Module 2: *Data Link and MAC Layer*

Part-III Multiple Access (Medium Access Control)

Dr. Prasanna Shete
Dept. of Computer Engineering
K. J. Somaiya College of Engineering

Slide Source: B. A. Forauzan, Data Communications and Networking, McGraw-Hill Online Learning Centre
http://highered.mheducation.com/sites/0072967757/information_center_view0/index.html

Figure 12.1 *Data link layer divided into two functionality-oriented sublayers*

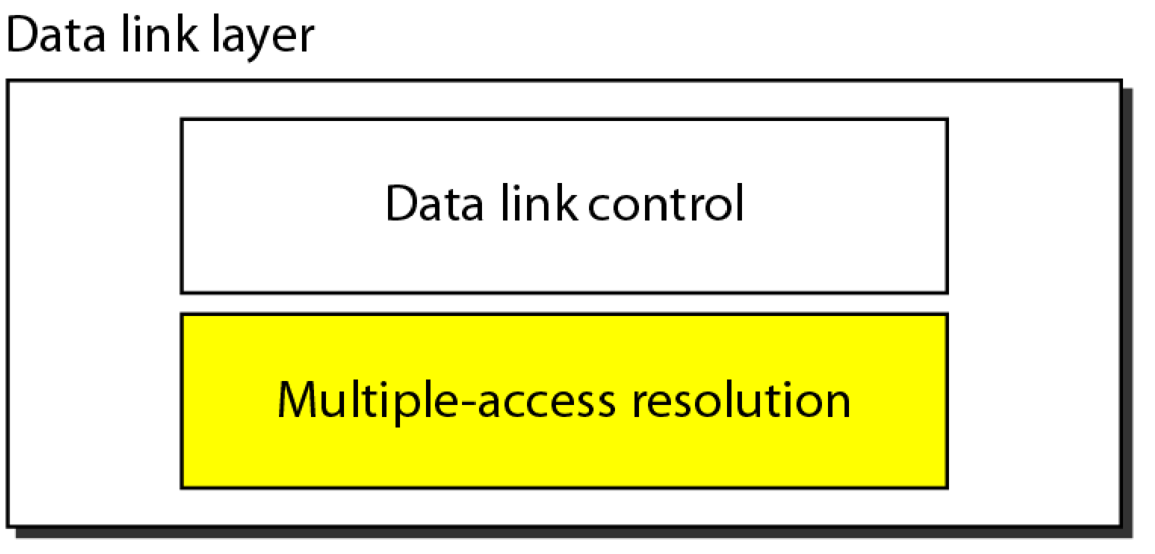
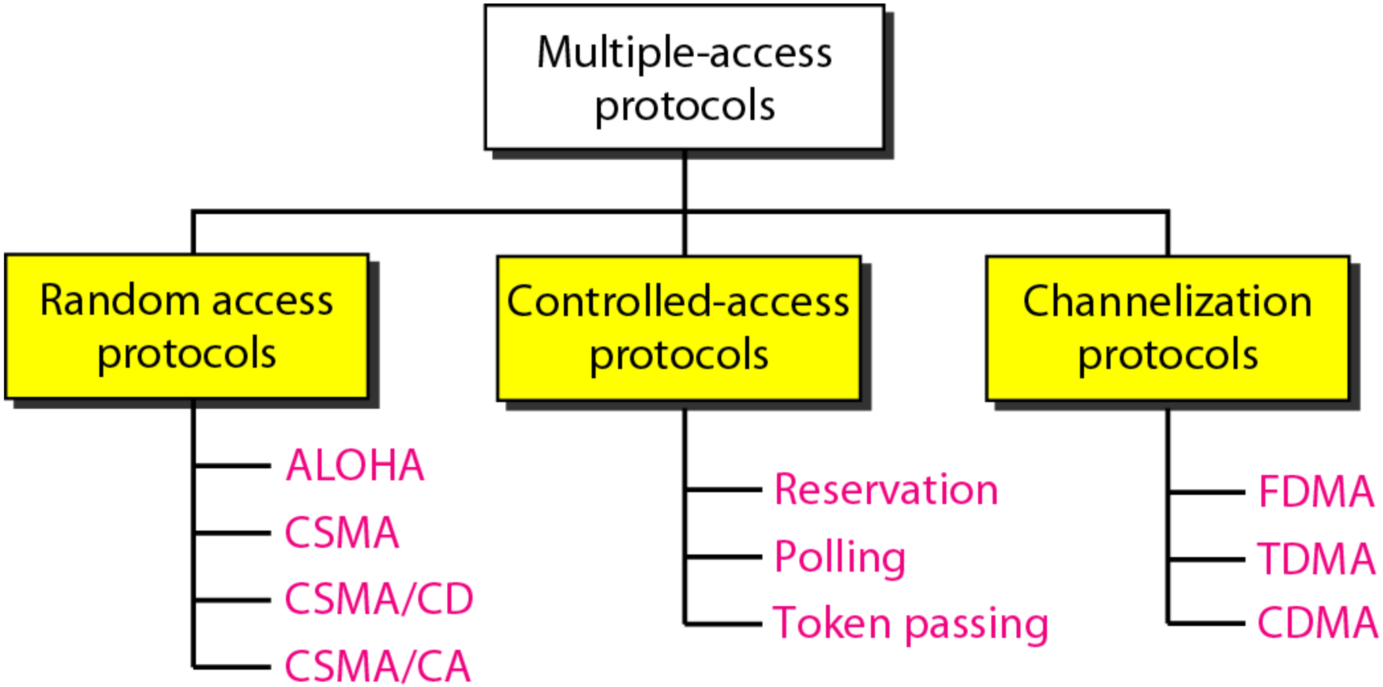


Figure 12.2 *Taxonomy of multiple-access protocols discussed in this chapter*



12-1 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. At each instance, a station that has data to send uses a procedure defined by the protocol to make a decision on whether or not 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

ALOHA

- Earliest random access method developed at University of Hawaii → early 70's

- Simple protocol, Each station sends a frame whenever it has a frame to send
- Possibility of collision between frames from different stations (due to shared channel)

Figure 12.3 *Frames in a pure ALOHA network*

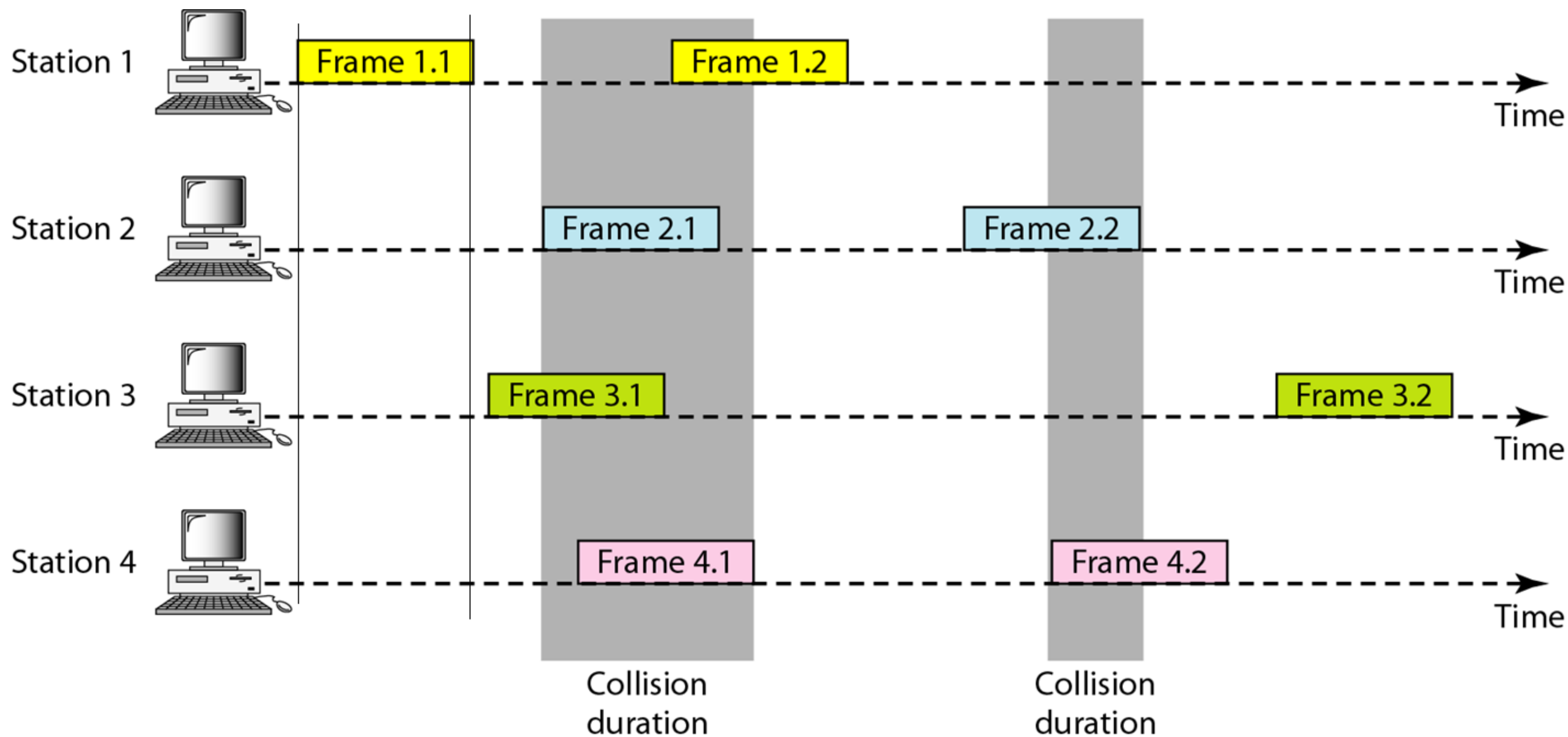
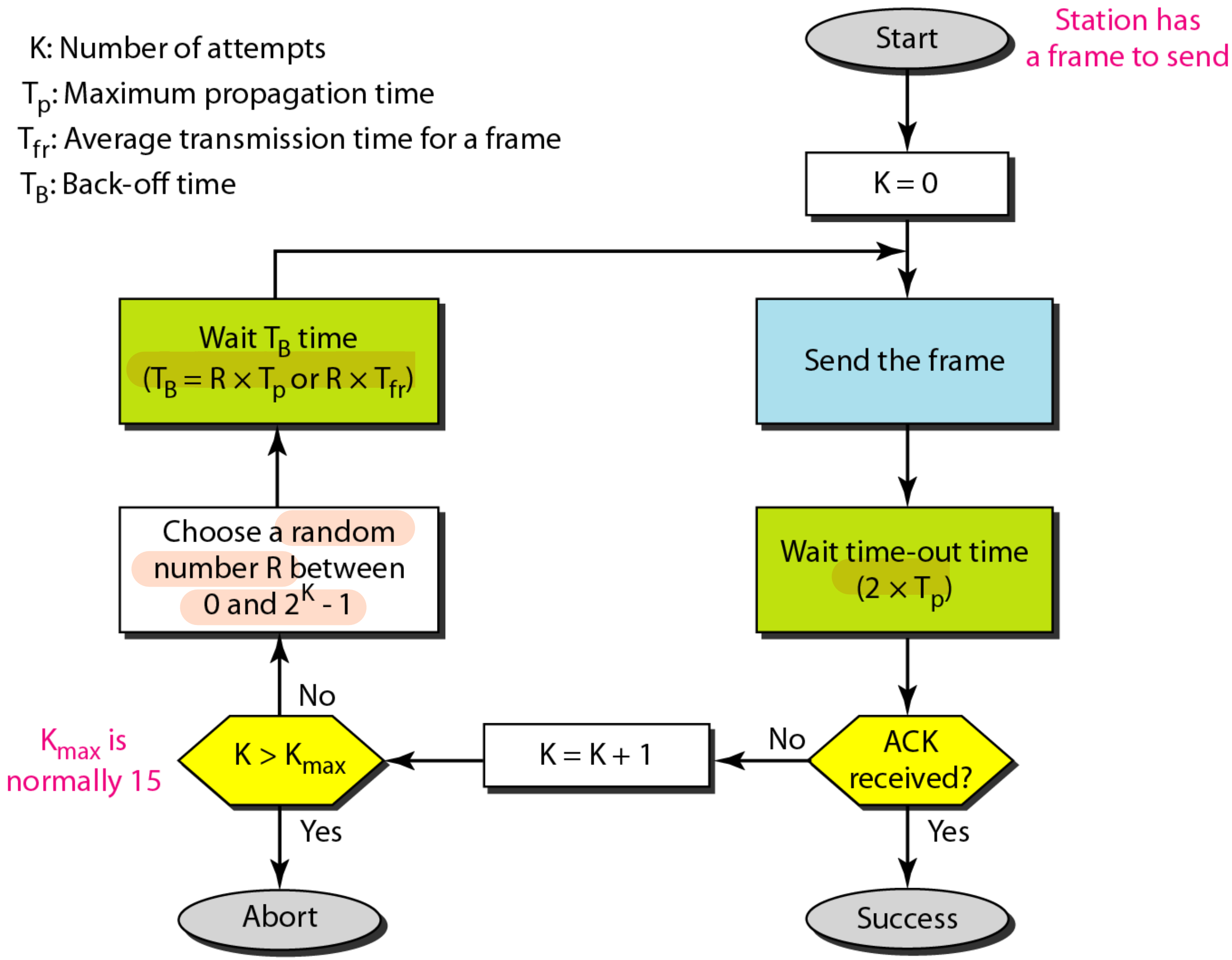


Figure 12.4 *Procedure for pure ALOHA protocol*



Example 12.1

The stations on a wireless ALOHA network are a maximum of 600 km apart. If we assume that signals propagate at 3×10^8 m/s, find the value of T_B for different values of K

→*Propagation delay is*

$T_p = (600 \times 10^3) / (3 \times 10^8) = 2$ ms

a. For $K = 1$, the range is {0, 1}. The station needs to generate a random number with a value of 0 or 1. This means that T_B is either 0 ms (0×2) or 2 ms (1×2), based on the outcome of the random variable.

Example 12.1 (continued)

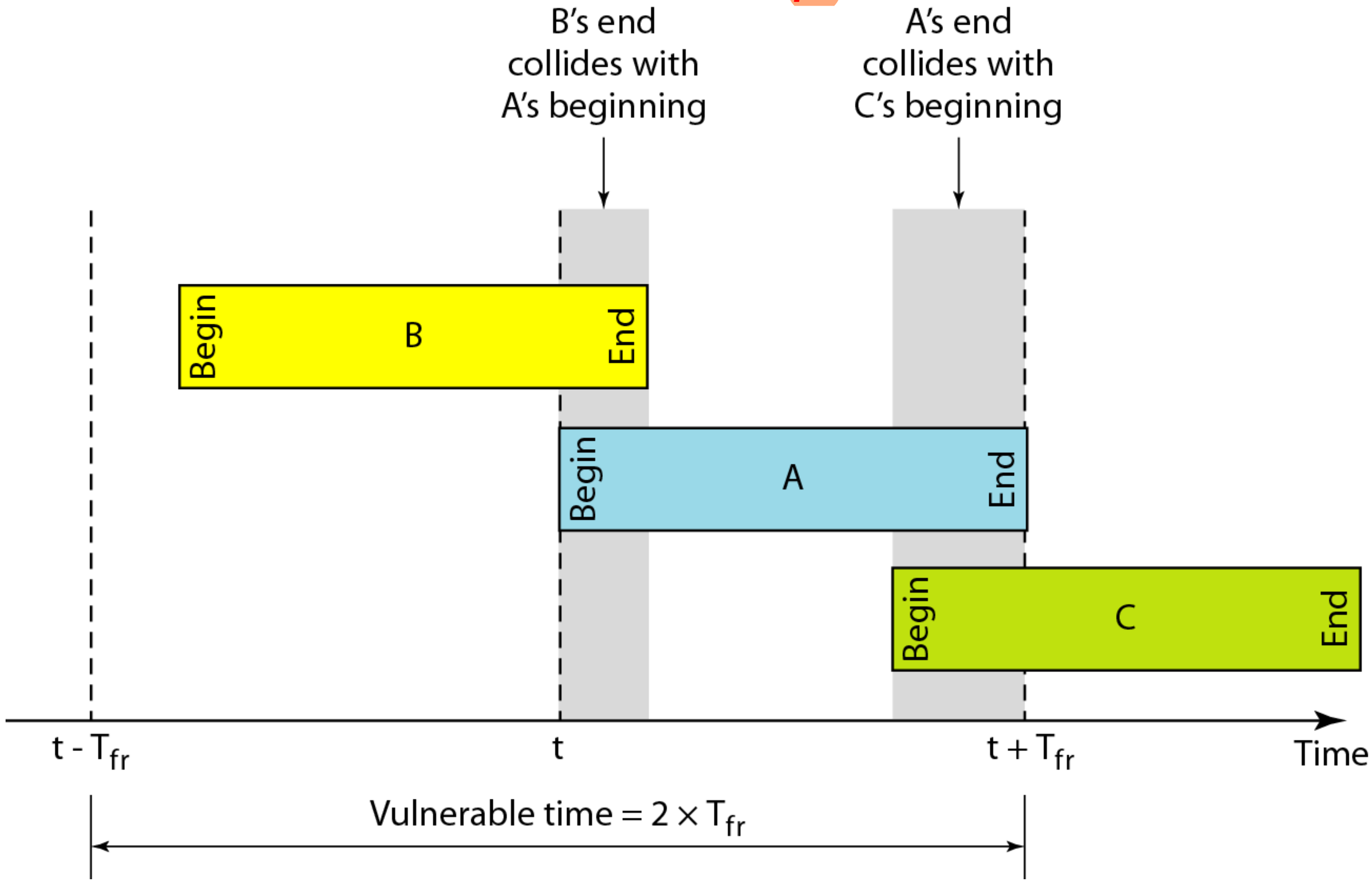
b. ***For $K = 2$, the range is $\{0, 1, 2, 3\}$. This means that T_B can be 0, 2, 4, or 6 ms, based on the outcome of the random variable.***

c. ***For $K = 3$, the range is $\{0, 1, 2, 3, 4, 5, 6, 7\}$. This means that T_B can be 0, 2, 4, . . . , 14 ms, based on the outcome of the random variable.***

We need to mention that if $K > 10$, it is normally set to 10.

Figure 12.5 *Vulnerable time for pure ALOHA protocol*

- *Vulnerable time*: Length of time in which there is a possibility of collision; for Pure ALOHA it **$2*T_{fr}$**



Example 12.2

A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the requirement to make this frame collision-free?

Solution

Average frame transmission time T_{fr} =200 bits/200 kbps =1 ms.

The vulnerable time = $2 \times T_{fr} = 2 \times 1$ ms = 2 ms.

This means no station should send later than 1 ms before this station starts transmission and no station should start sending during the one 1-ms period that this station is sending.

Throughput of ALOHA

- If '***G***' be the average number of frames generated by the system during one frame transmission time

- Then, average number of successful transmissions for ALOHA is given as: $\mathbf{S = G \times e^{-2G}}$

- The maximum throughput, $\mathbf{S_{max}}$ is 0.184 for $\mathbf{G = ½}$

- i.e. If one-half a frame is generated during 1 frame transmission time (or 1 frame during two frame transmission times), then 18.4% of these reach the destination successfully

Note

The throughput for pure ALOHA is

$$S = G \times e^{-2G}$$

The maximum throughput

$S_{max} = 0.184$ when $G = (1/2)$.

Example 12.3

A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the throughput if the system (all stations together) produces

a. 1000 frames per second b. 500 frames per second

c. 250 frames per second.

Solution

***Frame transmission time* = 200/200 kbps or 1 ms**

a. If system creates 1000 frames per second,

i.e.1 frame per millisecond → load is 1

$S = G \times e^{-2G} = \mathbf{0.135}$ (13.5 percent)

Throughput = 1000 × 0.135 = 135 frames/sec

This means only135 frames out of 1000 will probably survive.

Example 12.3 (continued)

b. If the system creates 500 frames per second, i.e. (1/2) frame per millisecond. The load is (1/2). In this case $S = G \times e^{-2G}$ or $S = 0.184$ (18.4 percent)
This means that the throughput is $500 \times 0.184 = 92$
That is only 92 frames out of 500 will probably survive.

Note that this is the maximum throughput case, percentagewise

c. If the system creates 250 frames per second, this is (1/4) frame per millisecond. The load is (1/4). In this case $S = G \times e^{-2G}$ or $S = 0.152$ (15.2 percent)
This means that the throughput is $250 \times 0.152 = 38$
Only 38 frames out of 250 will probably survive.

Slotted ALOHA

- Time is divided into slots of T_{fr} *sec*
- Stations can only send at the beginning of the time slot
- If a station misses the beginning of a particular time slot, it must wait until the beginning of next time slot
- Possibility of collision, if 2 stations try to send at the beginning of the same slot

- Vulnerable time is reduced to one-half that of pure ALOHA $= T_{fr}$

Figure 12.6 *Frames in a slotted ALOHA network*

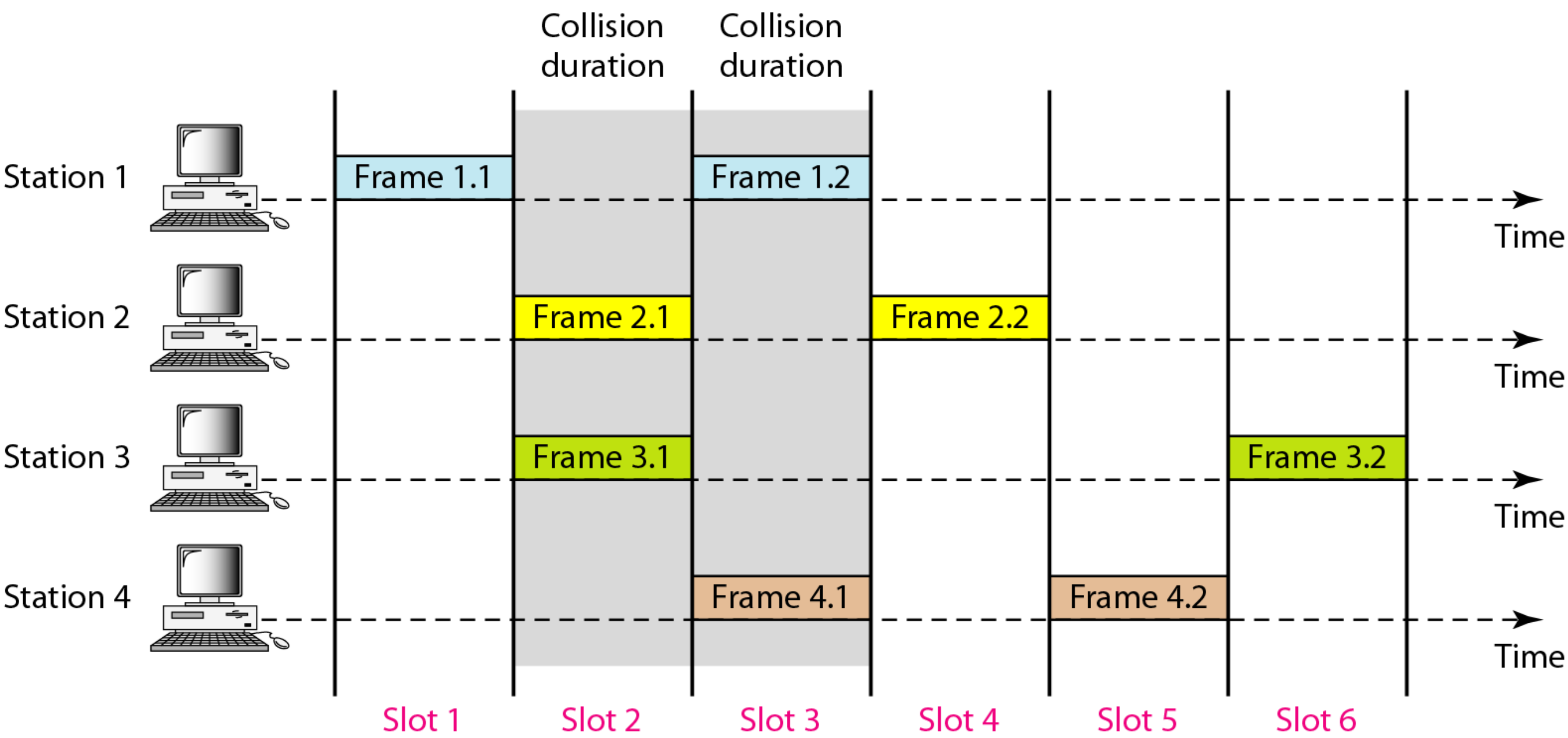
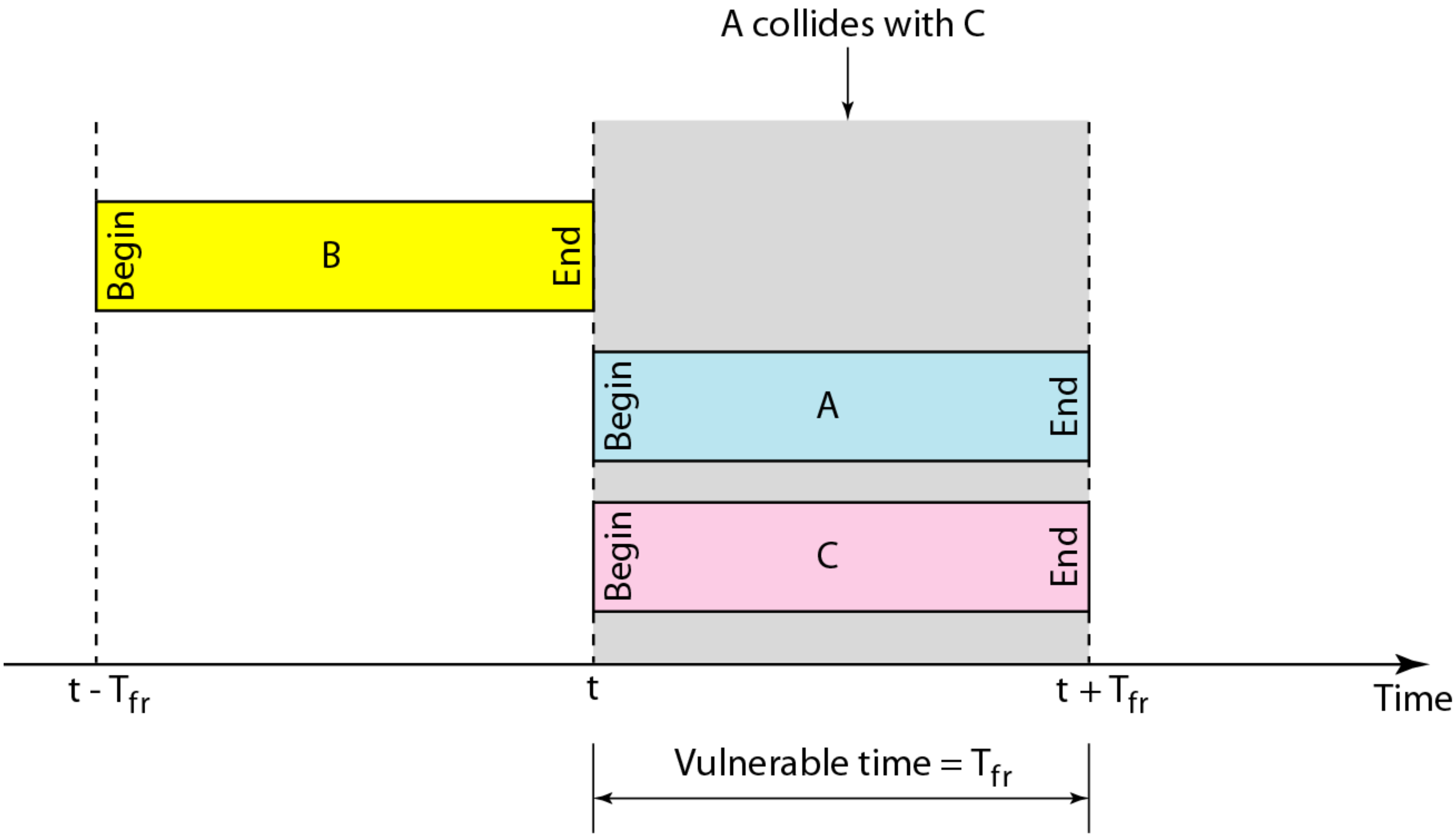


Figure 12.7 *Vulnerable time for slotted ALOHA protocol*



- Vulnerable time for slotted ALOHA is T_{fr}

Note

The throughput for slotted ALOHA is $S = G \times e^{-G}$.
The maximum throughput $S_{max} = 0.368$ when $G = 1$.

Example 12.4

A slotted ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the throughput if the system (all stations together) produces

a. 1000 frames per second b. 500 frames per second

c. 250 frames per second.

Solution

Frame transmission time = 200/200 kbps or 1 ms.

a. If the system creates 1000 frames per second, this is 1 frame per millisecond. The load is 1.

→ $S = G \times e^{-G}$ or $S = 0.368$ (36.8 percent)

Throughput = $1000 \times 0.368 =$ 368 frames

i.e. Only 386 frames out of 1000 will probably survive.

Example 12.4 (continued)

b. If the system creates 500 frames per second, this is (1/2) frame per millisecond. The load is (1/2). In this case $S = G \times e^{-G}$ or $S = 0.303$ (30.3 percent). This means that the throughput is $500 \times 0.303 = 151$. Only 151 frames out of 500 will probably survive.

c. If the system creates 250 frames per second, this is (1/4) frame per millisecond. The load is (1/4). In this case $S = G \times e^{-G}$ or $S = 0.195$ (19.5 percent). This means that the throughput is $250 \times 0.195 = 49$. Only 49 frames out of 250 will probably survive.

Carrier Sense Multiple Access (CSMA)

- Proposed to minimize the chances of collision and thus improve the performance
- Principle: Sense the medium before using it; "Listen before talk"
  - In CSMA, each station first checks the state of the medium before sending
  - If medium idle → transmit; busy medium → wait
- CSMA reduces the possibility of collisions, but cannot completely eliminate it
  - Possibility of collision exists due to propagation delay
  - *a sensing station may find medium to be idle only because the first bit sent by another station has not yet been received*

Figure 12.8 *Space/time model of the collision in CSMA*

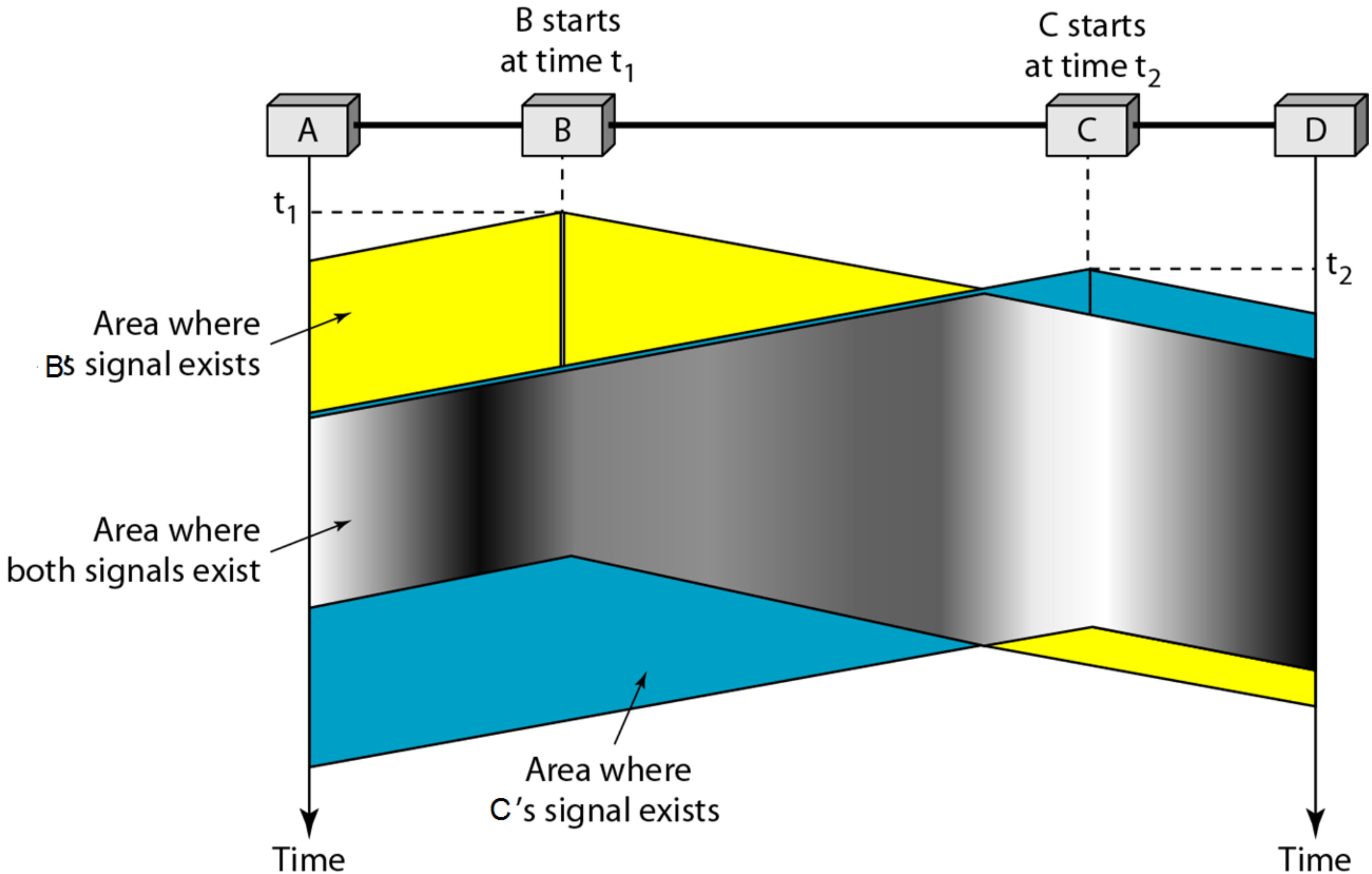
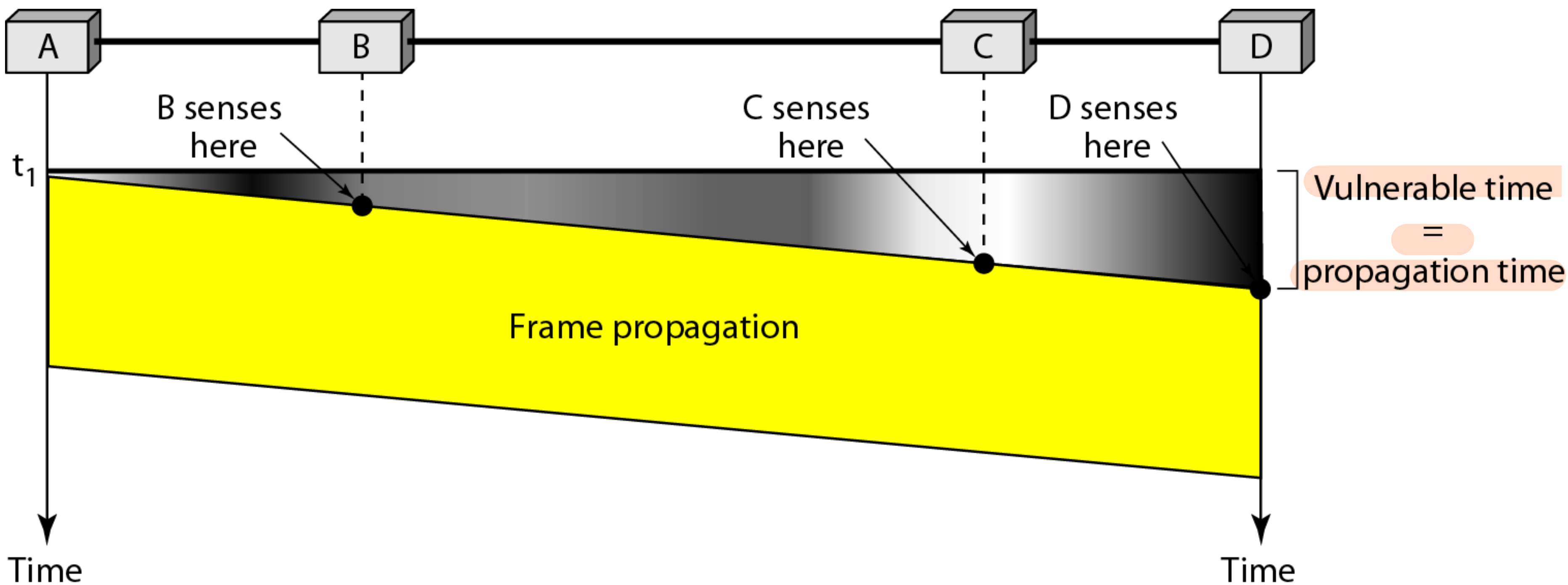


Figure 12.9 *Vulnerable time in CSMA*



- Vulnerable time for CSMA is the propagation time T_p

Carrier Sense: Persistence Methods

- These methods tell what should a station do if the channel is busy or idle
- ***1-persistent:***
- If a station senses channel is idle, it sends its frame immediately with probability '1'
  - Highest chance of collision, since 2 or more stations may find channel is idle and send their frames at once
- ***Nonpersistent:***
- If the channel is idle the station sends immediately; else waits random amount of time and senses the channel again
  - Reduces the chance of collision; its unlikely that 2 or more stations will wait same amount of time and retry to send

Carrier Sense: Persistence Methods contd…

- ***p-persistent:***
- Combines advantages of other 2 approaches to reduce the chance of collision and improves efficiency
- Used if the channel has time slots with slot duration => maximum propagation time
- After the station finds the channel is idle, it takes the following steps:
  1. **With probability *p*, the station sends its frame**
  2. **With probability *q* = (1 - *p*), the station waits for the beginning of next time slot and senses the channel again**
  3. **If line idle; goes to step 1**
  4. **If line busy, it acts as a collision has occurred, and uses the back-off procedure**

Persistence Methods: Figure 12.10 *Behavior of three persistence methods*

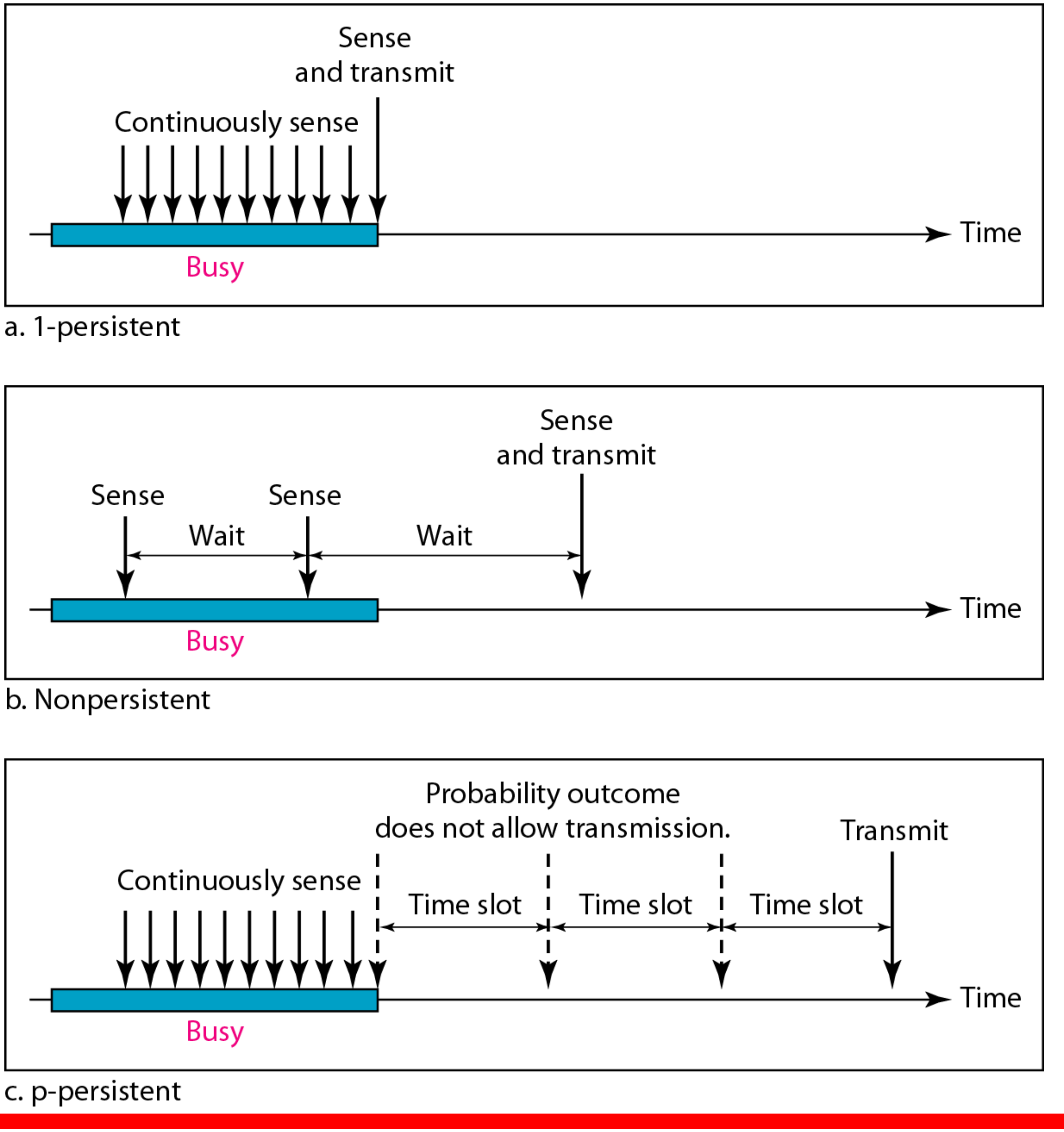
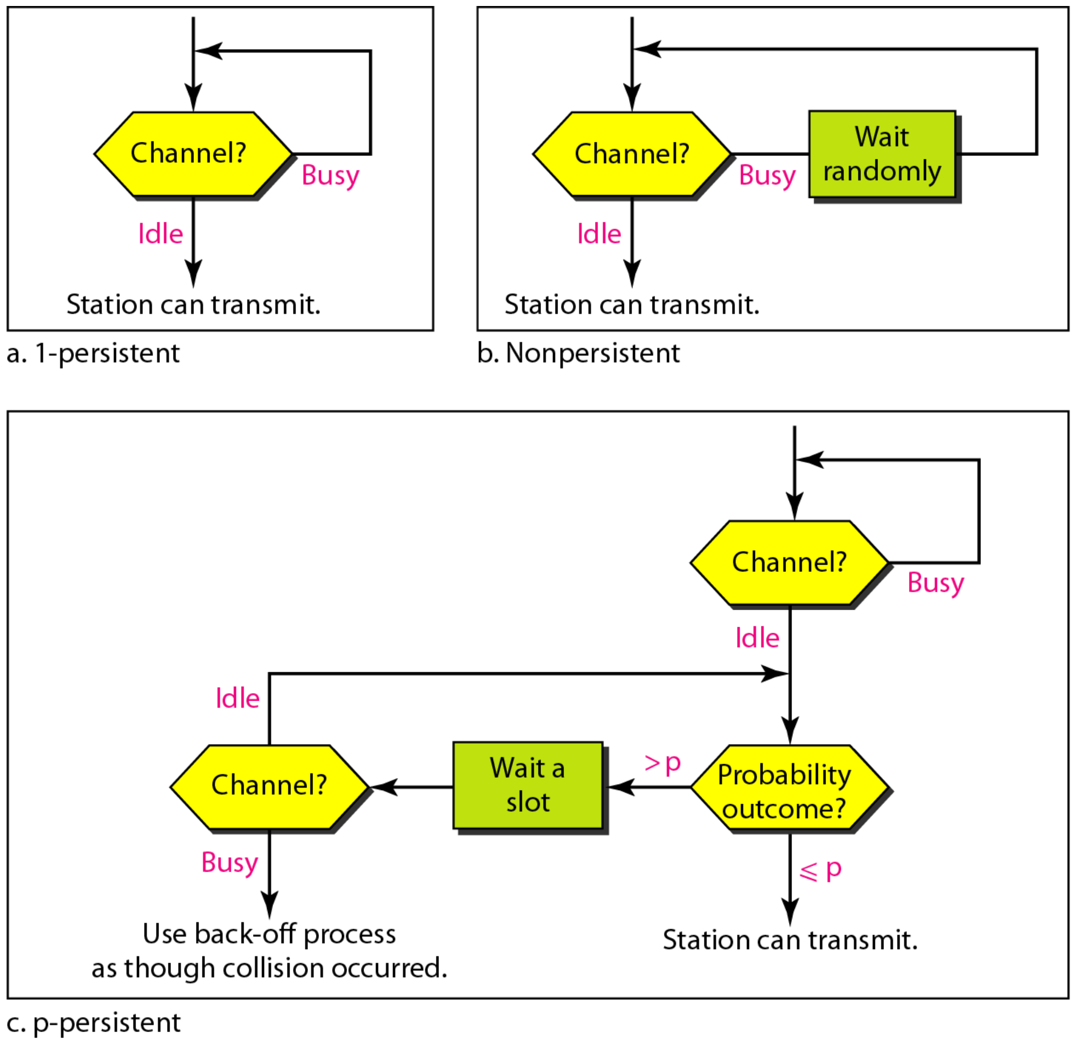


Figure 12.11 *Flow diagram for three persistence methods*

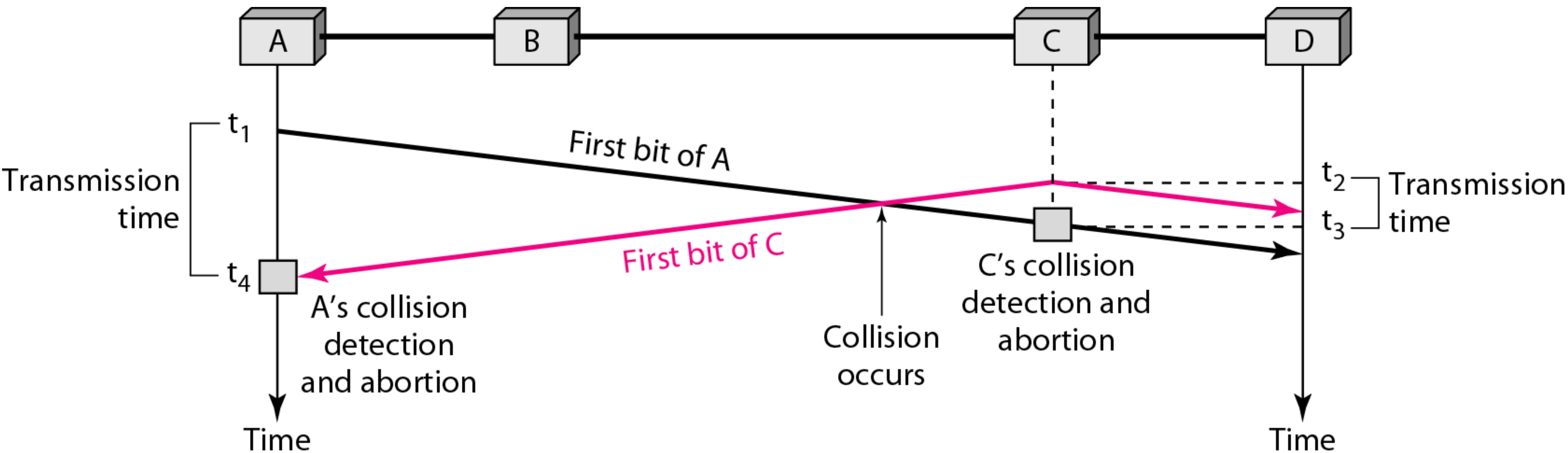


Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

- CSMA does not specify what to do after collision
- CSMA/CD augments the CSMA algorithm to handle collision

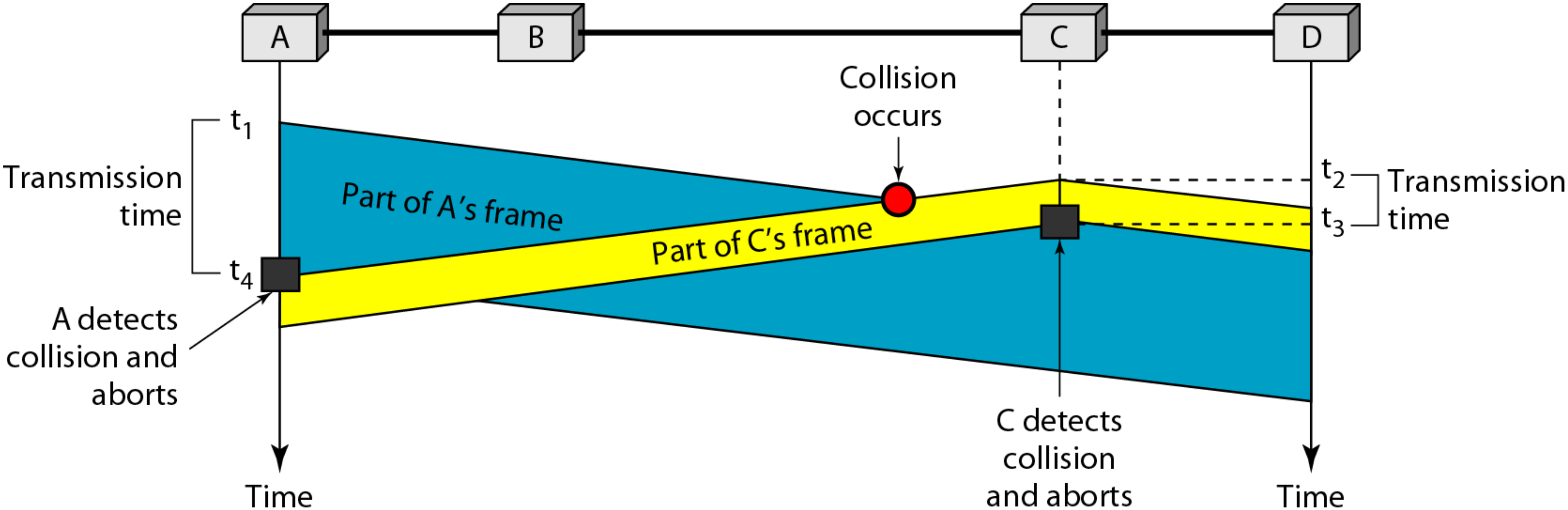
- Principle: Stations monitor the medium after sending a frame
  - If transmission successful → done; collision → resend the frame

Figure 12.12 *Collision of the first bit in CSMA/CD*



- At t_1, station ***A***, after executing its persistence procedure starts sending bits of its frame
- At t_2, station ***C*** has not yet sensed the first bit sent by ***A***; executes its persistence procedure and starts sending
- Collision occurs at some time after t_2
- ***C*** detects the collision at time t_3 and *A* detects the collision at time t_4 ; immediately abort transmission

Figure 12.13 *Collision and abortion in CSMA/CD*



Minimum frame size

- For CSMA/CD collision detection mechanism to work, the sending station must be able to hear a collision, before sending the last bit of the frame
- If two stations involved in the collision are maximum distance apart, the signal from the first station takes time T_p to reach the second, and the collision takes another time T_p to reach the first

- Therefore, the Frame transmission time T_{fr} must be at least two times the maximum propagation time T_p ($T_{fr} >= 2T_p$)

- ***Throughput***
- Maximum throughput is based on the persistence method
- 1-*persistent*: Max throughput around 50% when G=1
- *Non-persistent*: can be upto 90%, when G is between 3 & 8

Example 12.5

A network using CSMA/CD has a bandwidth of 10 Mbps. If the maximum propagation time (including the delays in the devices) is 25.6 μs, what is the minimum size of the frame?

Solution

The frame transmission time, $T_{fr} = 2 \times T_p = 51.2\ \mu s$.

i.e. In the worst case, a station needs to transmit for a period of 51.2 μs to detect the collision

→ Minimum size of the frame = 10 Mbps × 51.2 μs = 512 bits or 64 bytes

(This is the minimum size of the frame for Standard Ethernet)

Figure 12.14 *Flow diagram for the CSMA/CD*

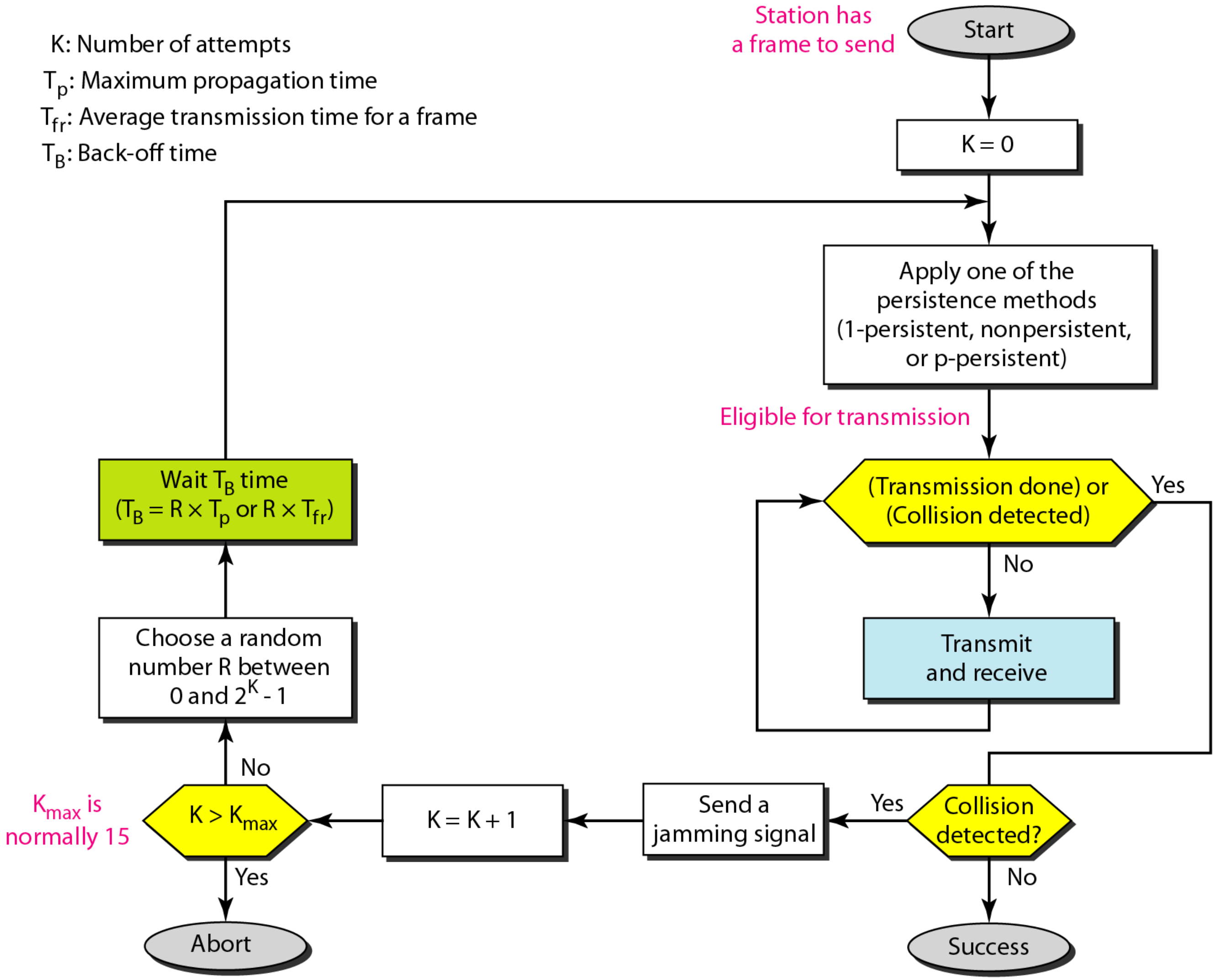
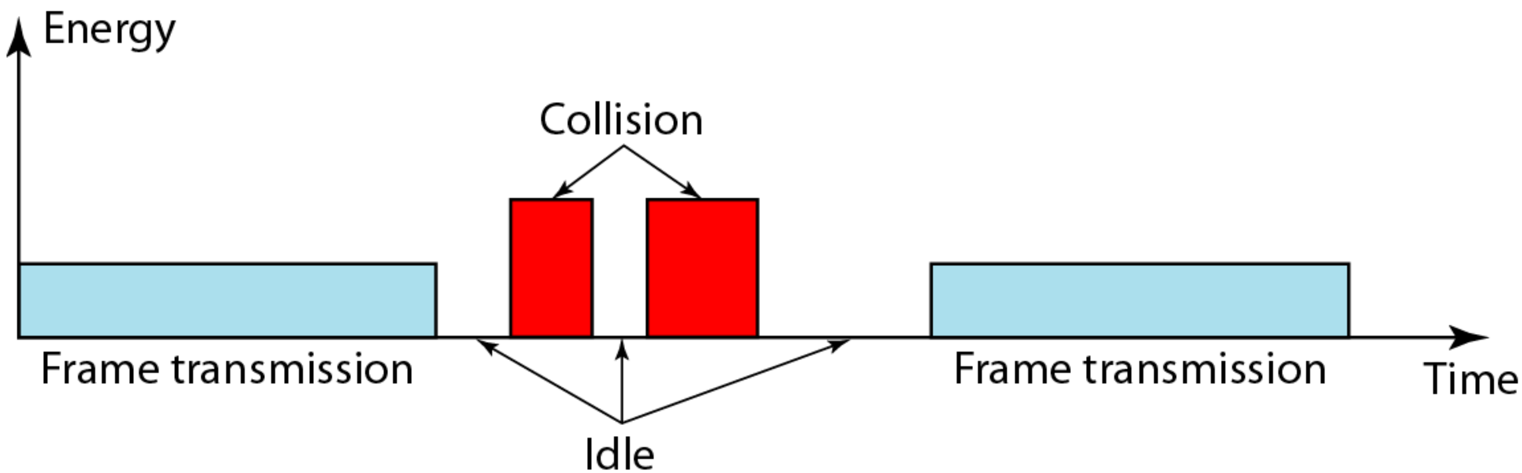


Figure 12.15 *Energy level during transmission, idleness, or collision*

Energy Level on a channel: ***Zero*** *(channel idle),* ***Normal*** *(station successfully captured the channel and sending its frame), and* ***Abnormal*** *(collision ; level of energy is twice the normal level)*

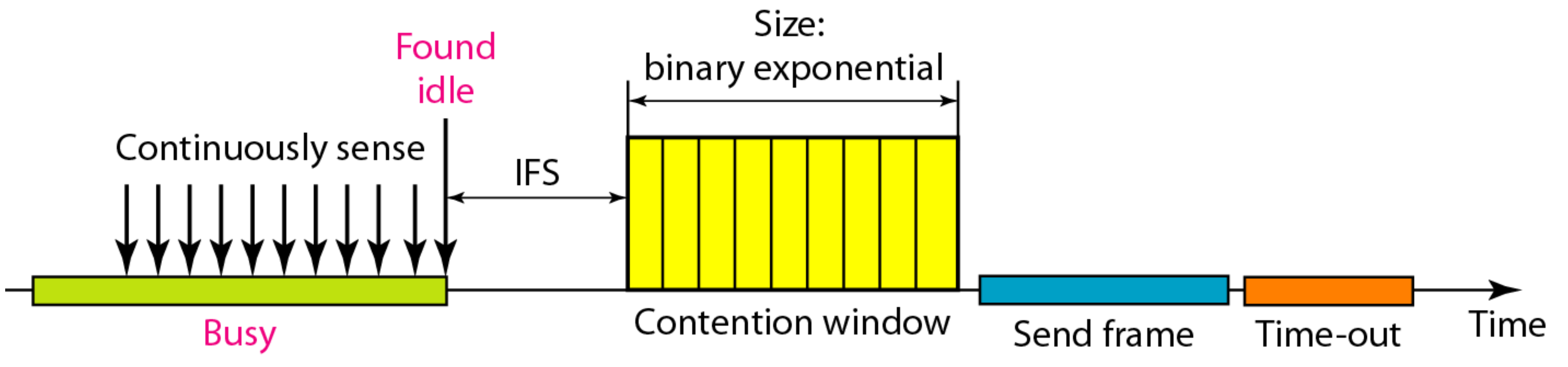


Carrier Sense Multiple Access with Collision Avoidance(CSMA/CA)

- For collision detection, significant energy needs to be added to the transmitted signal
  - CSMA/CD works well in wired networks, since the energy is almost double when collision occurs
- In wireless networks, much of the energy is lost in transmission (inverse square law)→ received signal has very little energy

  →Collision may add only 5 to 10% additional energy; thus collisions cannot be detected
- Need to avoid collisions→ CSMA/CA
- Collisions avoided through 3 strategies: the inter-frame space (IFS), contention window, and acknowledgements

Figure 12.16 *Timing in CSMA/CA*



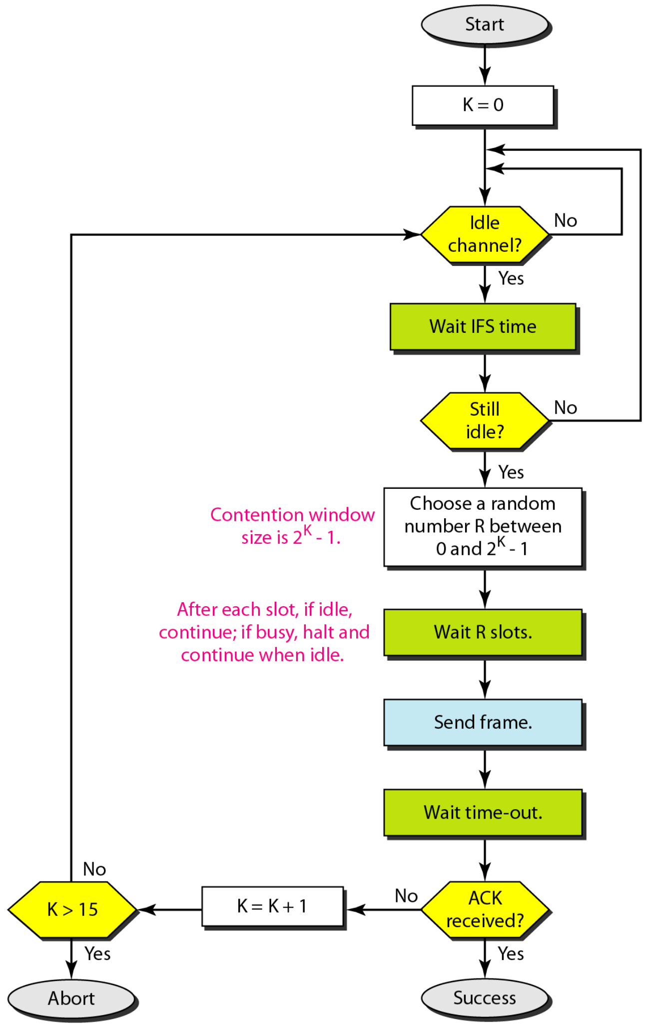
Note

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

Note

In CSMA/CA, if the station finds the channel busy, it does not restart the timer of the contention window; it stops the timer and restarts it when the channel becomes idle.

Figure 12.17 *Flow diagram for CSMA/CA*



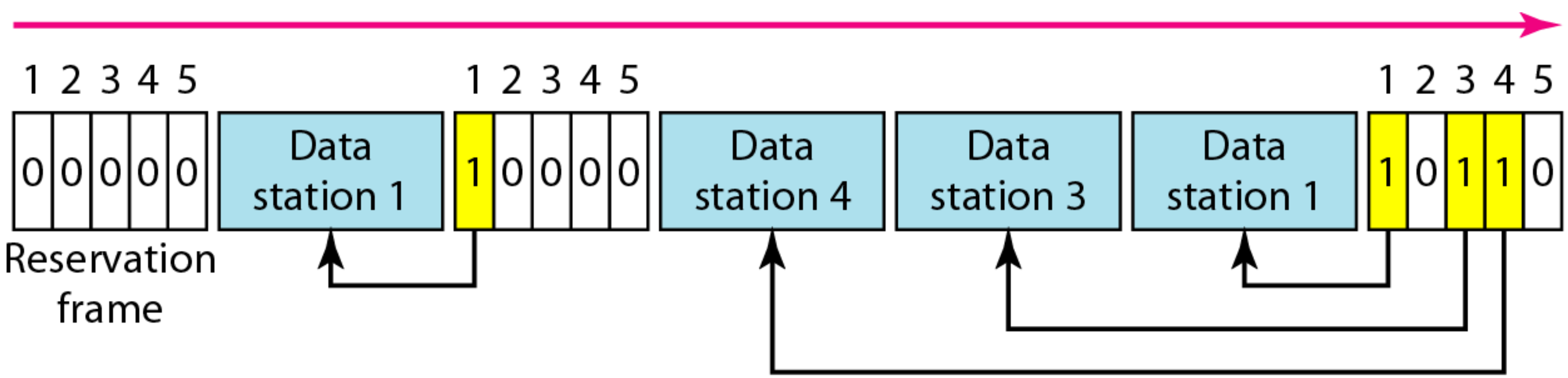
12-2 CONTROLLED ACCESS

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. We discuss three popular controlled-access methods.

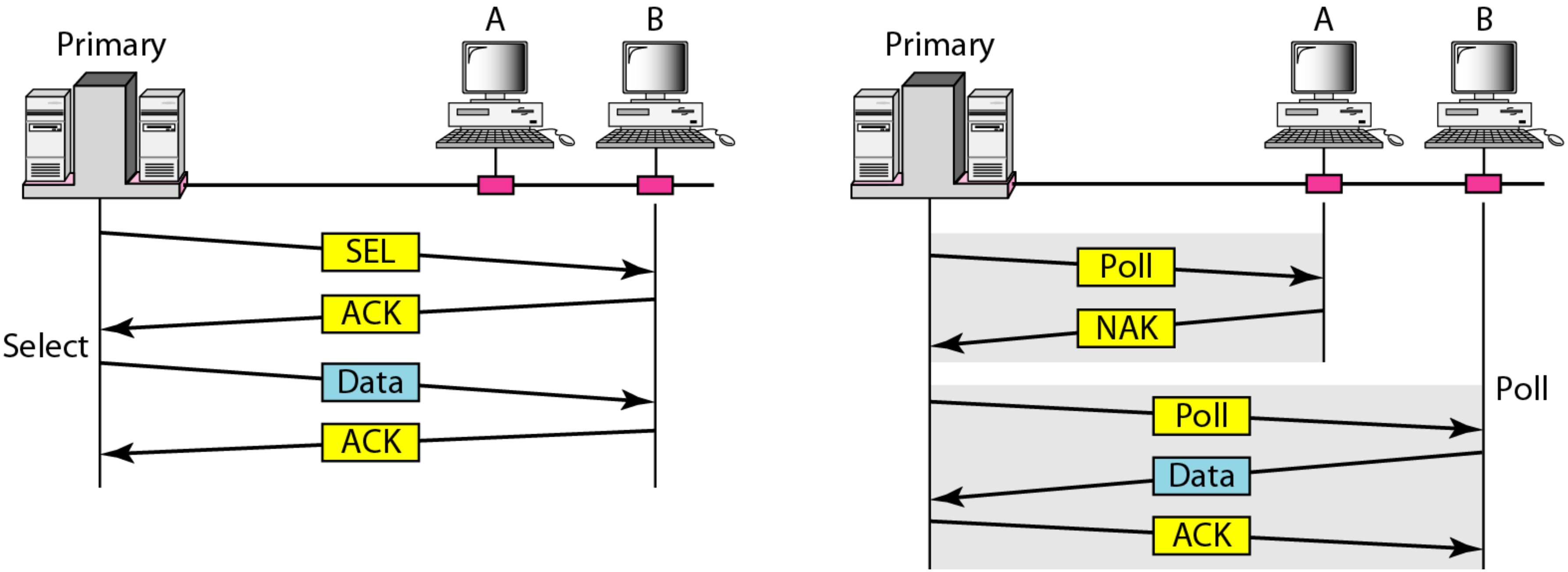
Topics discussed in this section:

Reservation
Polling
Token Passing

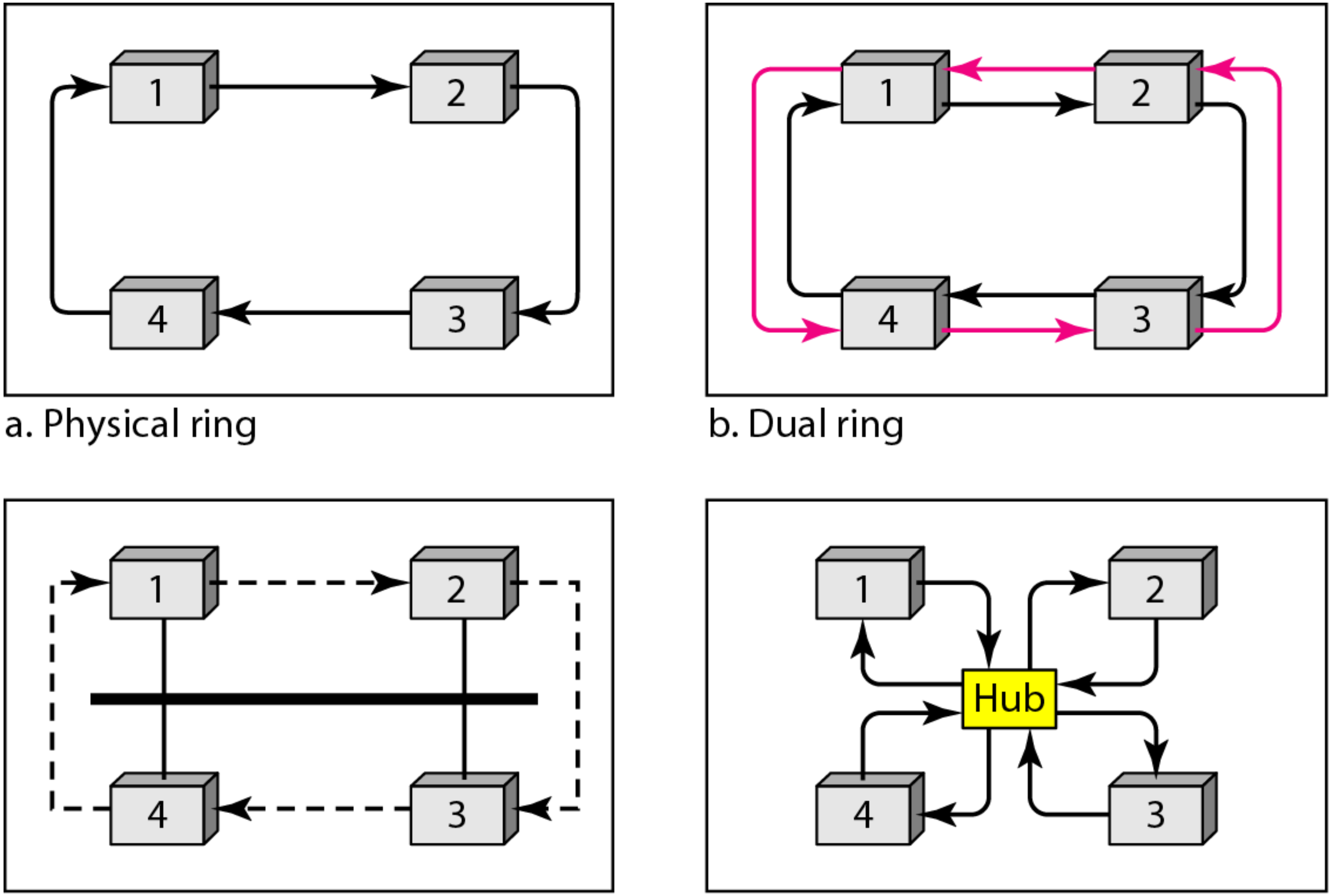
Reservation: Figure 12.18 ***Reservation access method***



Polling: Figure 12.19 *Select and poll functions in polling access method*



Token Passing: Fig 12.20 ***Logical ring and physical topology in token-passing***



a. Physical ring

b. Dual ring

c. Bus ring

d. Star ring

12-3 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.
-Three channelization protocols.

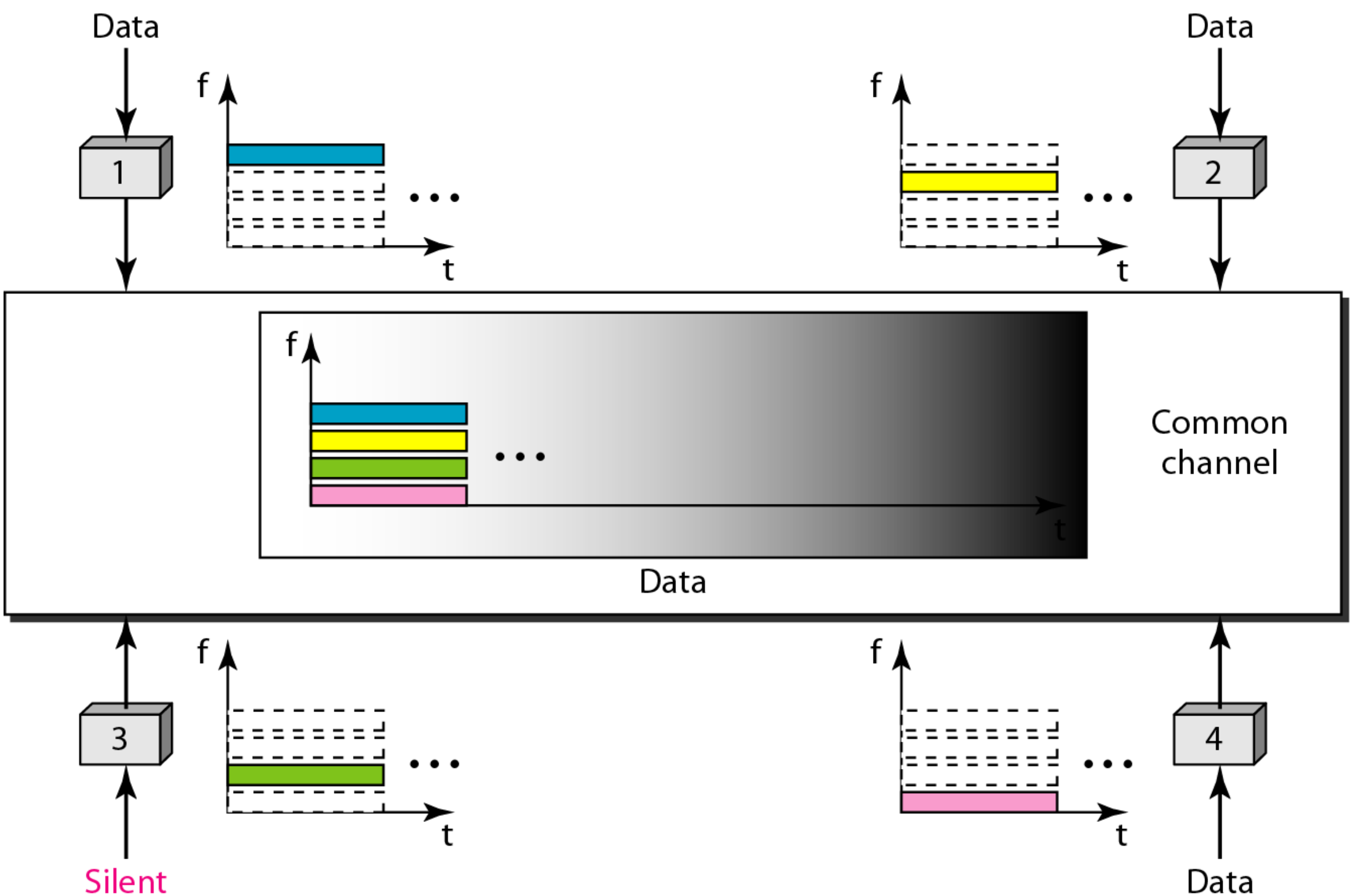
Topics discussed in this section:

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

Note

We see the application of all these methods in cellular phone systems

Figure 12.21 *Frequency-division multiple access (FDMA)*

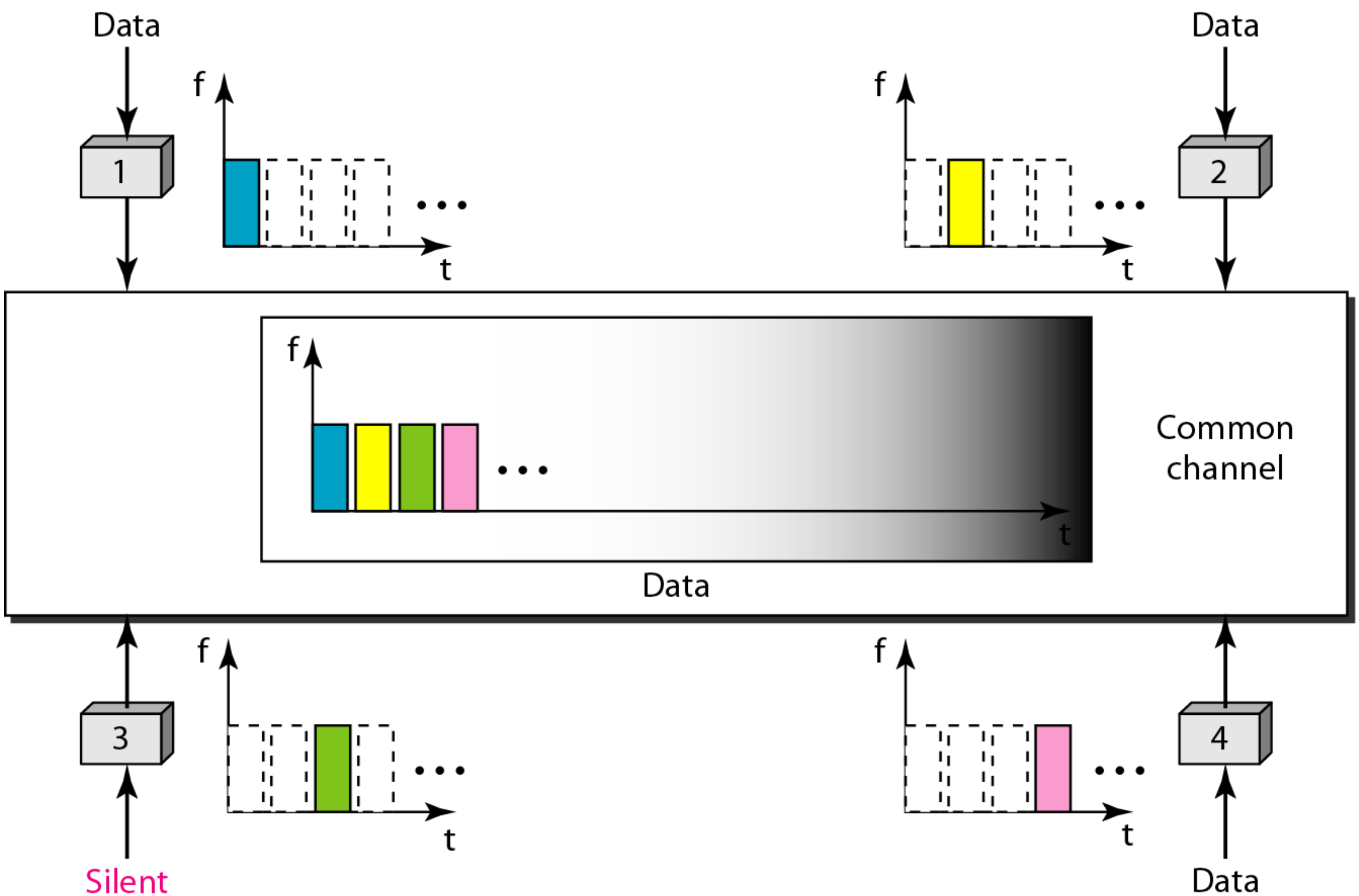


Note

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

FDMA specifies a predetermined frequency band for the entire period of communication

Figure 12.22 *Time-division multiple access (TDMA)*



Note

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

Note

In CDMA, one channel occupies the entire BW of the link, and carries all transmissions simultaneously.

Figure 12.23 *Simple idea of communication with code*

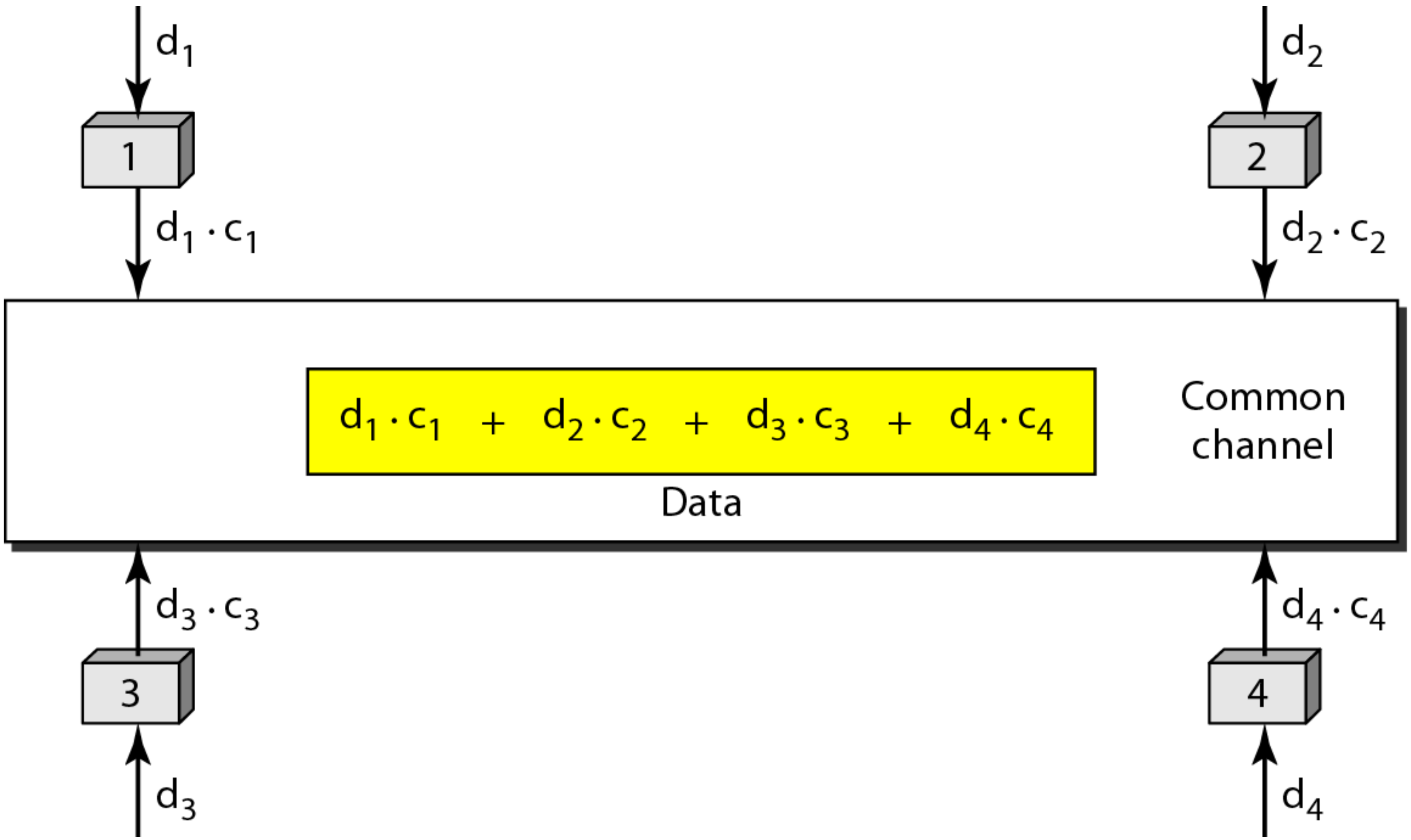


Figure 12.24 *Chip sequences*

C_1	C_2	C_3	C_4
[+1 +1 +1 +1]	[+1 -1 +1 -1]	[+1 +1 -1 - 1]	[+1 -1 -1 +1]

Figure 12.25 *Data representation in CDMA*

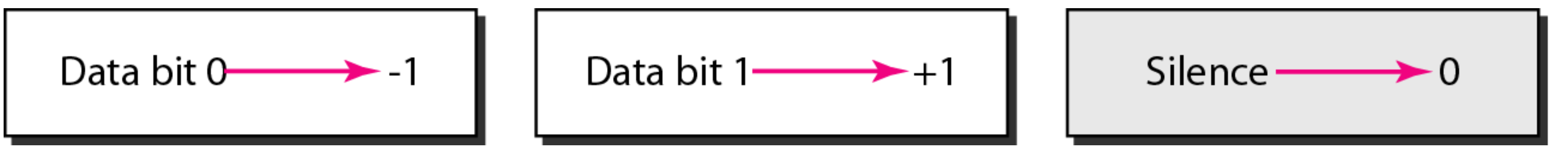


Figure 12.26 *Sharing channel in CDMA*

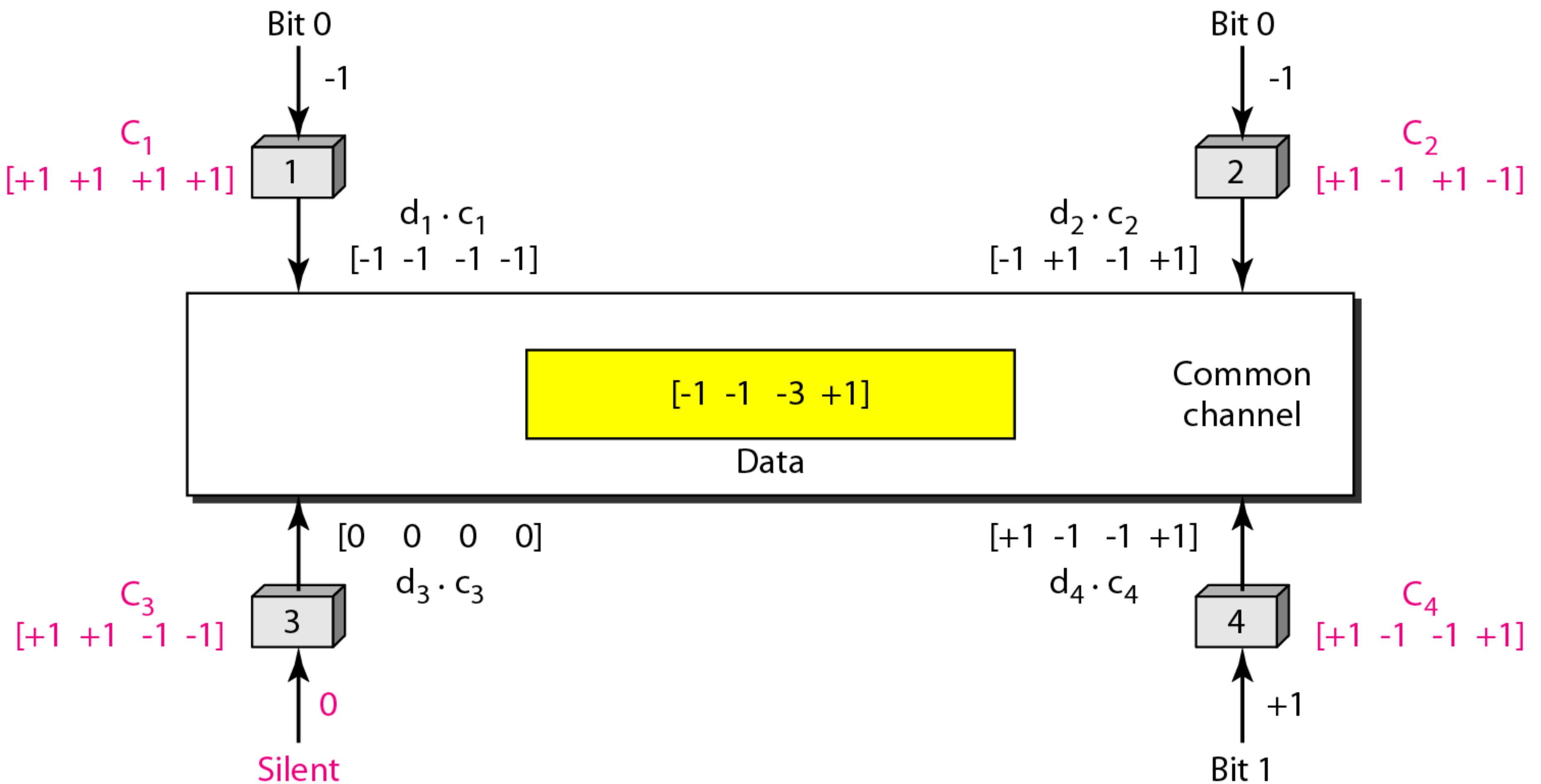


Figure 12.27 *Digital signal created by four stations in CDMA*

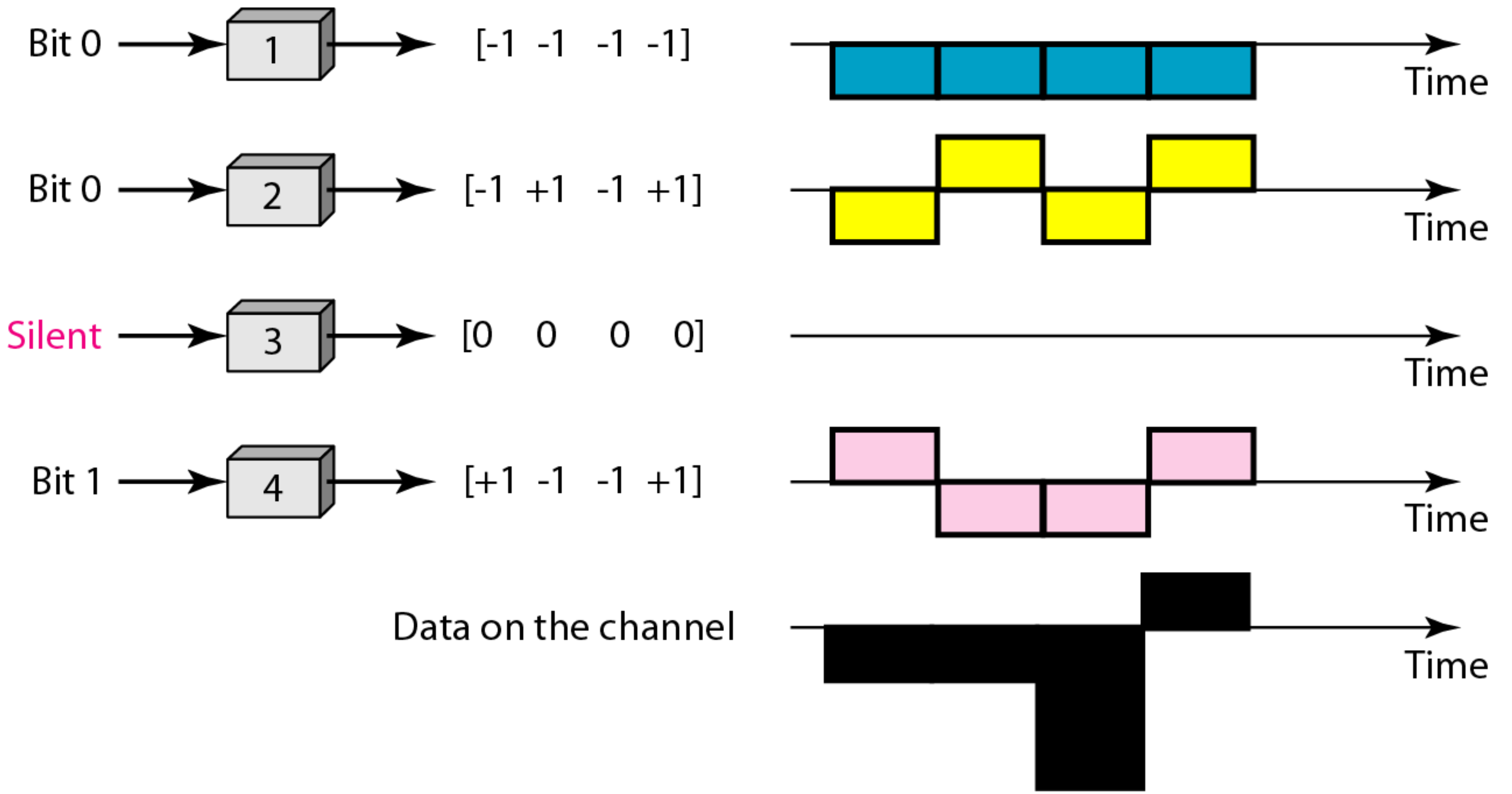
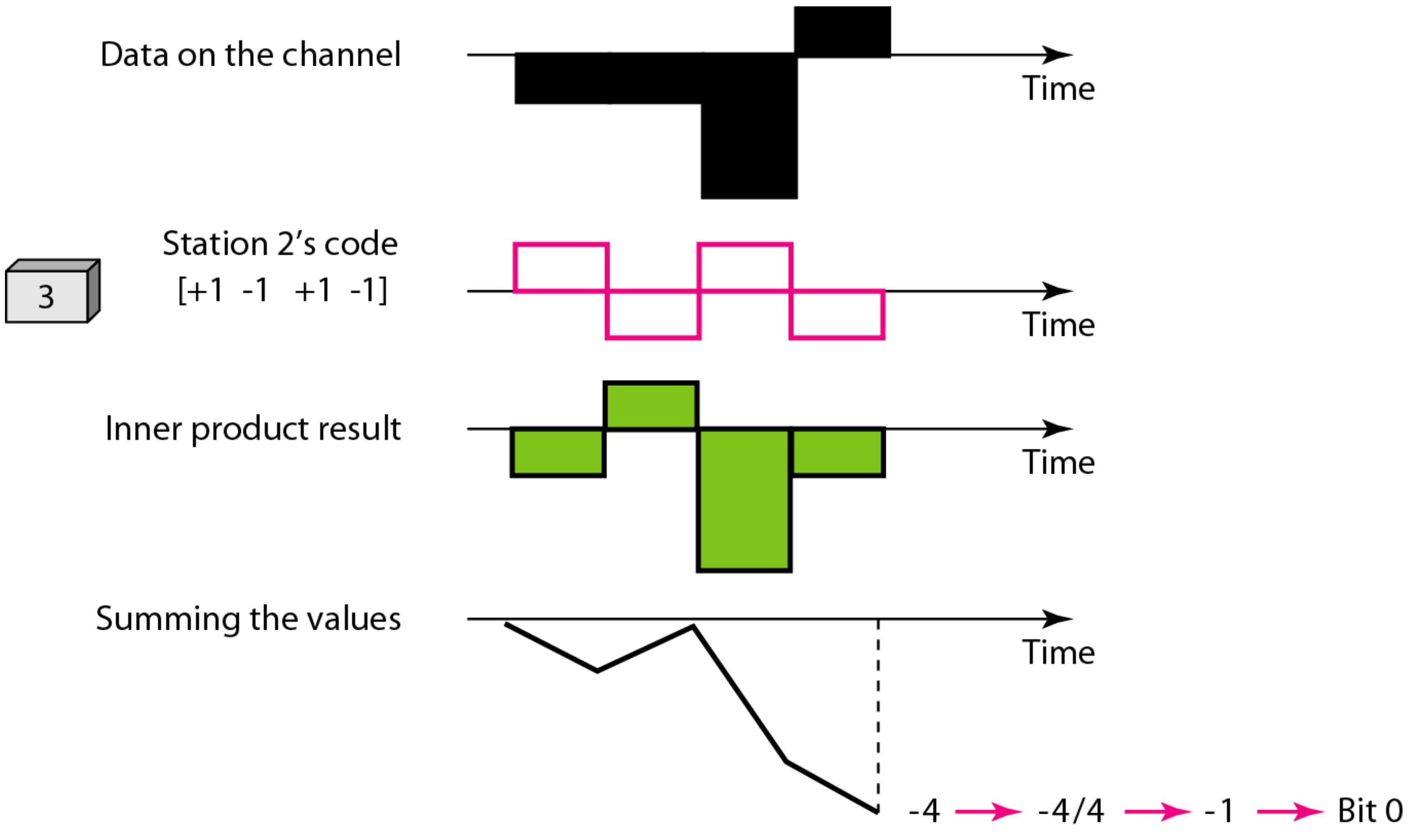


Figure 12.28 *Decoding of the composite signal for one in CDMA*



Sequence Generation:

Figure 12.29 *General rule and examples of creating Walsh tables*

$$W_1 = \begin{bmatrix} +1 \end{bmatrix} \qquad W_{2N} = \begin{bmatrix} W_N & W_N \\ W_N & \overline{W_N} \end{bmatrix}$$

a. Two basic rules

$$W_1 = \begin{bmatrix} +1 \end{bmatrix}$$

$$W_2 = \begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix} \qquad W_4 = \begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{bmatrix}$$

b. Generation of W_1, W_2, and W_4

Note

The number of sequences in a Walsh table needs to be $N = 2^m$.

Example 12.6

Find the chips for a network with

a. Two stations ***b. Four stations***

Solution

We can use the rows of W_2 and W_4 in Figure 12.29:

a. For a two-station network, we have

[+1 +1] and [+1 −1].

b. For a four-station network we have

[+1 +1 +1 +1], [+1 −1 +1 −1],

[+1 +1 −1 −1], ***and*** **[+1 −1 −1 +1].**

Example 12.7

What is the number of sequences if we have 90 stations in our network?

Solution

The number of sequences needs to be 2^m. We need to choose $m = 7$ and $N = 2^7$ or 128. We can then use 90 of the sequences as the chips.

Example 12.8

Prove that a receiving station can get the data sent by a specific sender if it multiplies the entire data on the channel by the sender's chip code and then divides it by the number of stations.

Solution

Let us prove this for the first station, using our previous four-station example. We can say that the data on the channel

$$D = (d_1 \cdot c_1 + d_2 \cdot c_2 + d_3 \cdot c_3 + d_4 \cdot c_4)$$

Example 12.8 (continued)

The receiver which wants to get the data sent by station 1 multiplies these data by c_1.

$$
\begin{aligned}
D \cdot c_1 &= (d_1 \cdot c_1 + d_2 \cdot c_2 + d_3 \cdot c_3 + d_4 \cdot c_4) \cdot c_1 \\
&= d_1 \cdot c_1 \cdot c_1 + d_2 \cdot c_2 \cdot c_1 + d_3 \cdot c_3 \cdot c_1 + d_4 \cdot c_4 \cdot c_1 \\
&= d_1 \times N + d_2 \times 0 + d_3 \times 0 + d_4 \times 0 \\
&= d_1 \times N
\end{aligned}
$$

When we divide the result by N, we get d_1.