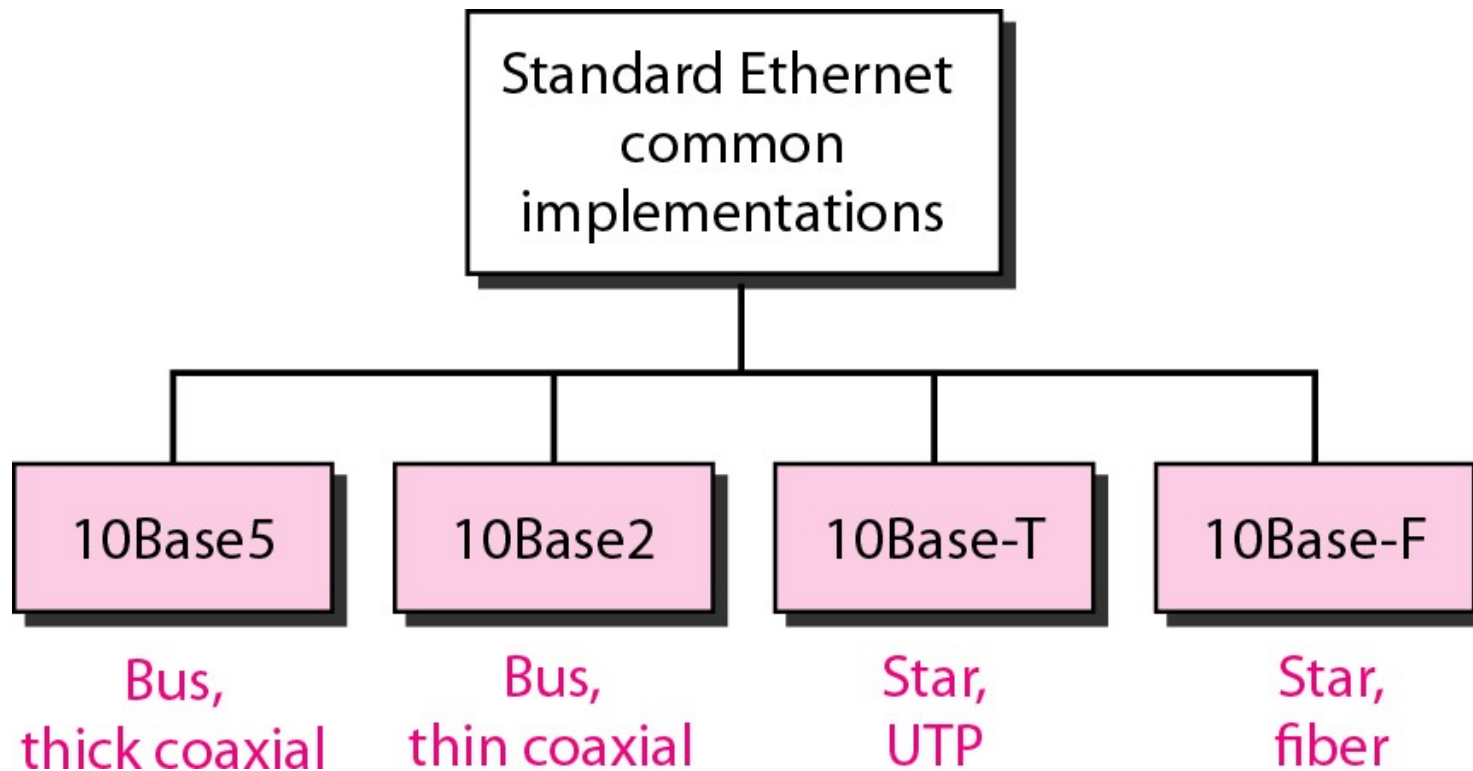


# Physical Layer

---

## *Categories of Standard Ethernet*

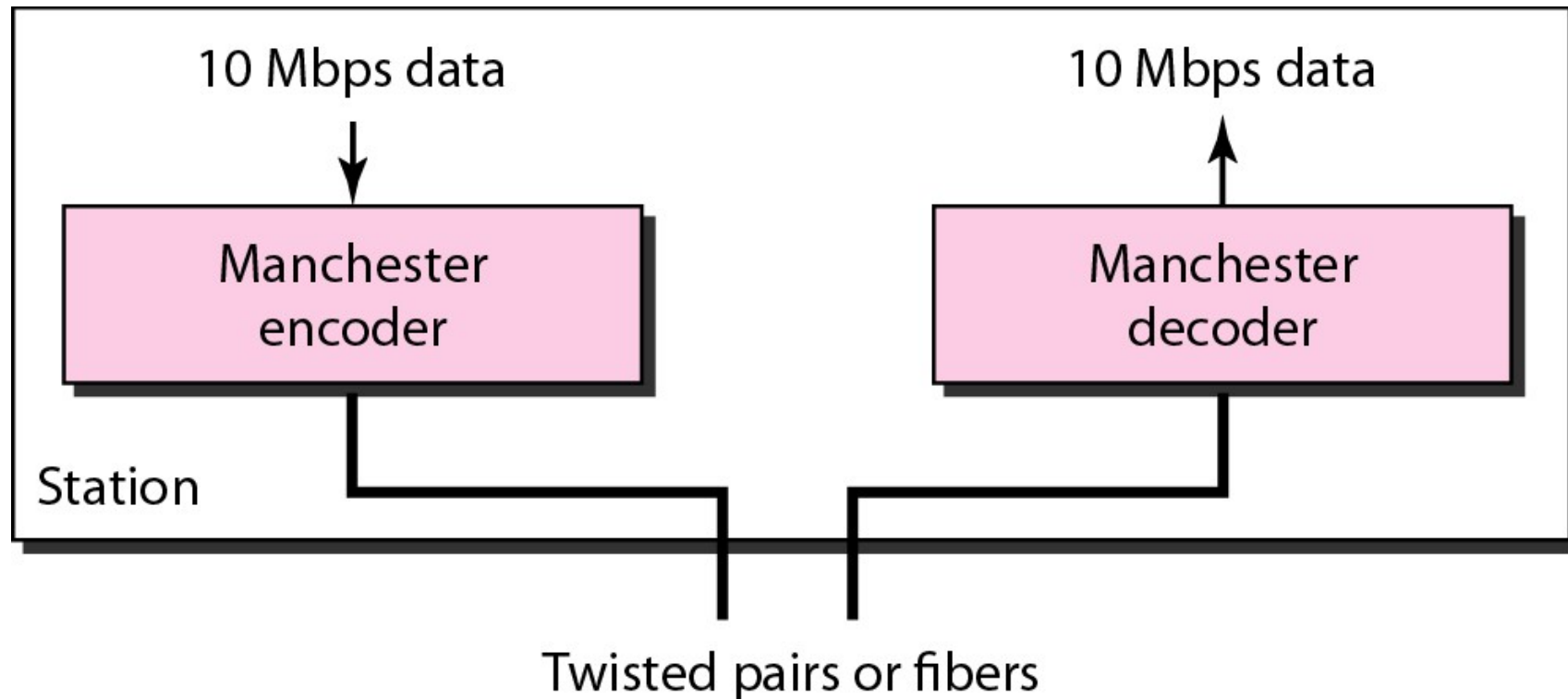
---



---

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

---



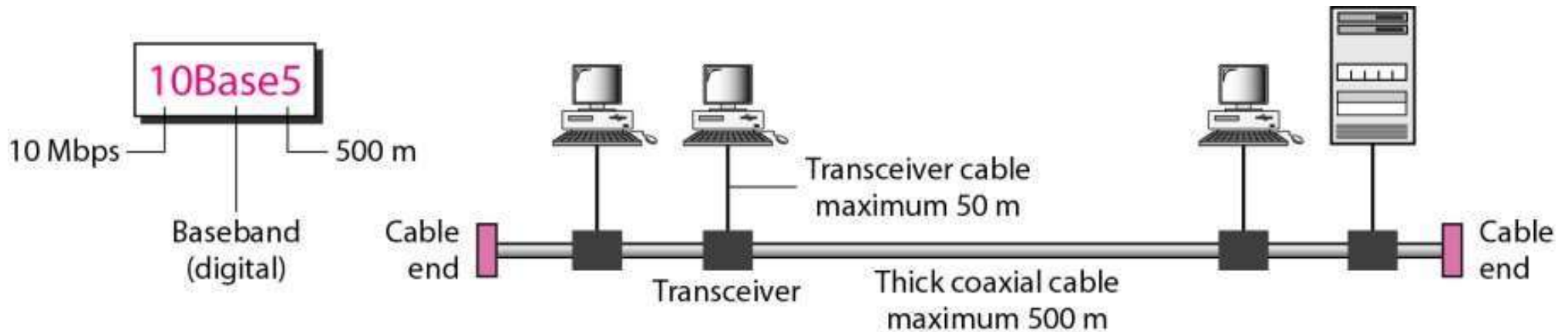
**Use digital signaling (baseband) at 10Mbps**

---

---

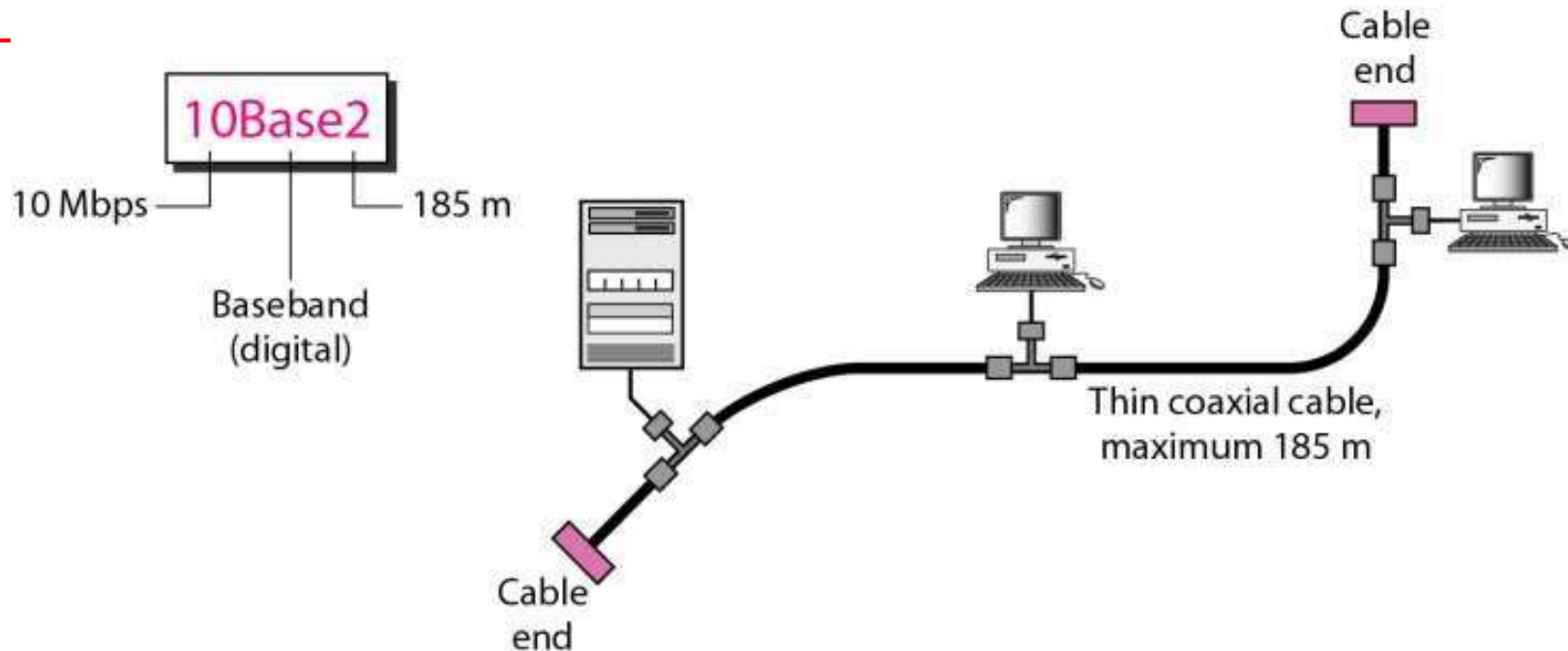
**Figure 13.10** *10Base5 implementation*

---



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*

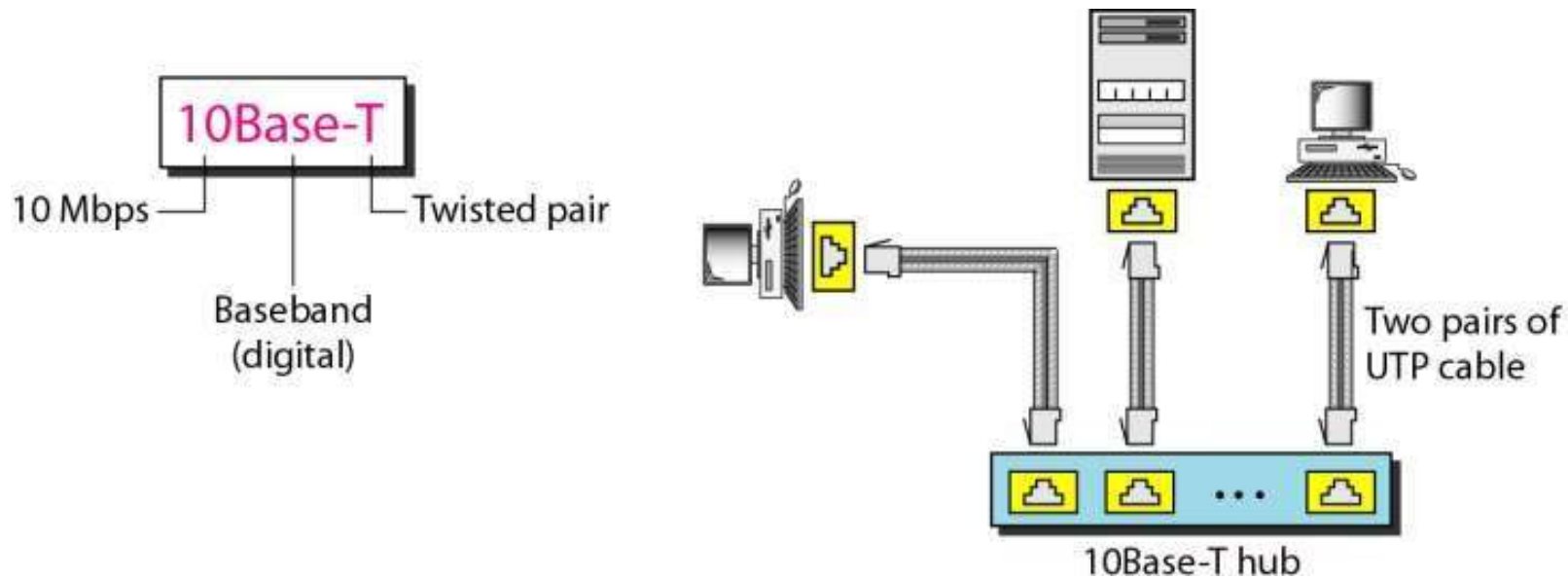


1. Known as Thin Ethernet
2. Uses bus topology with thin and flexible cable
3. Transceiver – part of NIC
4. Max length of each segment 185m

---

## *10Base-T implementation*

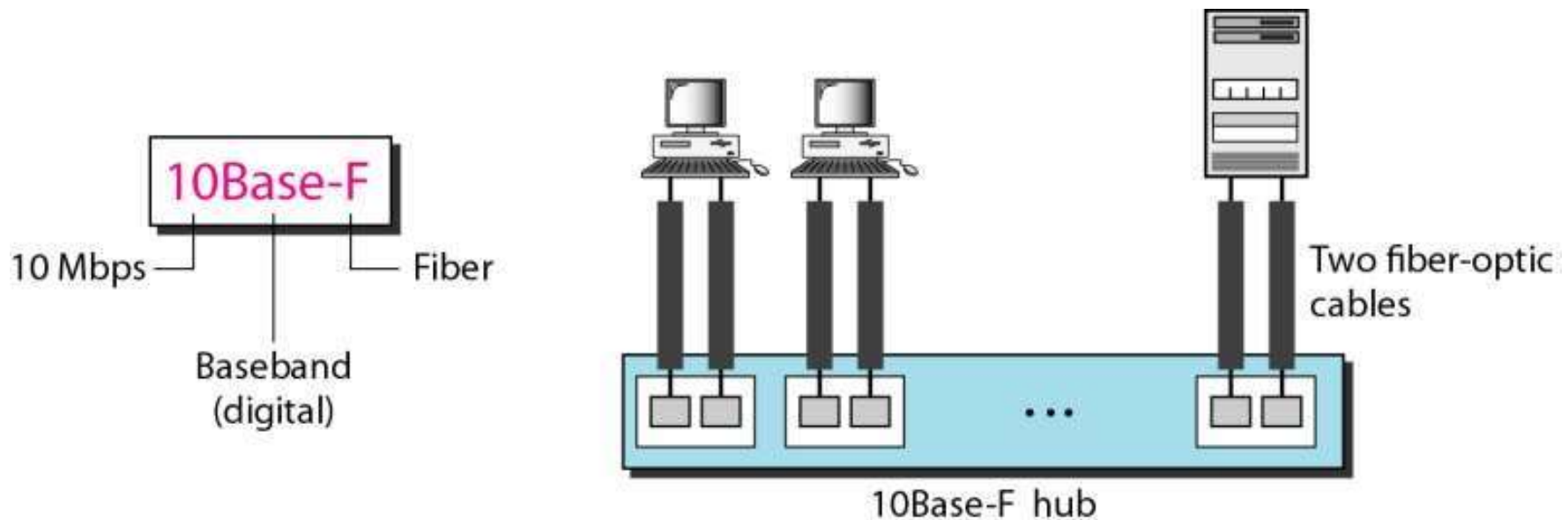
---



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

---

## *10Base-F implementation*



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

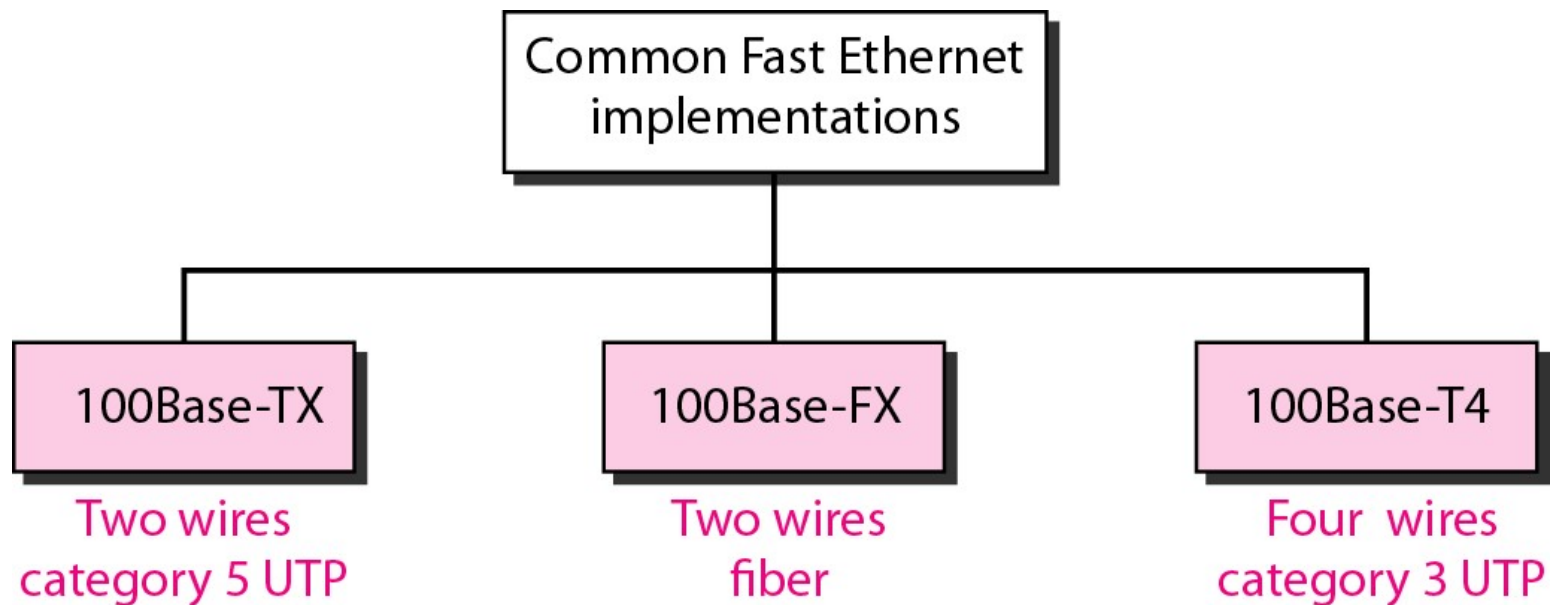
**Table 13.1** *Summary of Standard Ethernet implementations*

<i>Characteristics</i>	<i>10Base5</i>	<i>10Base2</i>	<i>10Base-T</i>	<i>10Base-F</i>
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

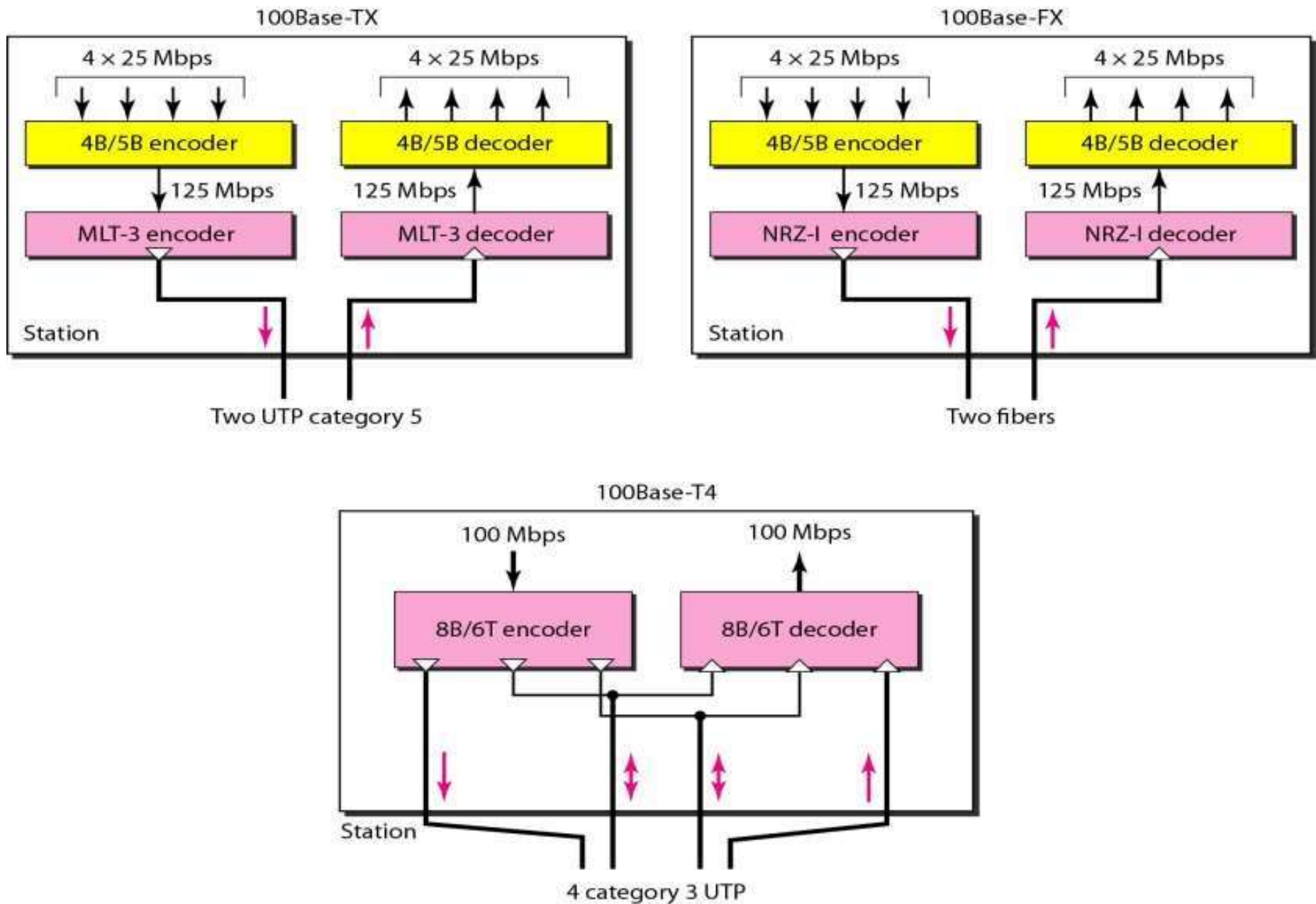
# FAST ETHERNET

---

**Figure 13.20** *Fast Ethernet implementations*



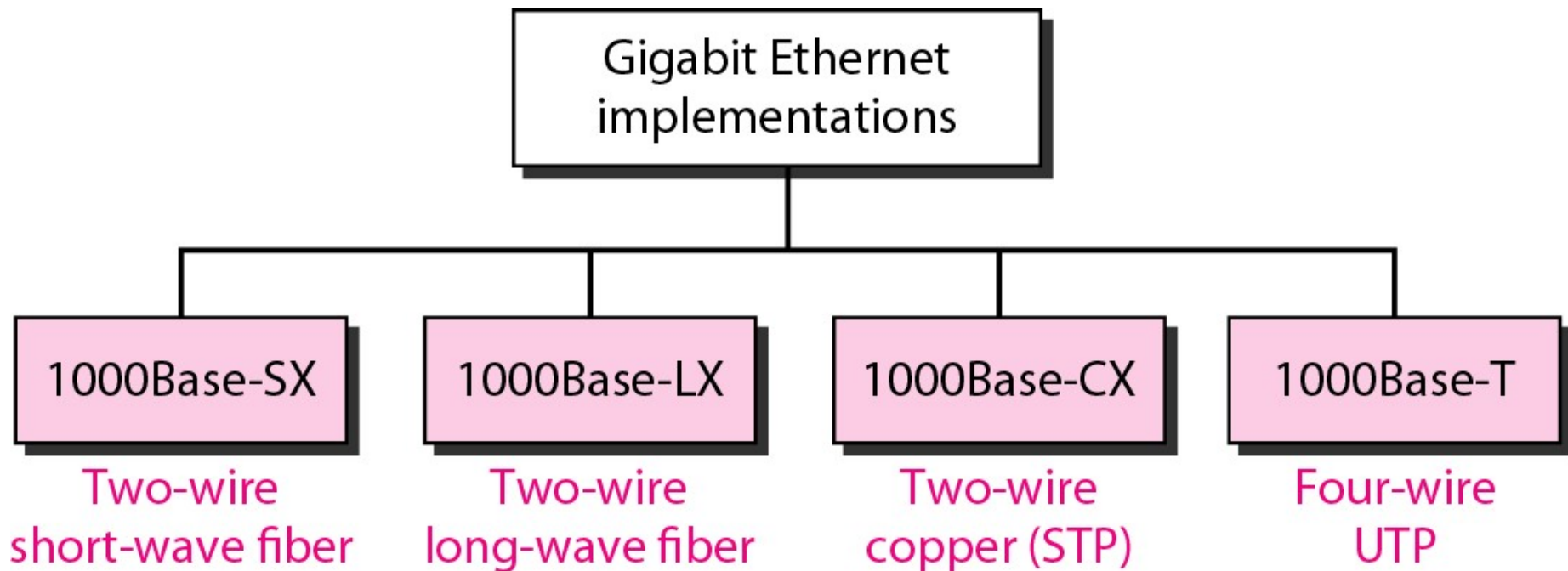
# Encoding for Fast Ethernet implementation



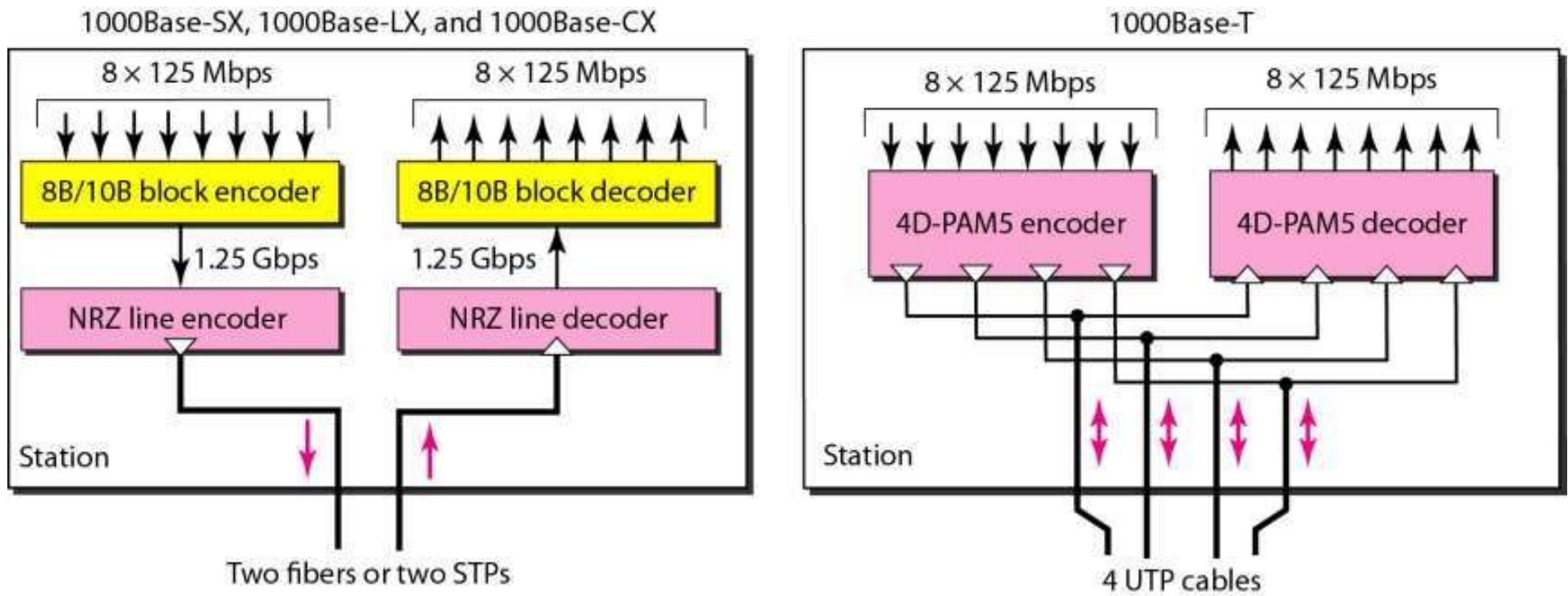
# GIGABIT ETHERNET

---

**Figure 13.23** *Gigabit Ethernet implementations*



**Figure 13.24** *Encoding in Gigabit Ethernet implementations*

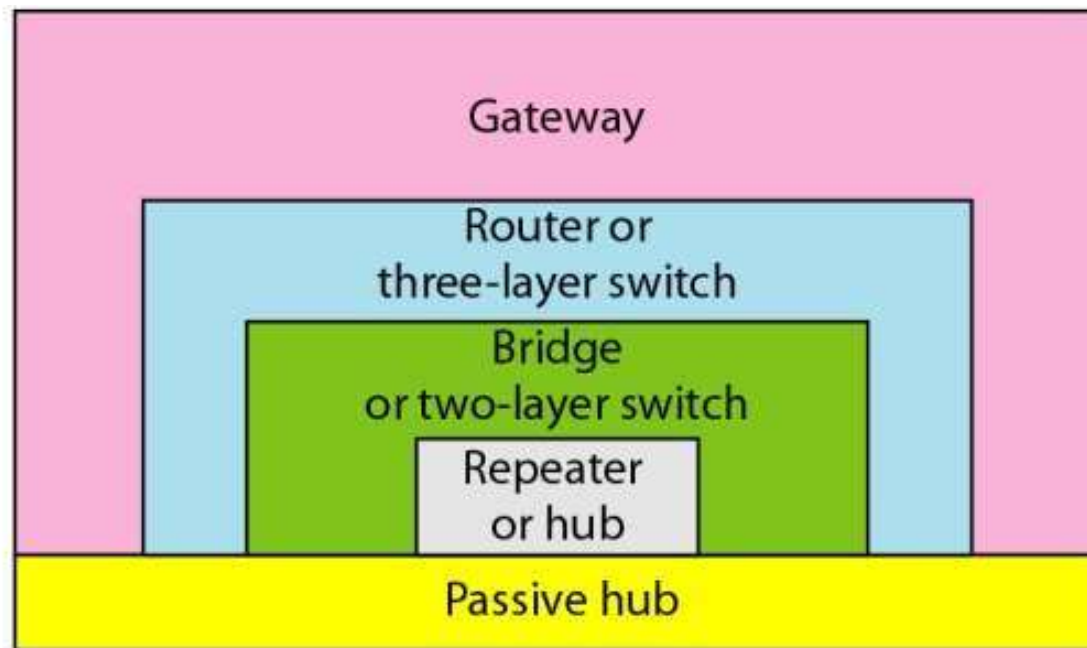


**Table 13.3** *Summary of Gigabit Ethernet implementations*

<i>Characteristics</i>	<i>1000Base-SX</i>	<i>1000Base-LX</i>	<i>1000Base-CX</i>	<i>1000Base-T</i>
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*

Application
Transport
Network
Data link
Physical

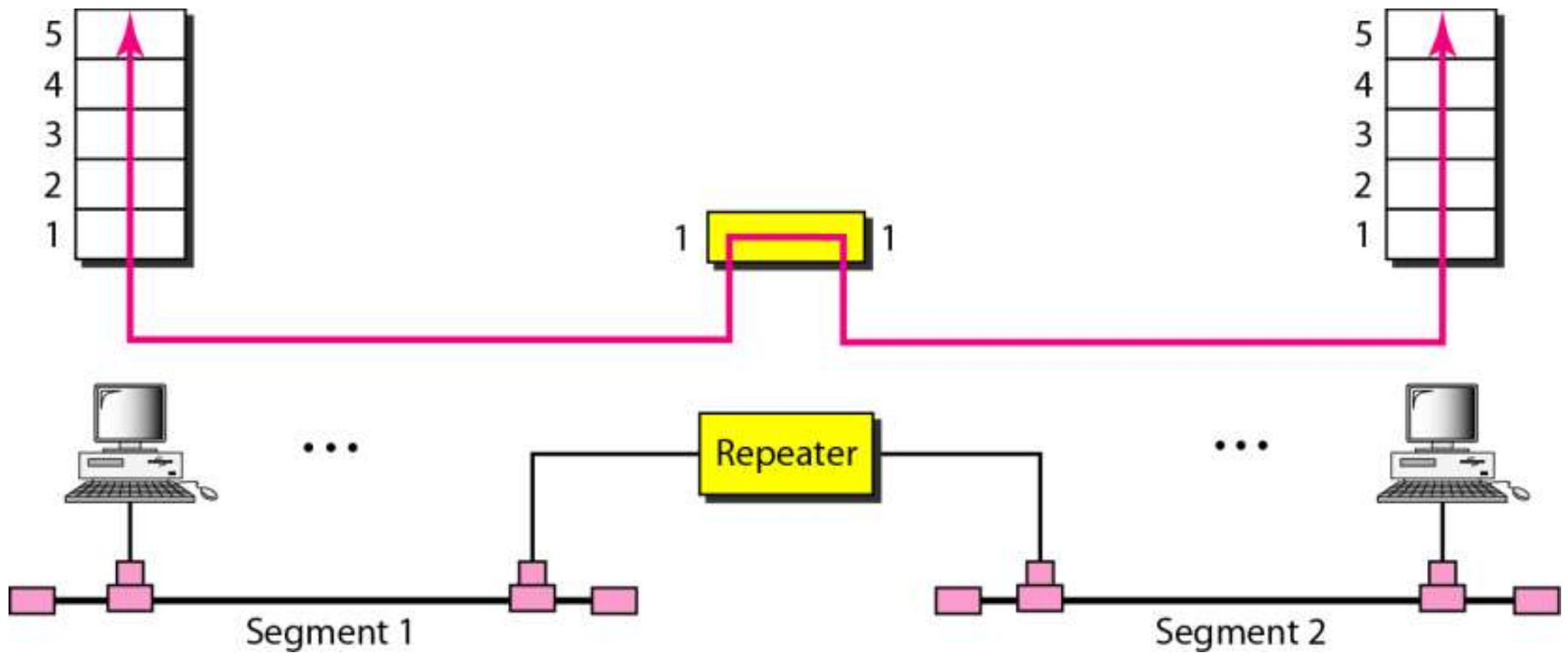


Application
Transport
Network
Data link
Physical

---

**Figure 15.2** *A repeater 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*

---

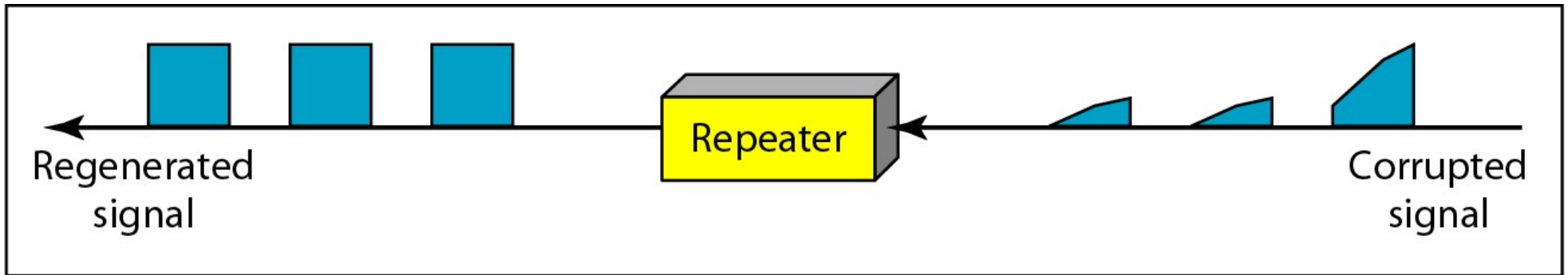
**A repeater is a regenerator,  
not an amplifier.**

---

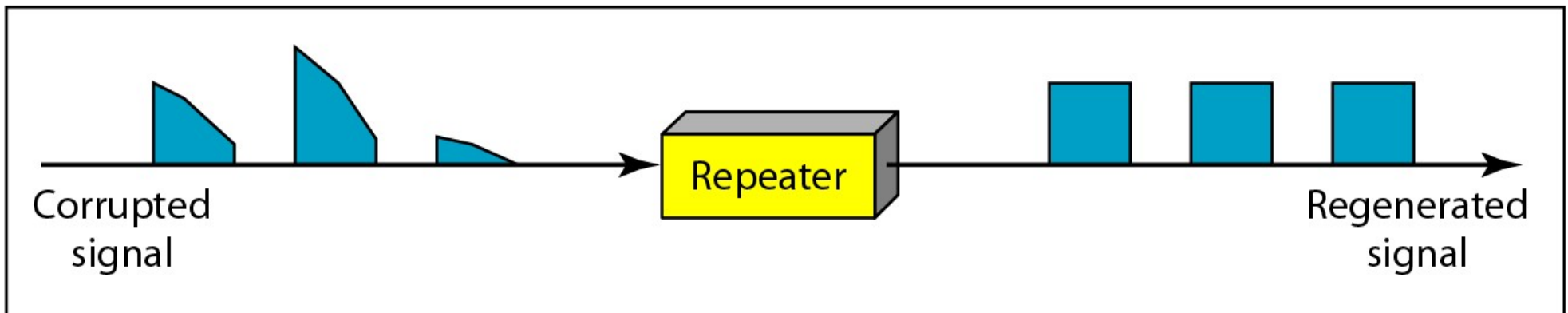
---

**Figure 15.3** *Function of a repeater*

---



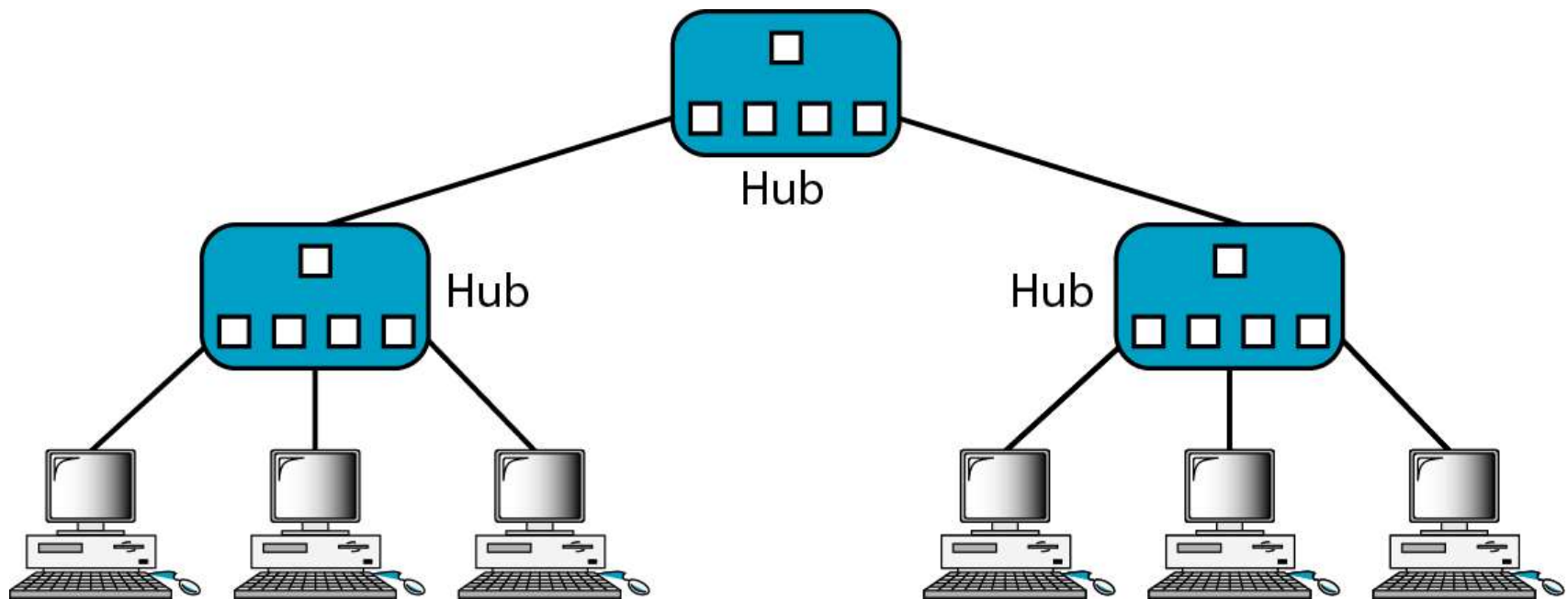
a. Right-to-left transmission.



b. Left-to-right transmission.

**Figure 15.4** *A hierarchy of hubs*

---



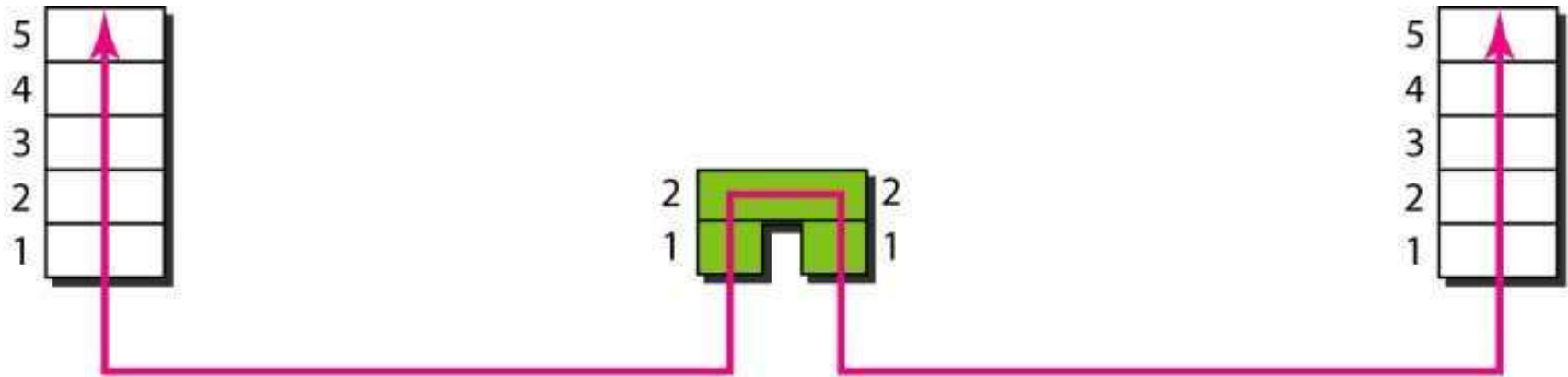


---

*Note*

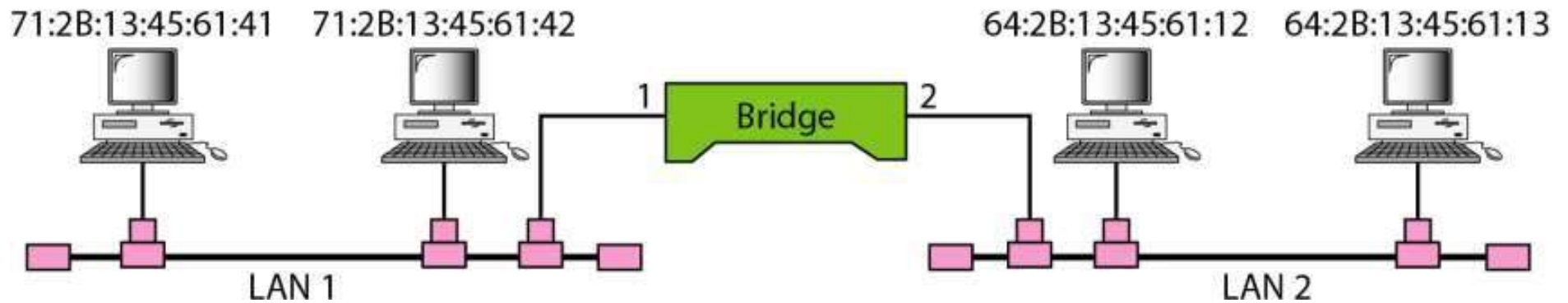
**A bridge has a table used in filtering decisions.**

**Figure 15.5** *A bridge 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

Bridge Table



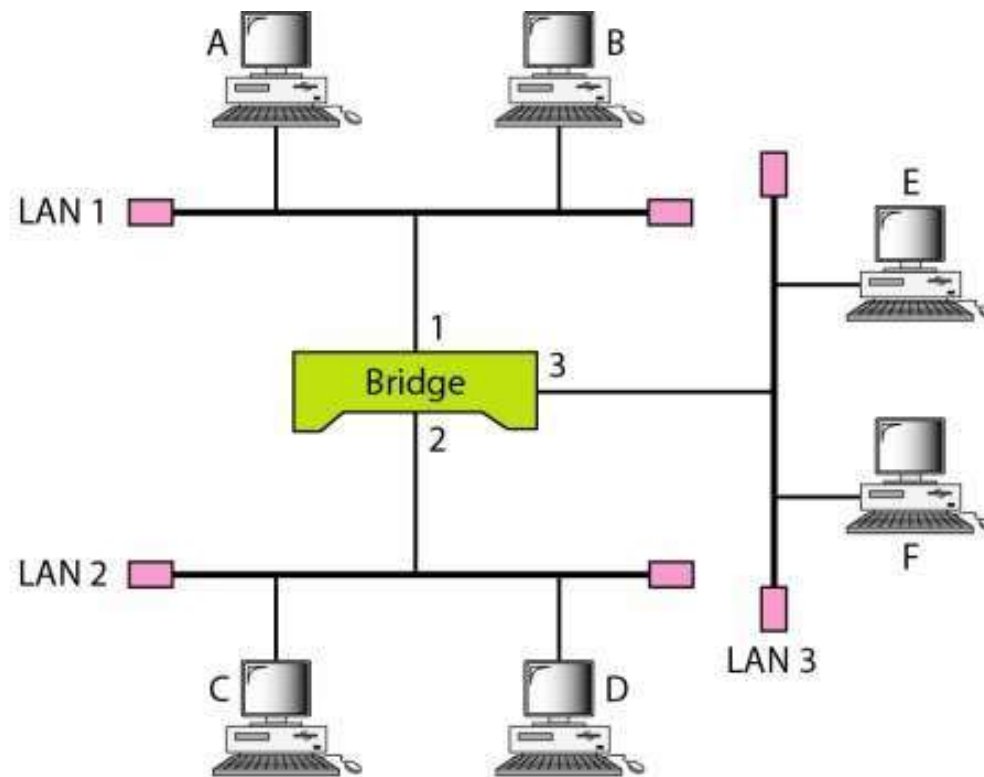


---

*Note*

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

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



Address	Port

a. Original

Address	Port
A	1

b. After A sends a frame to D

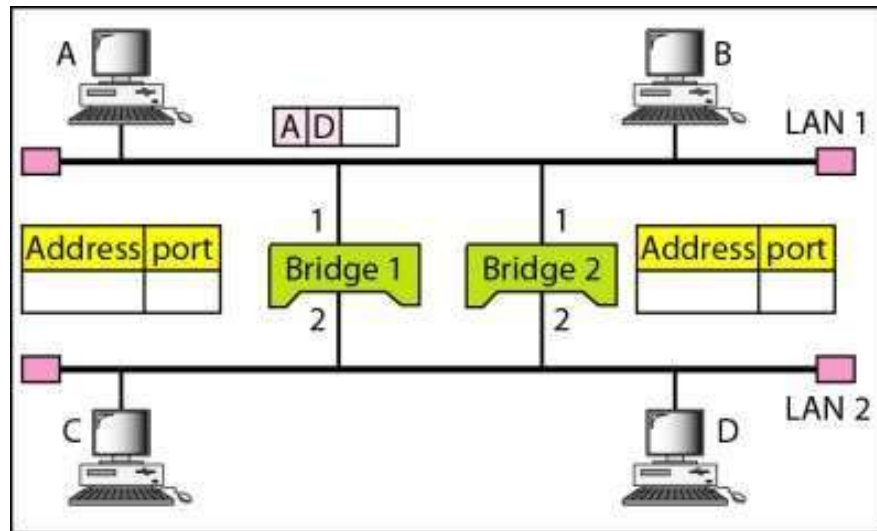
Address	Port
A	1
E	3

c. After E sends a frame to A

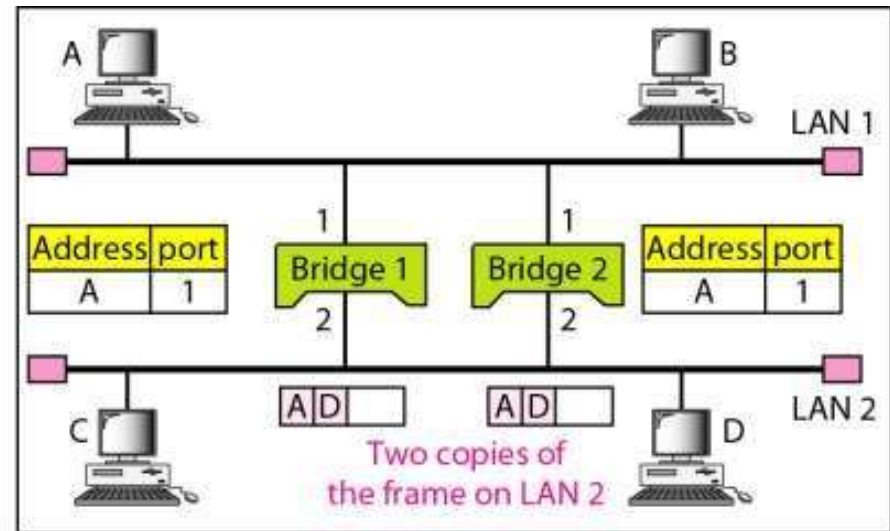
Address	Port
A	1
E	3
B	1

d. After B sends a frame to C

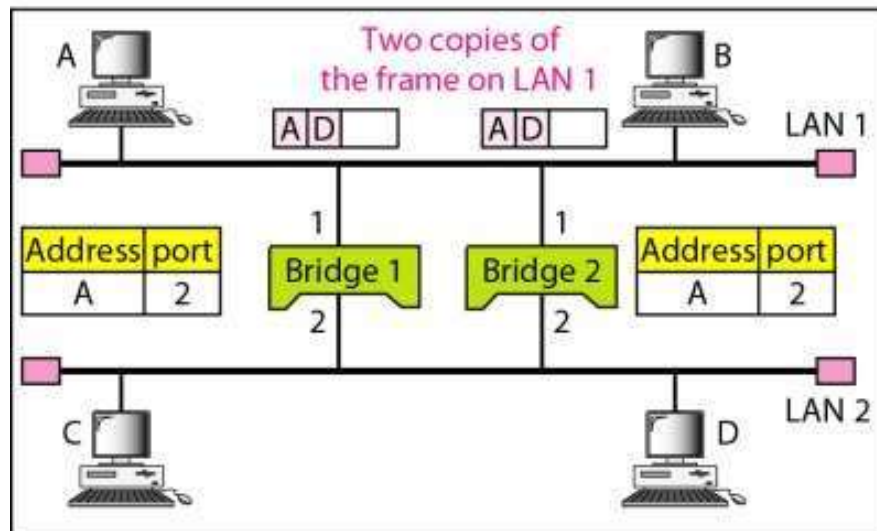
# Figure 15.7 Loop problem in a learning bridge



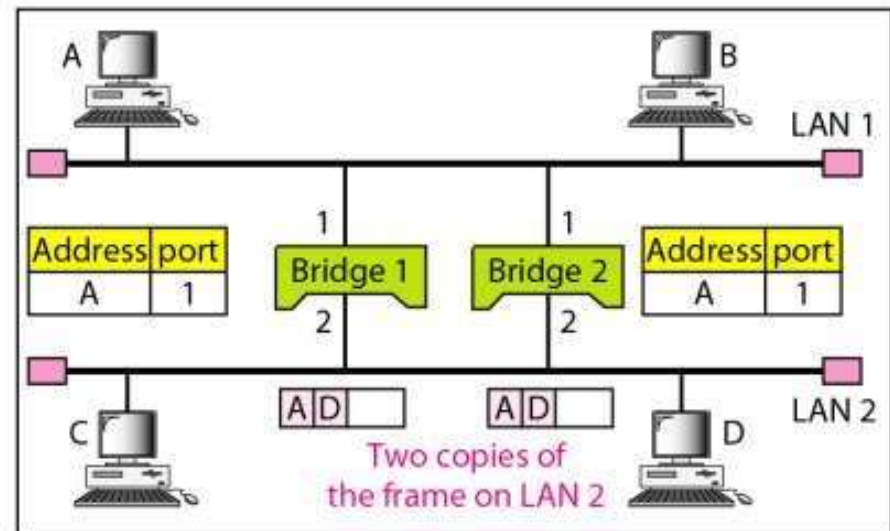
a. Station A sends a frame to station D



b. Both bridges forward the frame

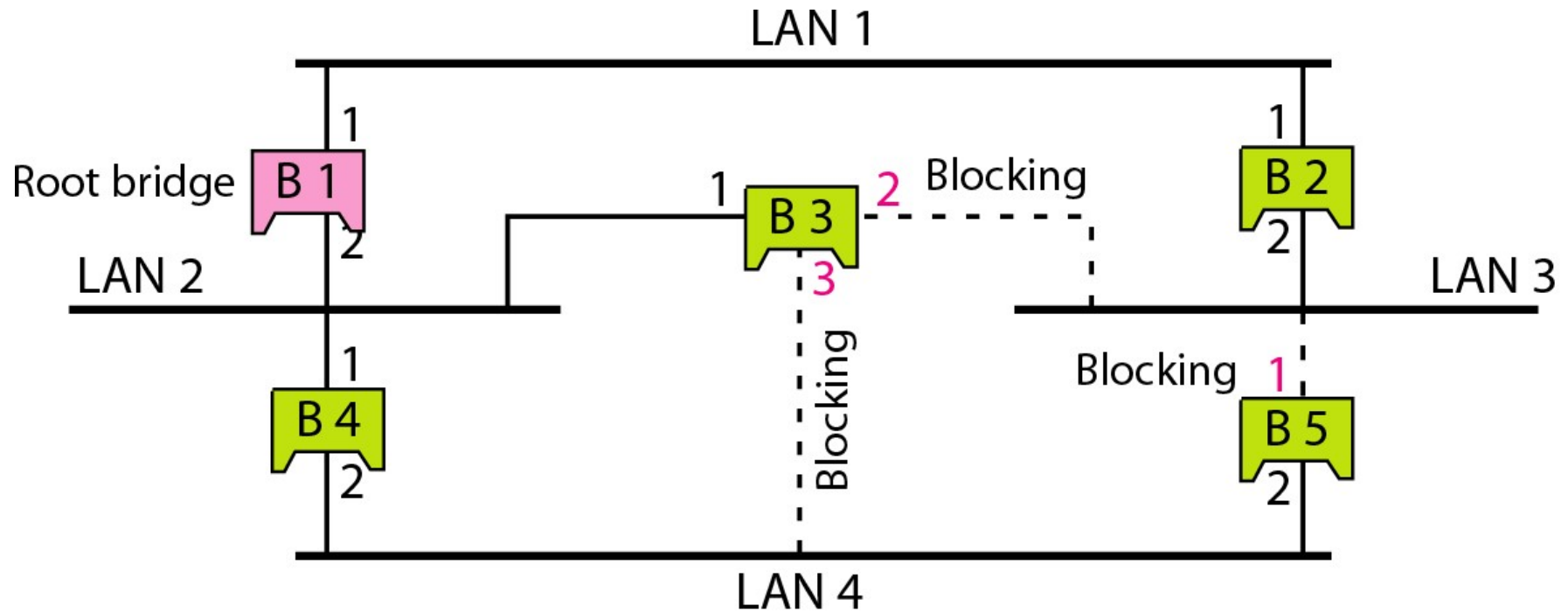


c. Both bridges forward the frame



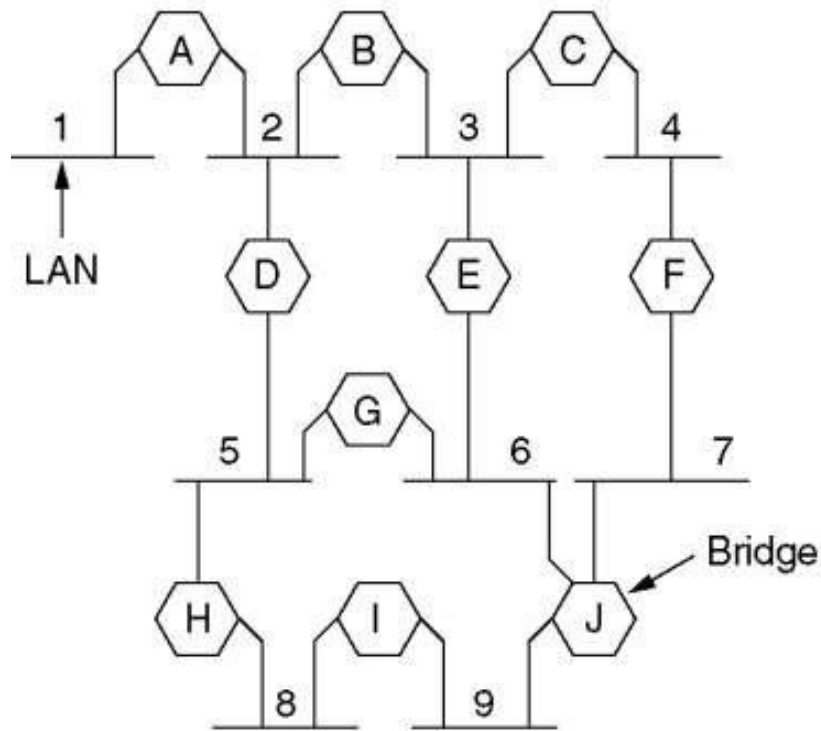
d. Both bridges forward the frame

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

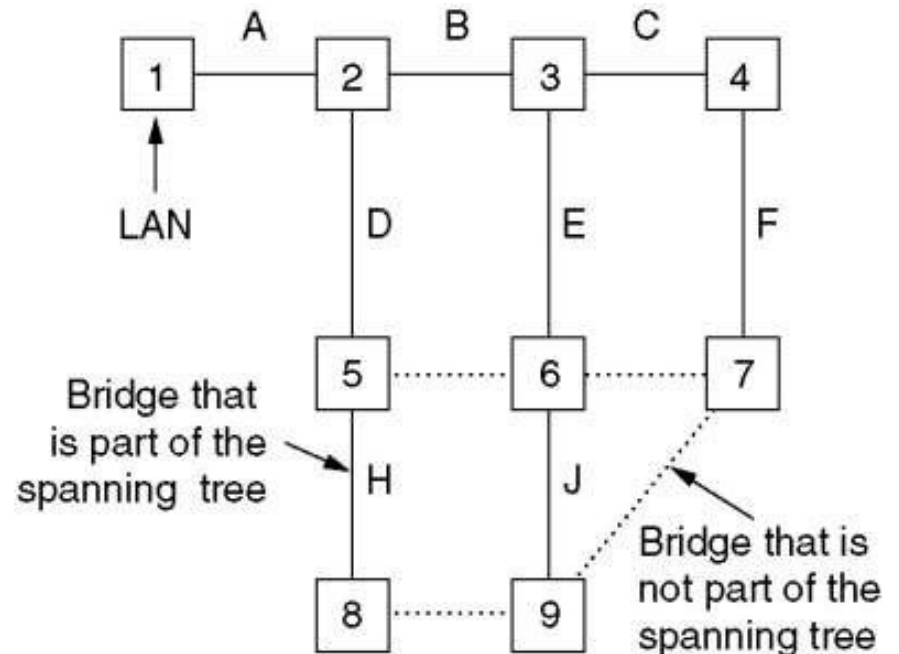


Ports 2 and 3 of bridge B3 are blocking ports (no frame is sent out of these ports).  
Port 1 of bridge B5 is also a blocking port (no frame is sent out of this port).

# Spanning Tree Bridges



(a)



(b)

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

# Bridge types

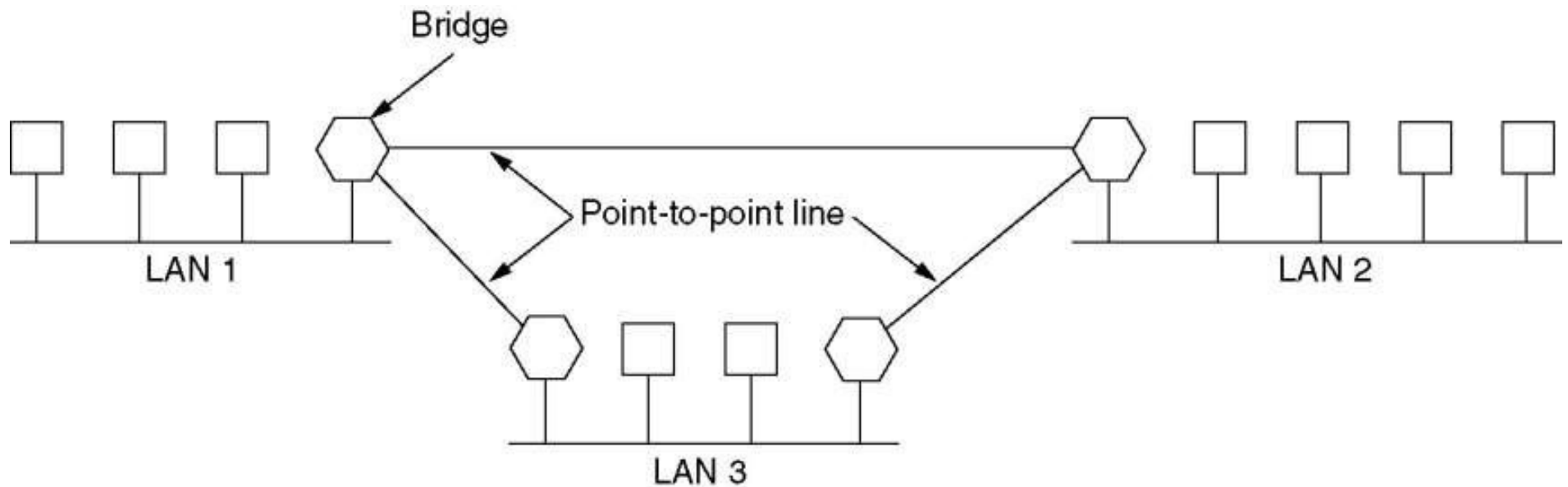
1. Source routing bridges

- \* Routing parameters will be take care by source only

1. Transparent bridges

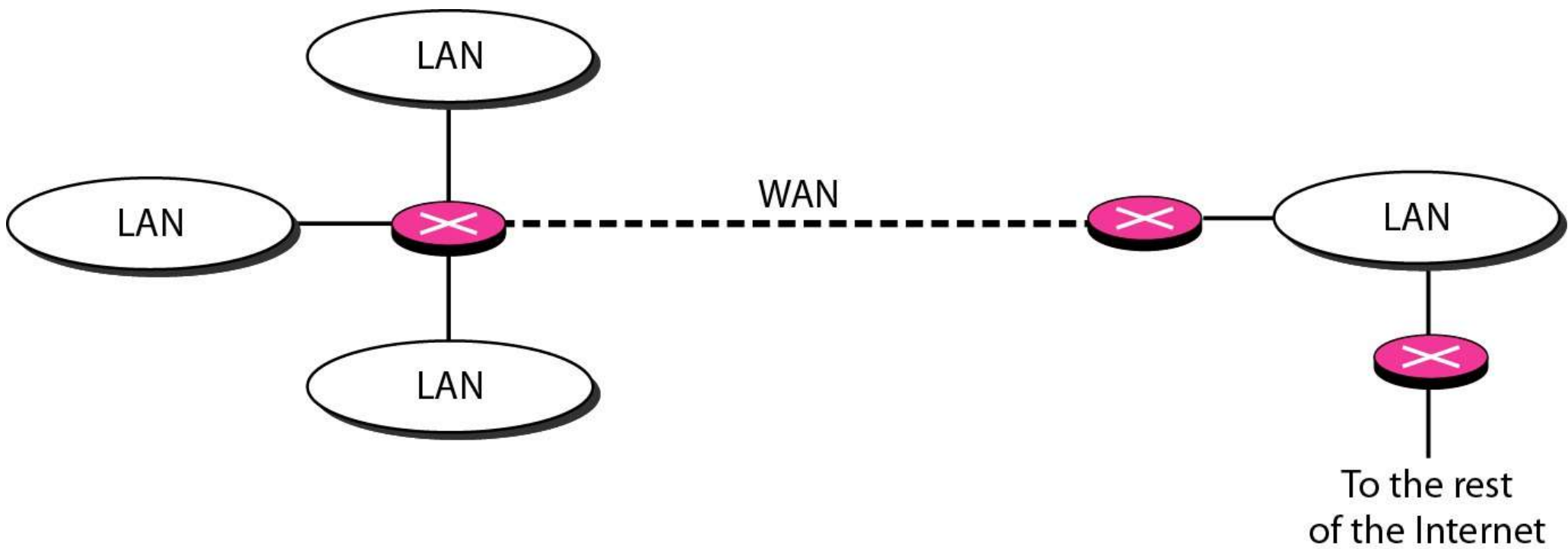
2. Remote bridges

# 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.