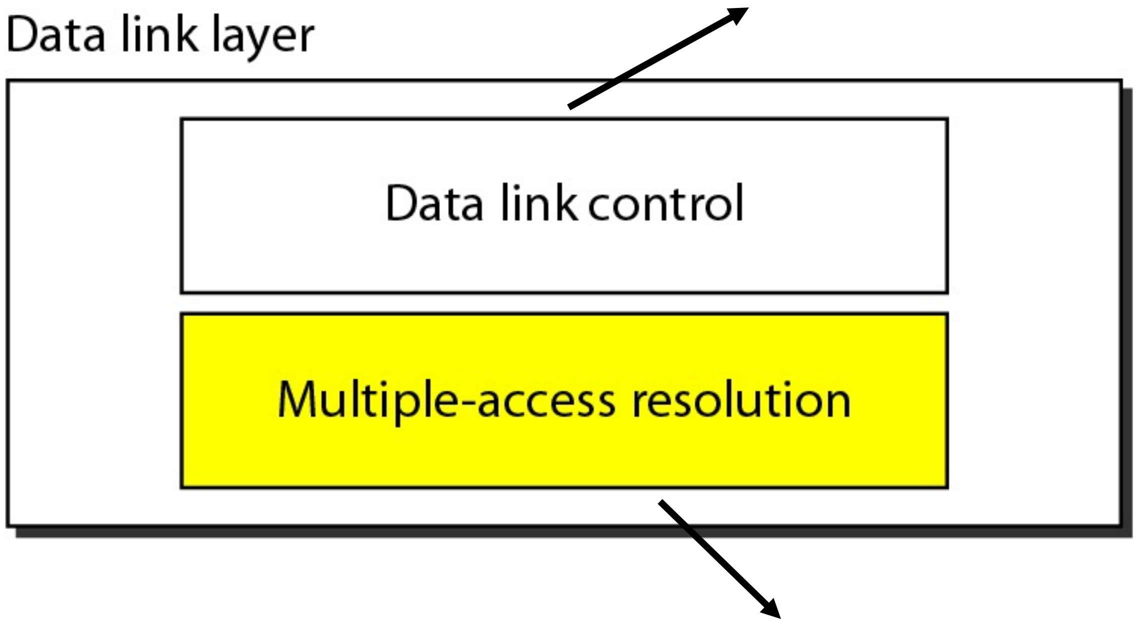
Chapter 12

Multiple Access

Partially Edited and
Presented by
Dr. Md. Abir Hossain

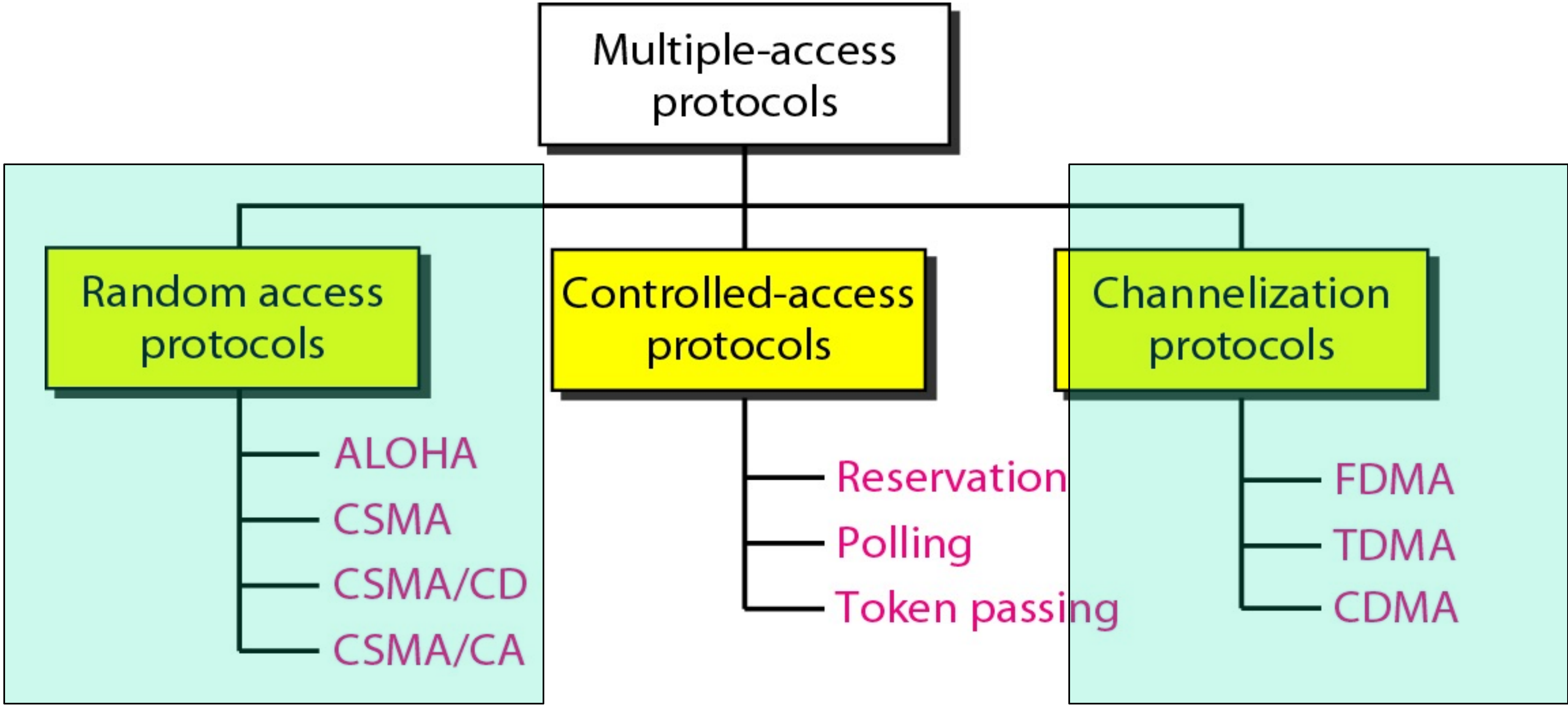
Figure 12.1 *Data link layer divided into two functionality-oriented sublayers*

Responsible for flow and error control is called the **Logical Link Control (LLC)** layer



Responsible for multiple- access resolution is called the **Media Access Control (MAC)** layer

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

- ALOHA, the earliest random access method, was developed at the University of Hawaii in early 1970.
- ALOHA can be classified as **PURE ALOHA**, and **slotted ALOHA**
- The original ALOHA protocol is called **pure ALOHA**
- The idea is that each station **sends a frame whenever it has a frame to send.**
- Since there is only one channel to share, there is the possibility of collision between frames from different stations.
  - Using short range radios.
  - Half duplex by nature. At a time, only can send or receiver. Switching also takes time.
  - Two different frequencies, one for sending, another for receiving.
  - But, problem of collision, how to solve it?
  - Solution: Let the users communicate, if signals collide, not acknowledged and so, sender resends data.
  - Adding randomness reduces the chance of collision.
  - Algorithm is called Binary Exponential Back-off Algorithm.
  - Also had problem: While transmitting, sender can not sense collision.
  - In ALOHA, maximum **18 out of 100** packets pass without collision if ALOHA works with optimum speed.

Figure 12.3 *Frames in a pure ALOHA network*

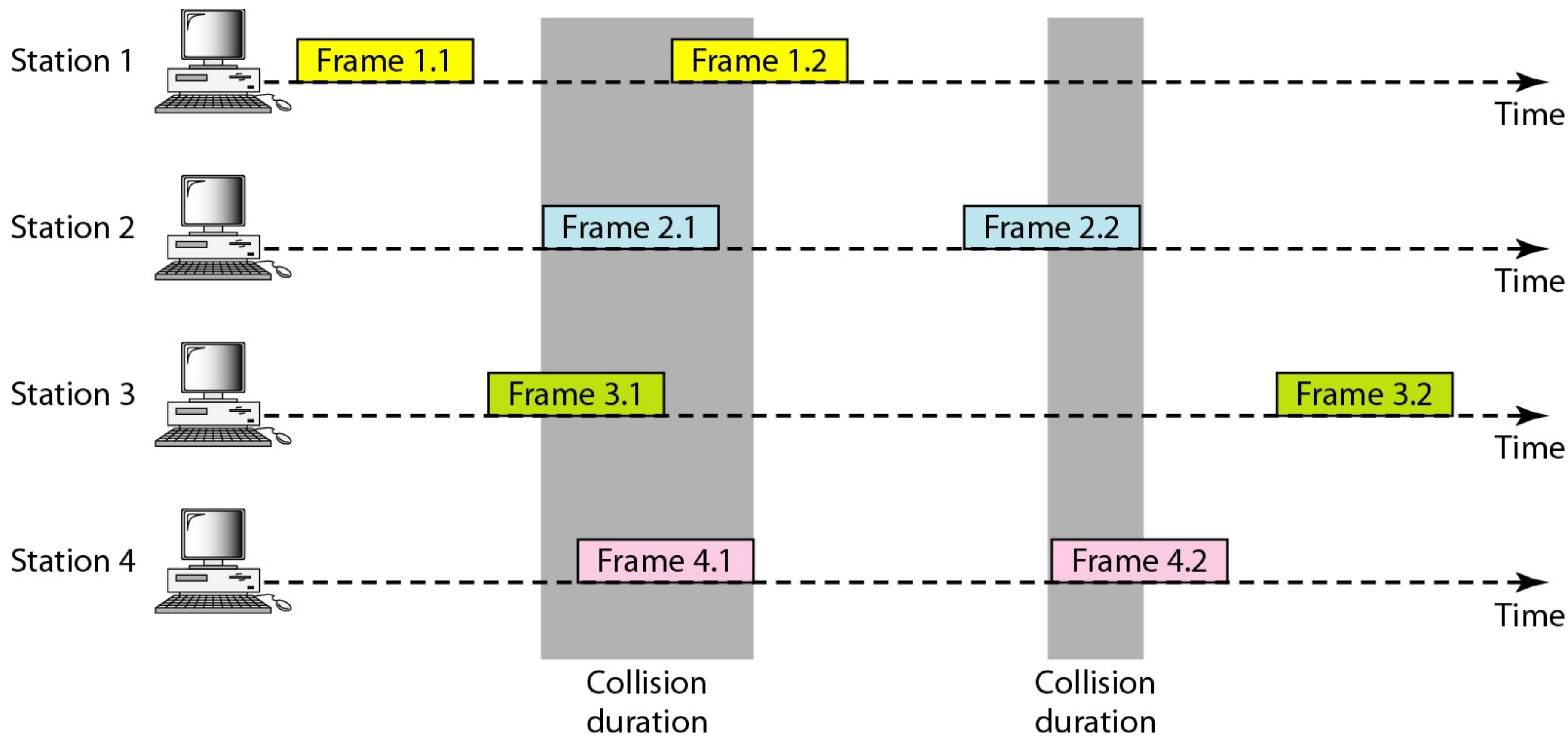


Figure 12.4 *Procedure for pure ALOHA protocol*

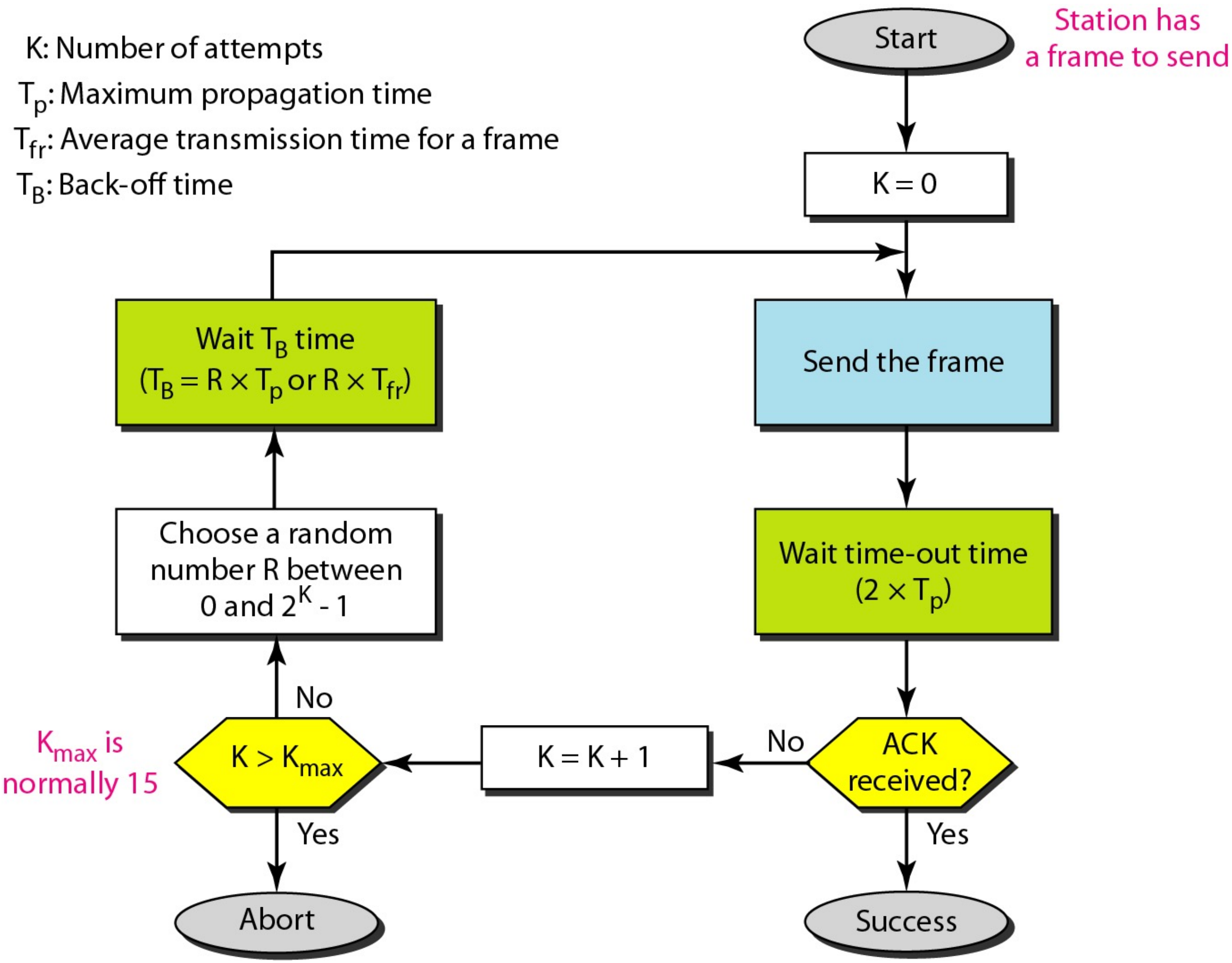
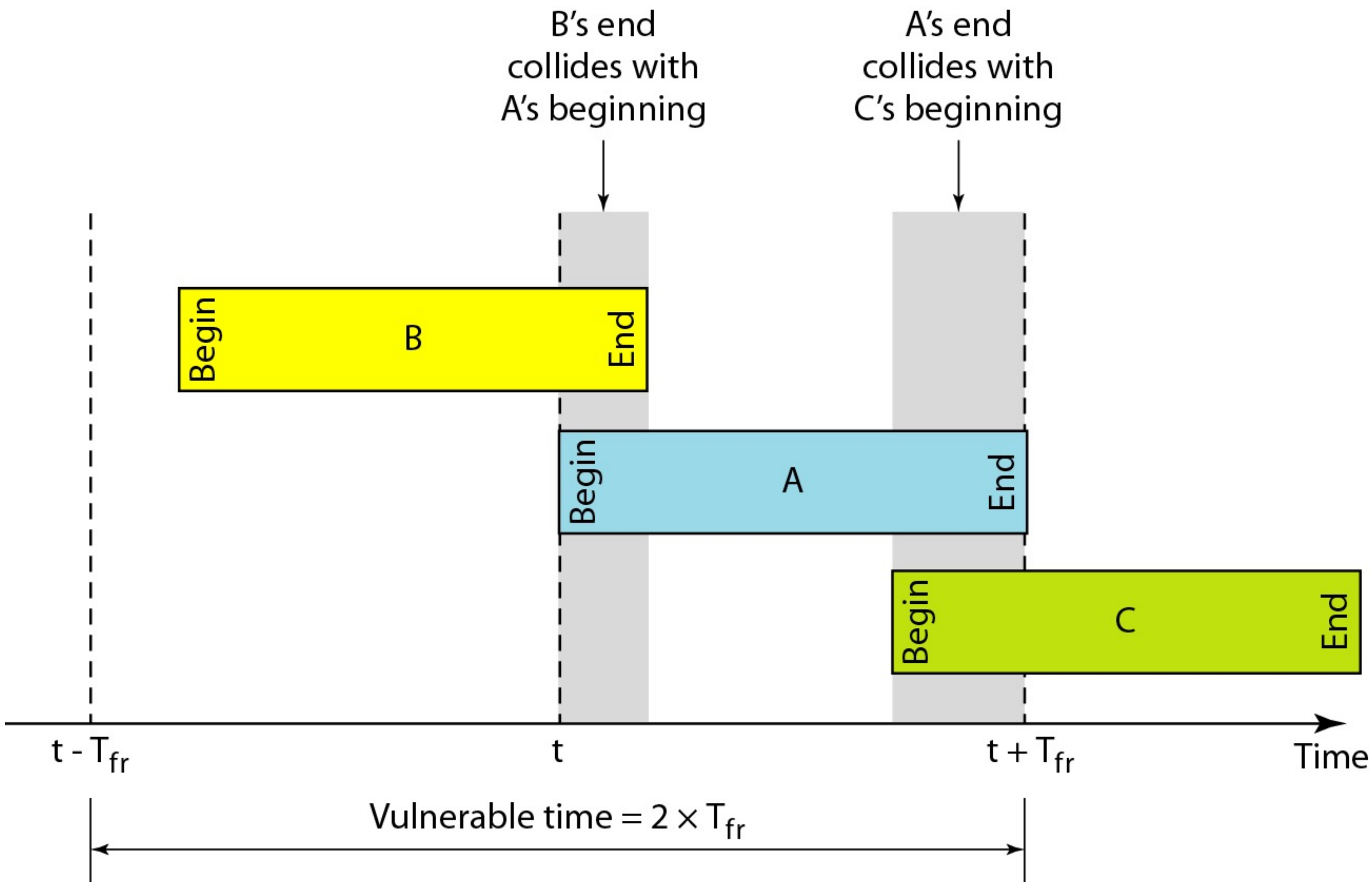


Figure 12.5 *Vulnerable time for pure ALOHA protocol*



pure ALOHA vulnerable time is 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} is 200 bits/200 kbps or 1 ms. The vulnerable time is 2×1 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.

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

The frame transmission time is 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. In this case $S = G \times e^{-2G}$ or $S = 0.135$ (13.5 percent). This means that the throughput is $1000 \times 0.135 = 135$ frames. Only 135 frames out of 1000 will probably survive.

Example 12.3 (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 $S = G \times e^{-2G}$ or $S = 0.184$ (18.4 percent). This means that the throughput is $500 \times 0.184 = 92$ and that 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

- Pure ALOHA has a vulnerable time of $2 \times T_{fr}$. This is so because there is no rule that defines when the station can send.
- Solution: Slotted ALOHA
  - Robert, in 1972 proposed a scheme.
  - Packets are vulnerable to collide with only those packets which were transmitted before, but not during the lifetime.
  - In slotted ALOHA time is divided into fixed intervals, each corresponding to one frame. That means timeslot=lifetime of packets.
  - A computer is not permitted to send whenever it has data to send. Instead it is required to wait for the next available slot and transmitted only in beginning of next slot only.
  - Slotted ALOHA introduces additional delay.
  - Eg : B is to be transmitted during A's lifetime, B will be delayed till next slot.
  - Thus, reducing collision probability to half and performance is doubled.
  - In slotted ALOHA, **36 out of 100 packets** are delivered without collision at optimum speed.

Figure 12.6 *Frames in a slotted ALOHA network*

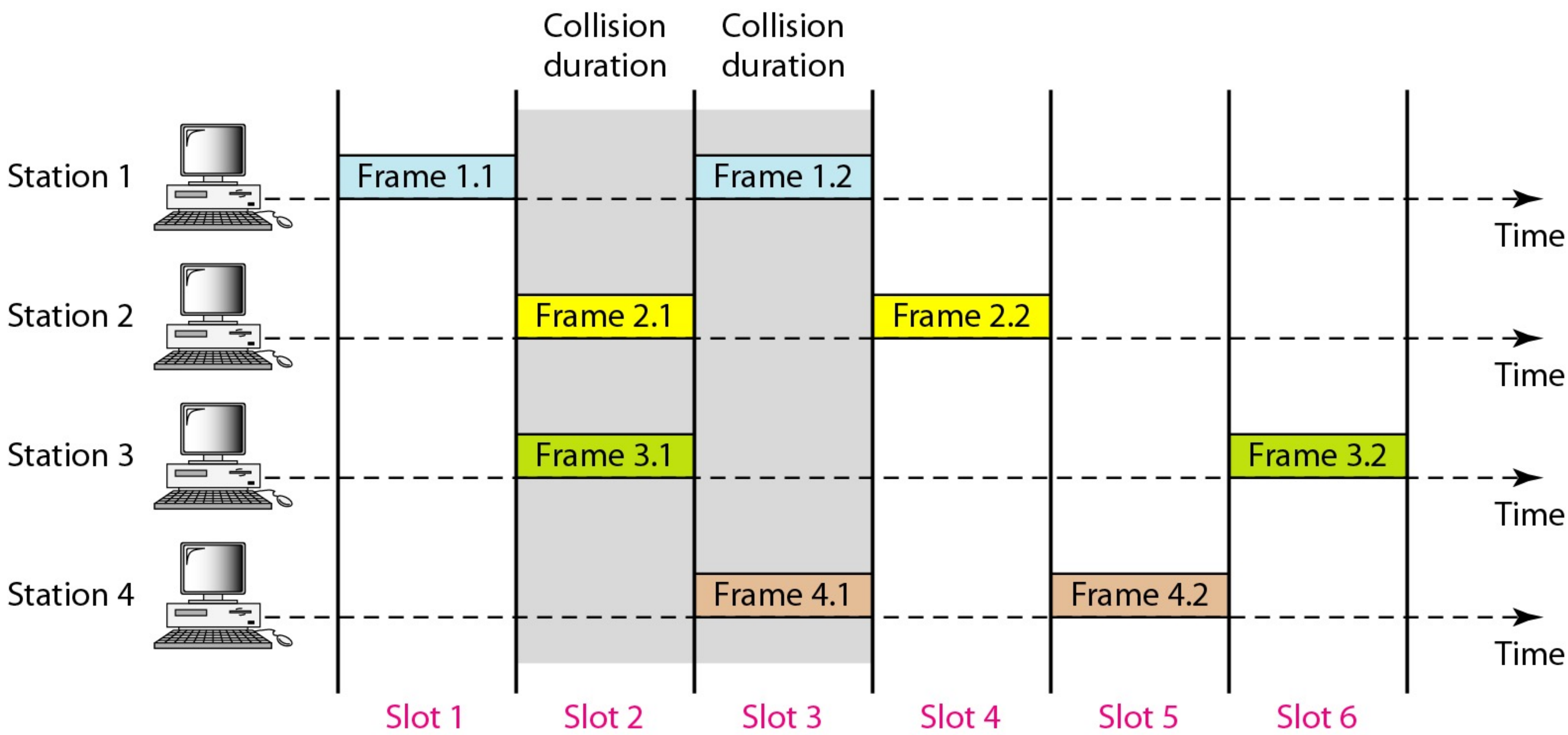
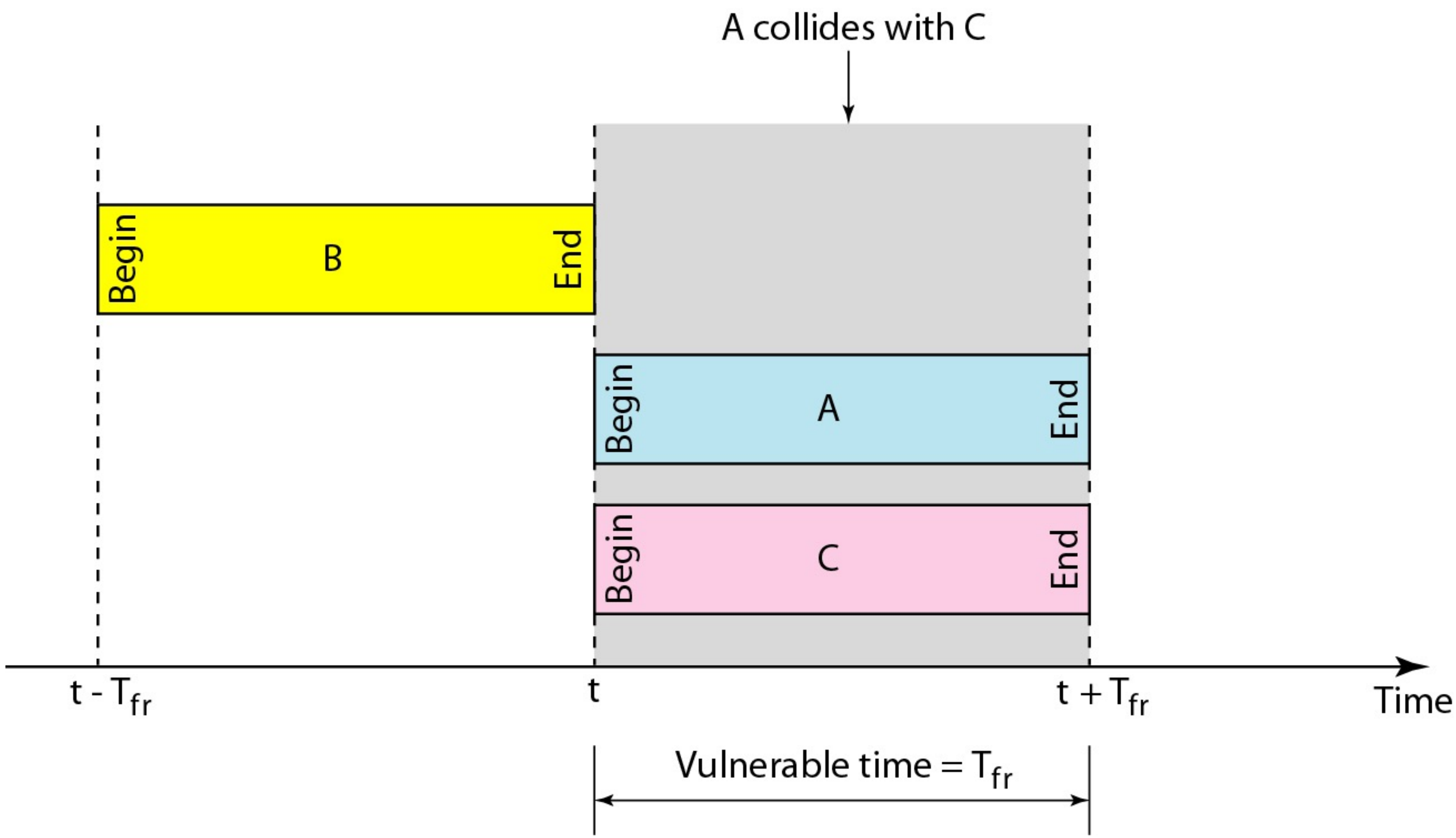


Figure 12.7 *Vulnerable time for slotted ALOHA protocol*



slotted ALOHA vulnerable time 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

The frame transmission time is 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. In this case $S = G \times e^{-G}$ or $S = 0.368$ (36.8 percent). This means that the throughput is $1000 \times 0.0368 = 368$ frames. Only 386 frames out of 1000 will probably survive.

Example 12.4 (continued)

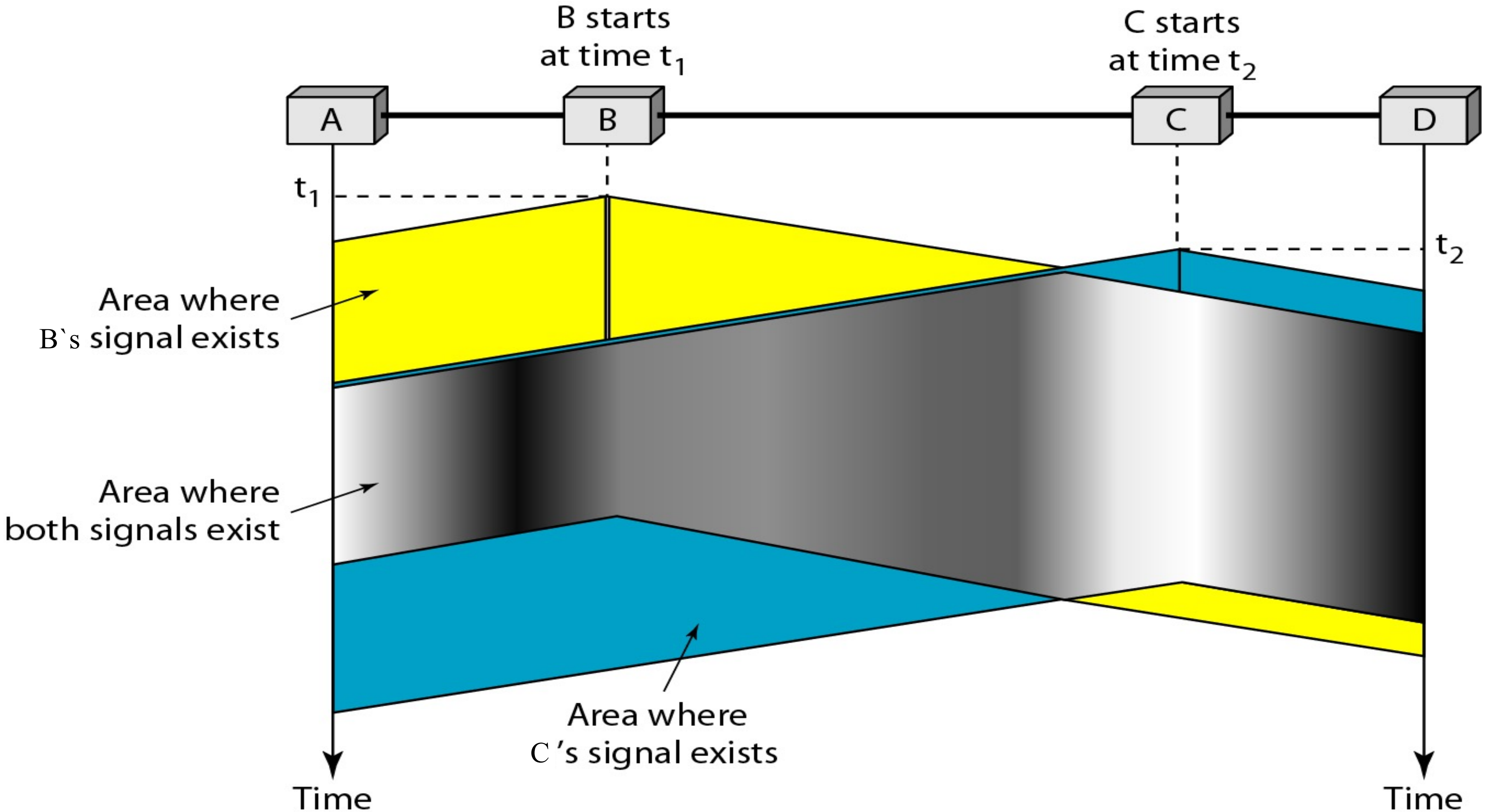
b. ***If the system creates 500 frames per second, this is (1/2) frame per millisecond. The load is (1/2). In this case $S = G \times e^{-G}$ or $S = 0.303$ (30.3 percent). This means that the throughput is $500 \times 0.0303 = 151$. Only 151 frames out of 500 will probably survive.***

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)

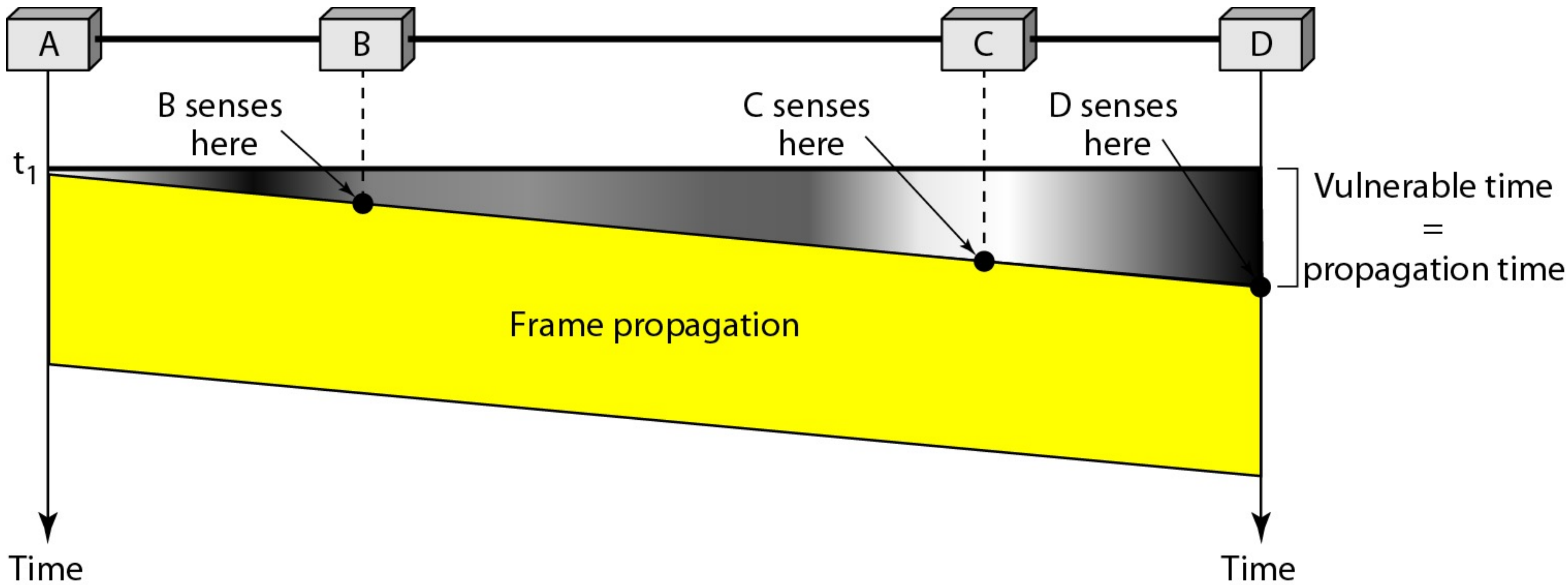
- The previous protocol (Pure and slotted ALOHA) cannot sense the station before transmit. Due to this collision is high.
- CSMA is based on the principle “sense before transmit” or “listen before talk.”
- Carrier sense multiple access (CSMA) requires that each station first listen to the medium before sending.
- CSMA can reduce the possibility of collision due to sensing but still collision exist because of **propagation delay**
- A station may sense the medium and find it 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*



- At time t1, station B senses the medium and finds it idle, so it sends a frame.
- At time t2 (t2 > t1), station C senses the medium and finds it idle because, at this time, the first bits from station B have not reached station C.
- Station C also sends a frame. The two signals collide and both frames are destroyed.

Figure 12.9 *Vulnerable time in CSMA*



- The vulnerable time for CSMA is the **propagation time Tp**.
- This is the time needed for a signal to propagate from one end of the medium to the other.
- When a station sends a frame and any other station tries to send a frame during this time, a collision will result.
- But if the first bit of the frame reaches the end of the medium, every station will already have heard the bit and will refrain from sending.
- The leftmost station, A, sends a frame at time t1, which reaches the rightmost station, D, at time t1 + Tp. The gray area shows the vulnerable area in time and space.

CSMA:

- TYPES:
- 1 Persistent CSMA
- Non Persistent CSMA
- P- Persistent CSMA
- CSMA/CD

Carrier Sense Multiple Access (CSMA)

- Protocols in which stations listen for a carrier (i.e. transmission) and act accordingly are called **carrier sense protocols**.

1. 1-persistent CSMA

   Channel Busy ➔ Continue sensing until free and then grab.

   Channel Idle ➔ Transmit with probability 1.

   Collision ➔ Wait for a random length of time and try again.

2. Non-persistent CSMA:

   Channel Busy ➔ Does not continually sense the channel. Wait for a random length of time and try again.

   Channel Idle ➔ Transmit.

   Collision ➔ Wait for a random length of time and try again.

3. P-persistent CSMA:

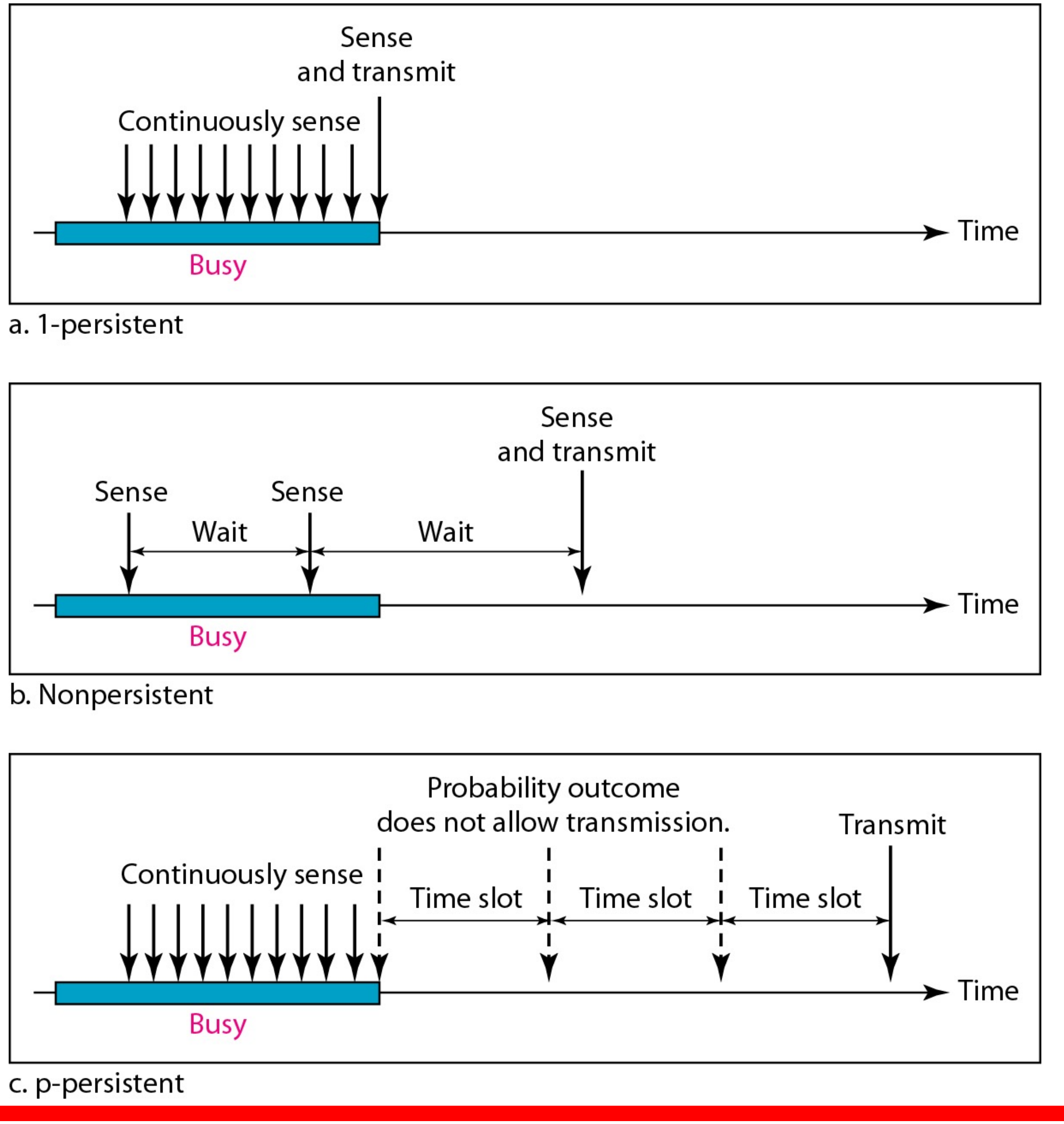
Channel Busy ➔ Continue sensing until free (same as idle).

Channel Idle ➔ Transmit with probability p, and defer transmitting until the next slot with probability q = 1-p.

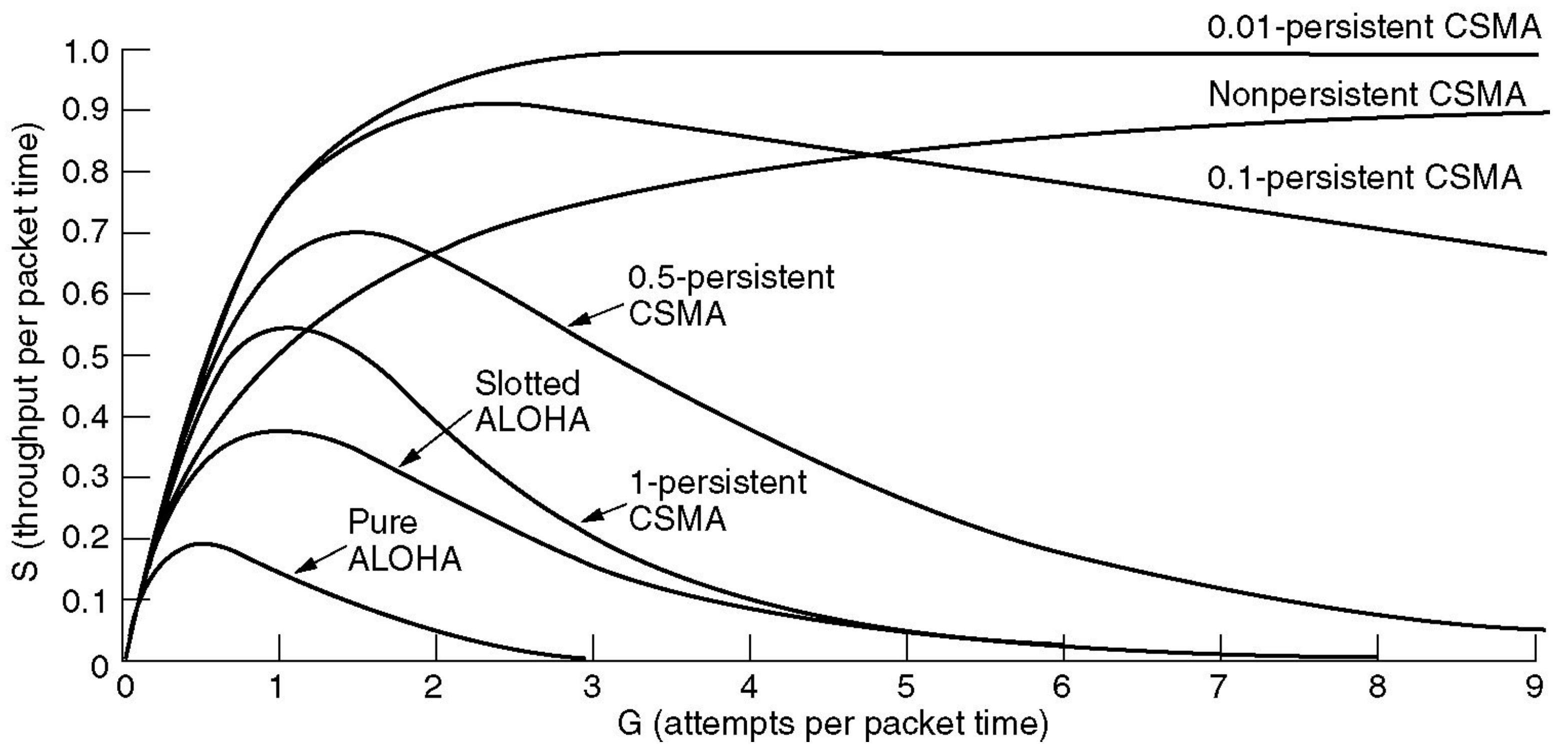
Collision ➔ Wait for a random length of time and try again.

- **Analysis:**
  - The non-persistent CSMA has better channel utilization but longer delays than 1-persistent CSMA.
  - CSMA are an improvement over ALOHA because they ensure that no station begins to transmit when it senses the channel busy.
  - Another improvement is for stations to abort their transmissions as soon as they detect a collision.
  - Quickly terminating damaged frames saves time and bandwidth.
  - This protocol is called CSMA/CD (CSMA with Collision Detection).

Figure 12.10 *Behavior of three persistence methods*



Persistent and Nonpersistent CSMA



Comparison of the Throughput versus load for various random access protocols.

CSMA/Collision Detection(CD)

- In this method, a station monitors the medium after it sends a frame to see if the transmission was successful.
- Collision Detection: Protocol listen when transmission is going on and stop transmitting when it finds colliding.
- The frame transmission time T_{fr} must be at least two times the maximum propagation time as 2Tp

Figure 12.14 *Flow diagram for the CSMA/CD*

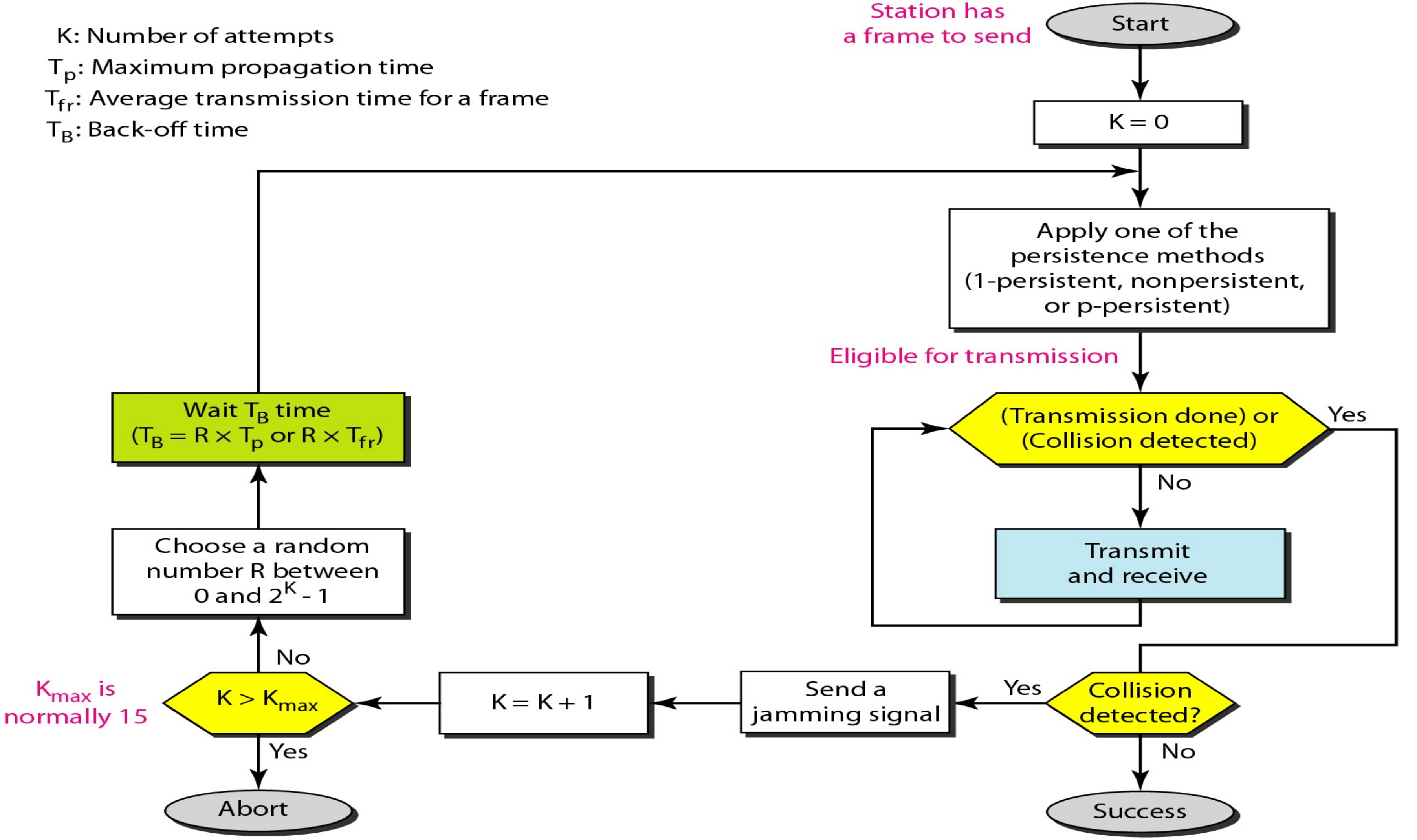
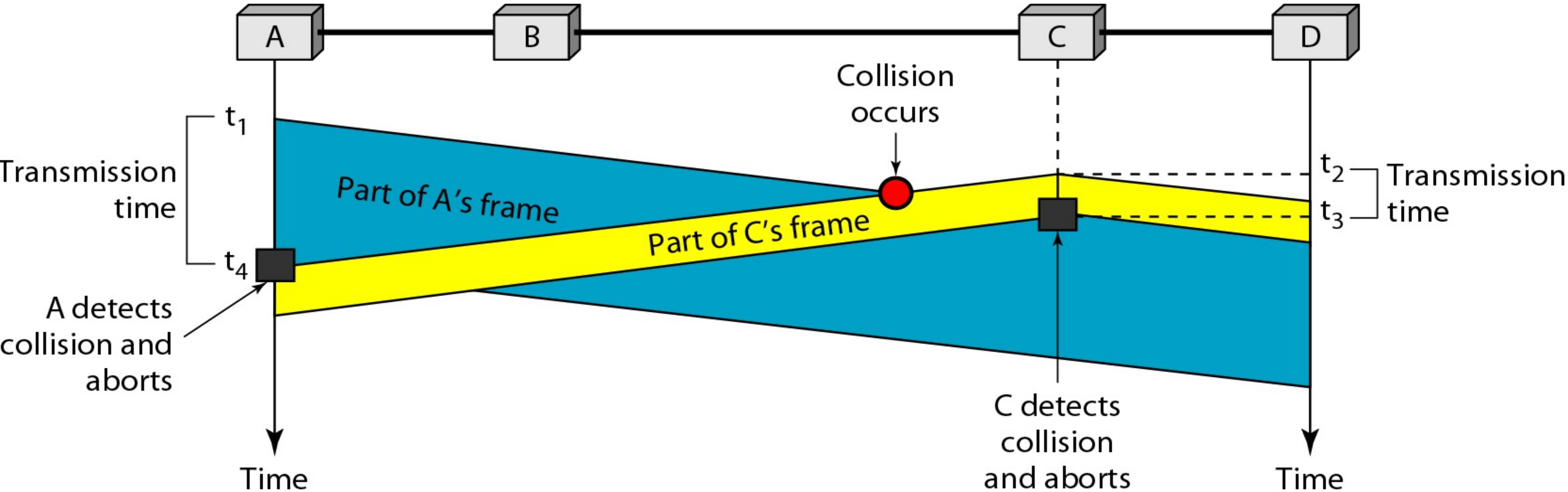


Figure 12.12 *Collision of the first bit in CSMA/CD*



- At time t1, station A starts sending the bits of its frame.
- At time t2, station C has not yet sensed the first bit sent by A. Station C starts sending the bits in its frame, which propagate both to the left and to the right.
- The collision occurs sometime after time t2. Station C detects a collision at time t3 when it receives the first bit of A's frame.
- Station C immediately aborts transmission. Station A detects collision at time t4 when it receives the first bit of C's frame; it also immediately aborts transmission.
- Looking at the figure, we see that A transmits for the duration t4 – t1; C transmits for the duration t3 – t2.

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 and ignoring the time needed to send a jamming signal, as we see later) is 25.6 μs, what is the minimum size of the frame?

Solution

The frame transmission time is $T_{fr} = 2 \times T_p = 51.2\ \mu s$. This means, in the worst case, a station needs to transmit for a period of 51.2 μs to detect the collision. The minimum size of the frame is 10 Mbps × 51.2 μs = 512 bits or 64 bytes. This is actually the minimum size of the frame for Standard Ethernet.

CSMA/Collision Avoidance

- Carrier sense multiple access with collision avoidance (CSMA/CA) was invented for wireless networks
- Collisions are avoided through the use of CSMA/CA's three strategies:
  - 1. Inter-frame Spacing (IFS)
  - 2. Contention Window – Binary Exponential Back off Algorithm
  - 3. Acknowledgement

Figure 12.17 *Flow diagram for CSMA/CA*

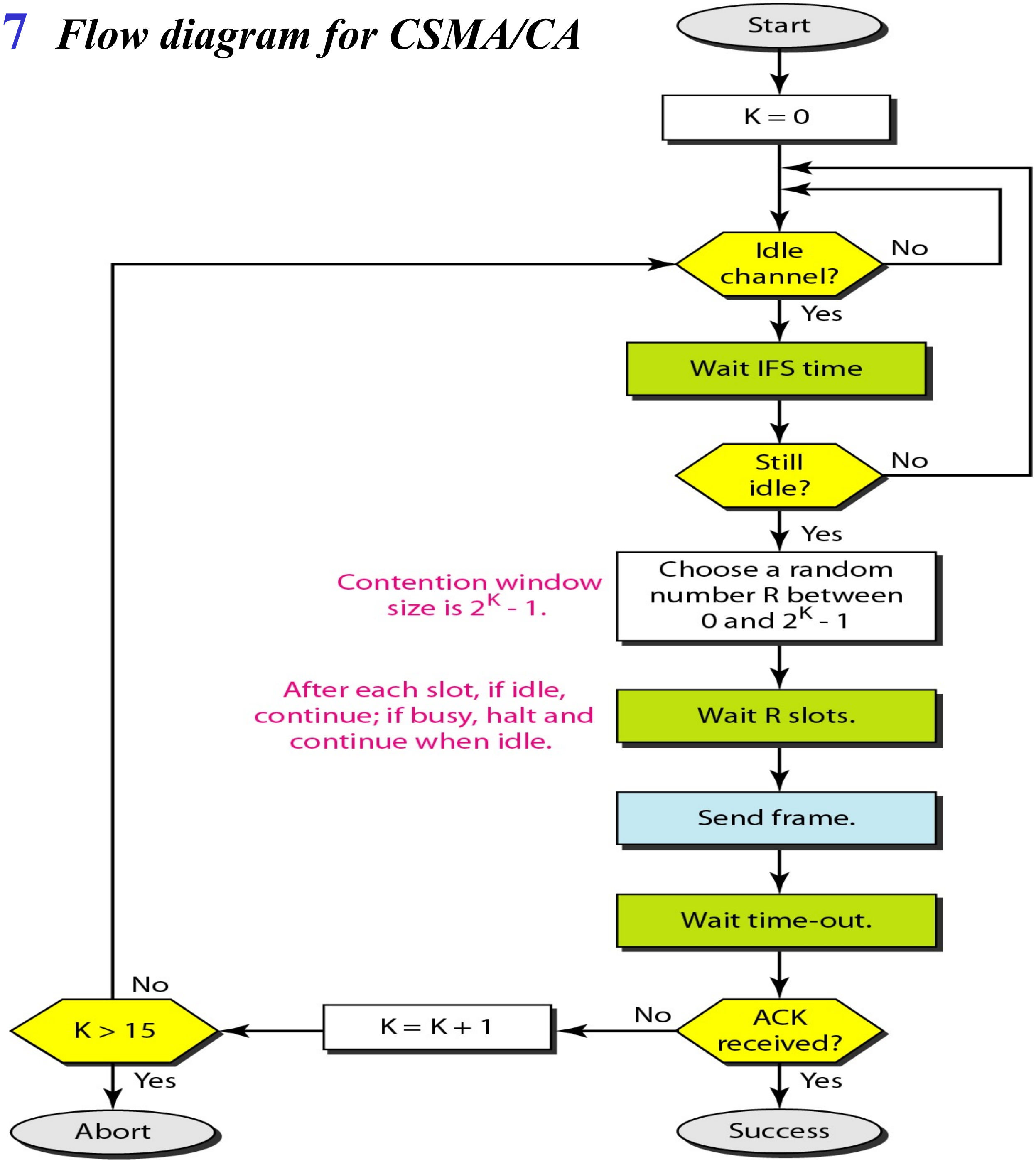
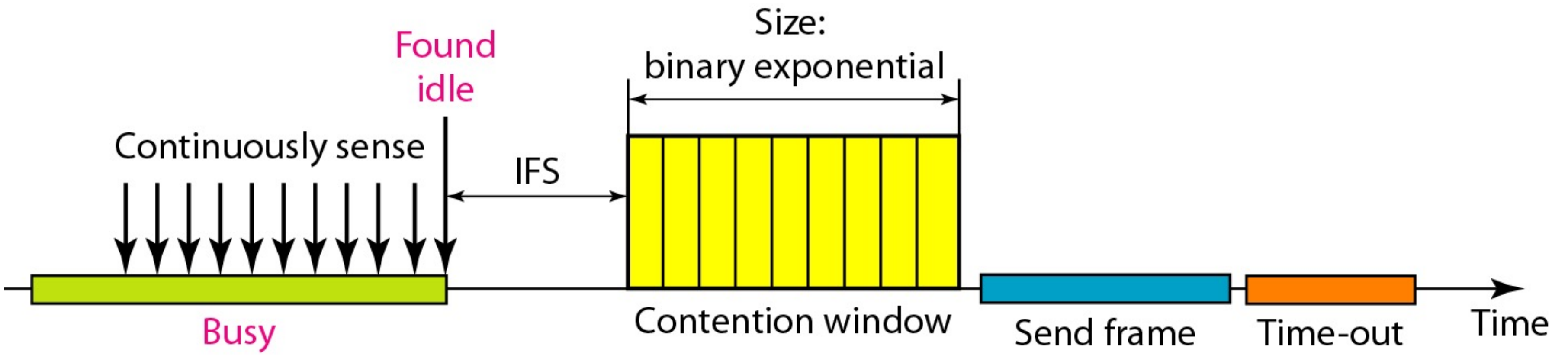


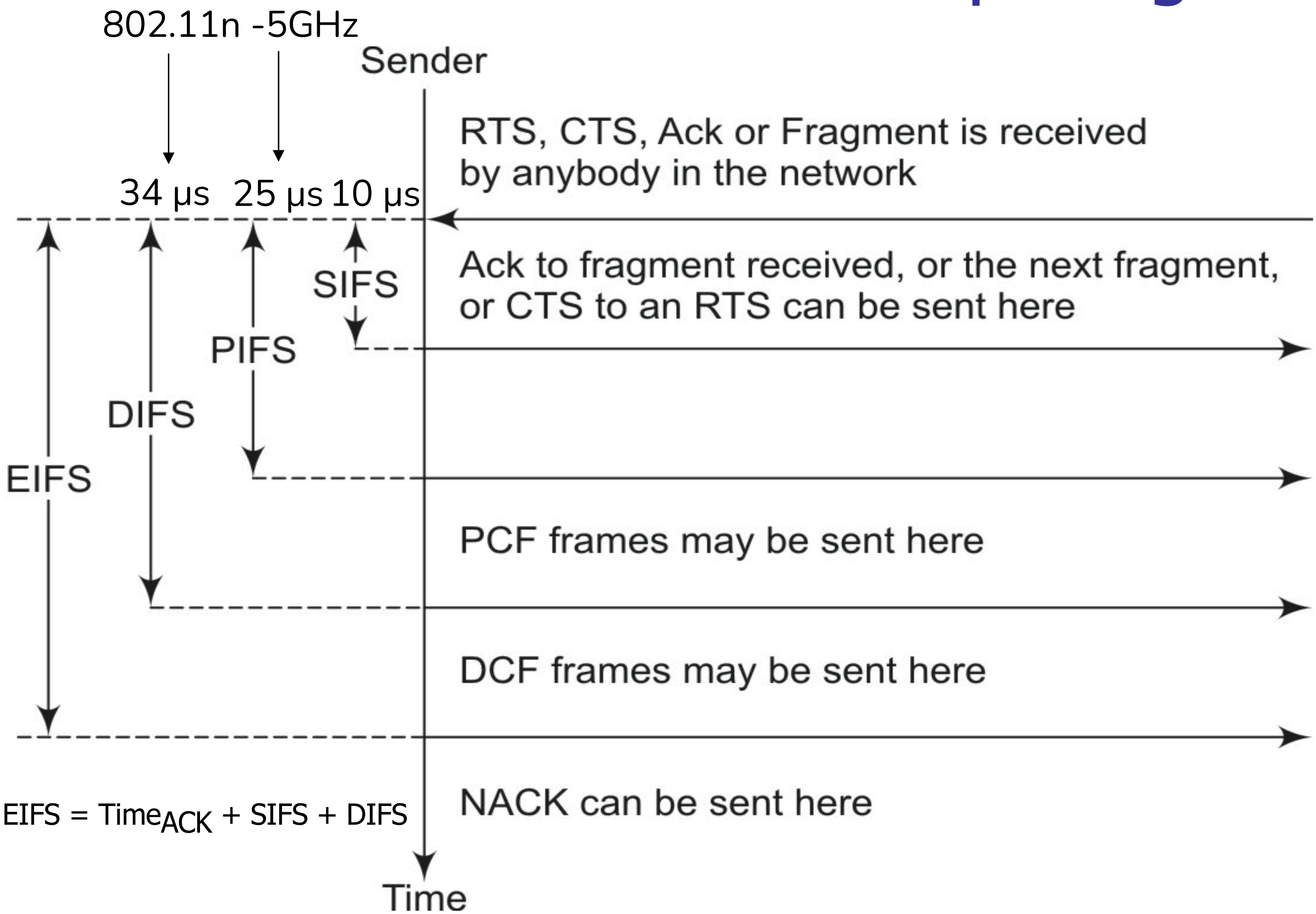
Figure 12.16 *Timing in CSMA/CA*



Note

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

Different Inter-frame spacing



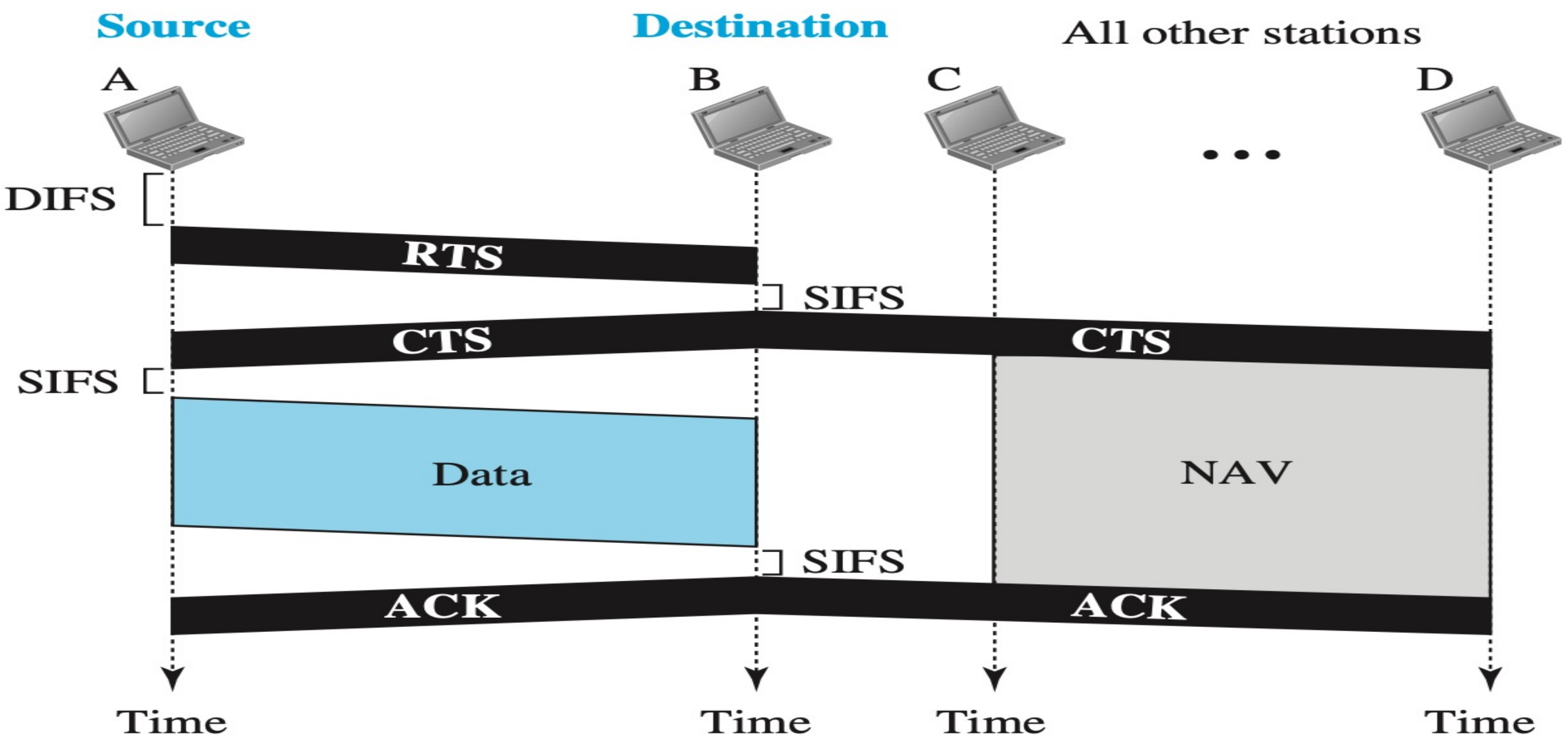
Binary Exponential Backoff

- Sender sends immediately with idle channel
- Continues to listen while transmitting
- In case of a collision, the sender waits for a random period (maximum of two time slots)
- In case they collide again, the interval is just doubled every time it experiences a collision
- When doubling is repeated to the slot size to 0–1023 it will not increase further

Binary Exponential Back off Algorithm

- Time is divided into discrete slots whose length is equal to the worst-case round-trip propagation time on the either (2τ).
  - minimum frame is 64 bytes (header + 46 bytes of data) = 512 bits
  - Channel capacity 10 Mbps, 512/10 M = 51.2μ
- After 1^{st} collision, each station waits for **0 or 1** time slot before trying again.
- After 2^{nd} collision, each station picks up **either 0,1,2 or 3** at random and waits for that much time slots.
- If 3^{rd} collision occurs, then next time number of slots to wait is chosen randomly from interval **0 to 2^3-1**.
- In general, after i^{th} collision, random number between **0 to 2^i-1** is chosen, that number of time slot is skipped.
- After 10^{th} collision, randomized interval is frozen at **max of 1023 slots**.
- After 16^{th} collision, controller **reports failure** back to computer sending and further recovery is upto higher layers.
- This algorithm is called Binary Exponential Back off Algorithm.
- **Advantage:** Ensures a low delay when only a few stations collide, but also assures that the collision is resolved in a reasonable interval when many stations collide.
- **Disadvantage:** Could introduce significant delay.

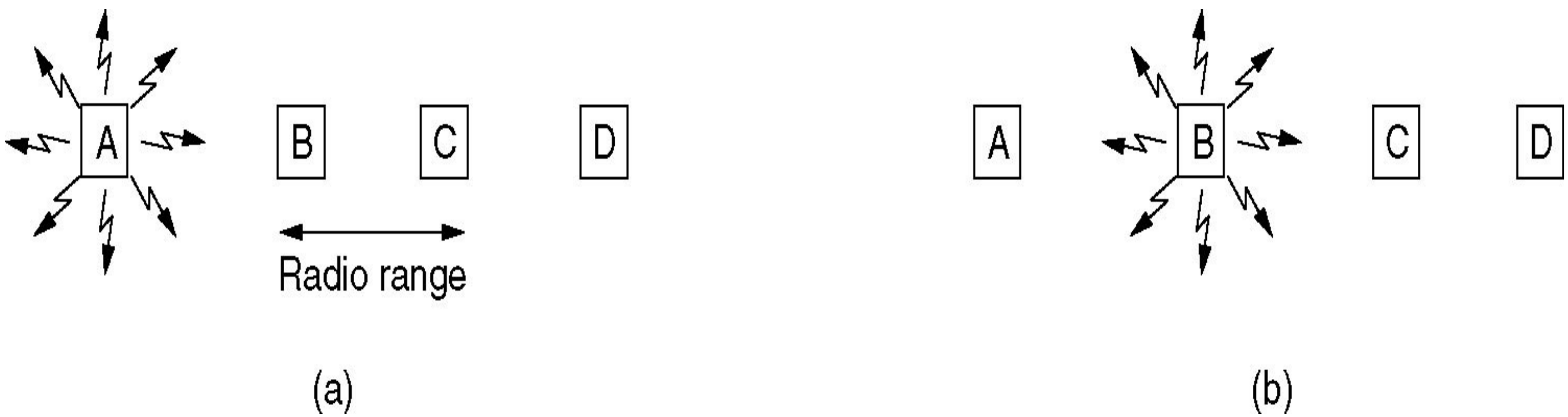
Frame Exchange Time Line in CSMA/CA



- When a station sends an RTS/CTS frame, it includes the duration of time that it needs to occupy the channel.
- The stations that are affected by this transmission create a timer called a network allocation vector (NAV) that shows how much time must pass before these stations are allowed to check the channel for idleness.

Wireless LAN Protocols

- *Hidden station problem:* A is transmitting to B. C cannot hear A. If C starts transmitting, it will interfere at B.
- *Exposed station problem:* B is transmitting to A. C concludes that it may not send to D but the interference exists only between B and C.



A wireless LAN.
(a) A transmitting. (b) B transmitting.

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. In this section, we discuss three channelization protocols.*

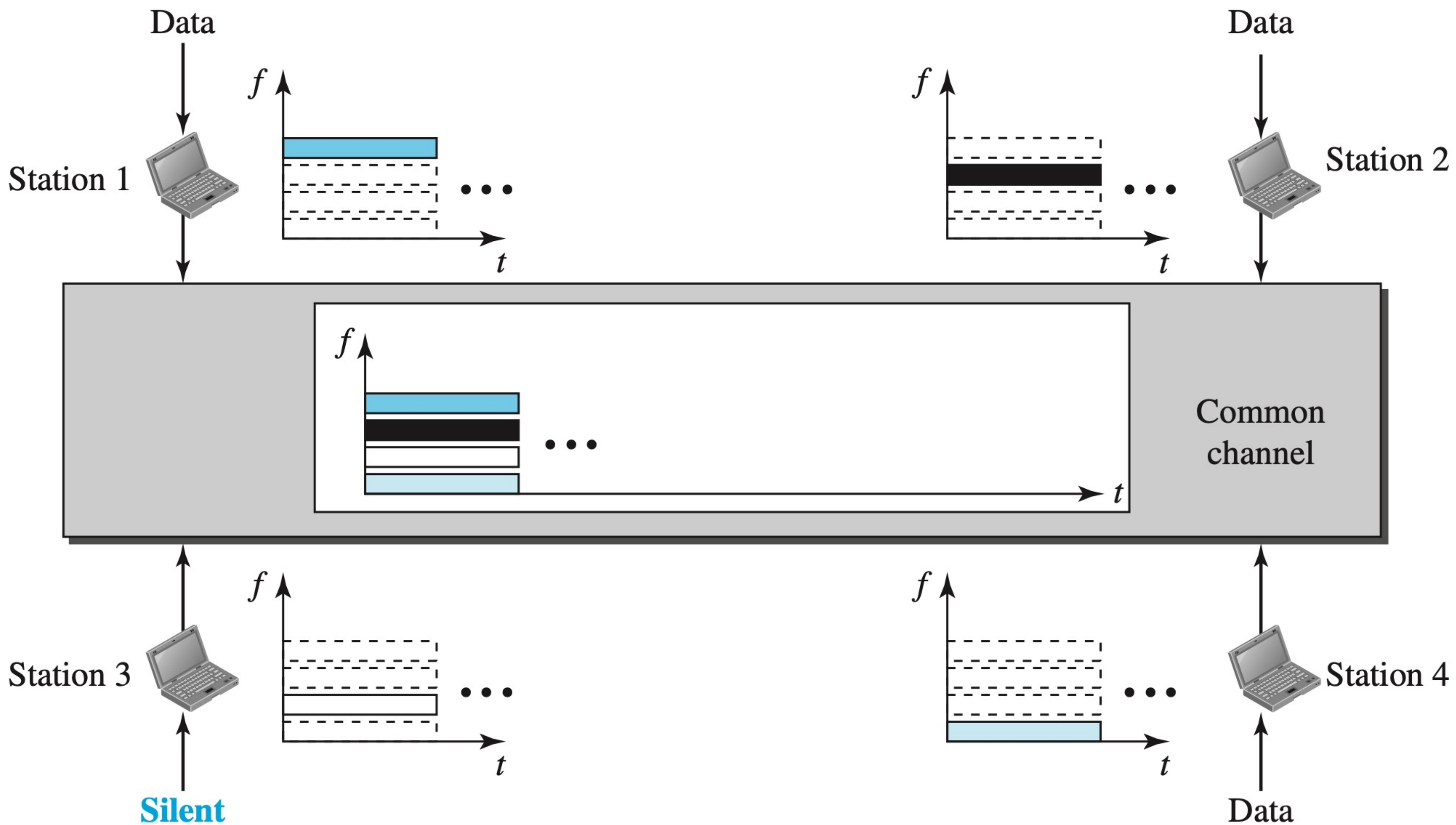
Topics discussed in this section:

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

Frequency-division multiple access (FDMA)

- In frequency-division multiple access (FDMA), the available bandwidth is divided into frequency bands.
- Each station is allocated a band to send its data. In other words, each band is reserved for a specific station, and it belongs to the station all the time.
- Each station also uses a bandpass filter to confine the transmitter frequencies.
- To prevent station interferences, the allocated bands are separated from one another by small guard bands

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



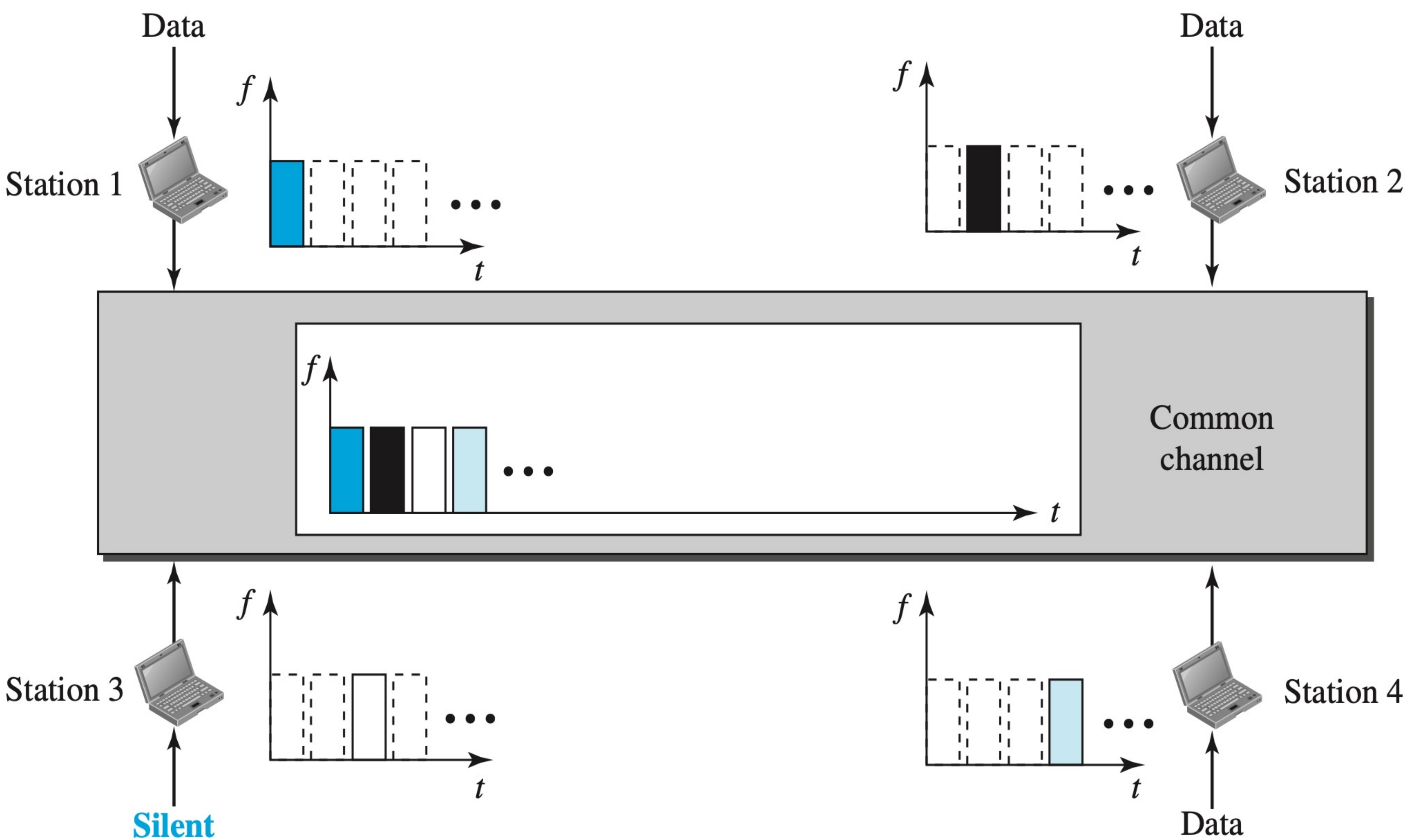
Note

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

Time-division multiple access (TDMA)

- In time-division multiple access (TDMA), the stations share the bandwidth of the channel in time.
- Each station is allocated a time slot during which it can send data.
- Each station transmits its data in its assigned time slot.
- The main problem with TDMA lies in achieving synchronization between the different stations.
- Each station needs to know the beginning of its slot and the location of its slot. This may be difficult because of propagation delays introduced in the system if the stations are spread over a large area.
- To compensate for the delays, we can insert guard times. Synchronization is normally accomplished by having some synchronization bits (normally referred to as preamble bits) at the beginning of each slot.

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



Note

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

Code-division multiple access (CDMA)

- In Code-division multiple access (CDMA) simply means communication with different codes.
- Consider, we have four stations, 1, 2, 3, and 4, connected to the same channel. The data from station 1 are d_1, from station 2 are d_2, and so on.
- The code assigned to the first station is c_1, to the second is c_2, and so on.
- Station 1 multiplies its data by its code to get $d_1 \cdot c_1$. Station 2 multiplies its data by its code to get $d_2 \cdot c_2$, and so on.
- The data that go on the channel are the sum of all these terms.
- Now, Station 2 wants to hear what station 1 is saying. It multiplies the data on the channel by c_1, the code of station 1.
- Because ($c1 \cdot c1$) is 4, but ($c2 \cdot c1$), ($c3 \cdot c1$), and ($c4 \cdot c1$) are all 0s, station 2 divides the result by 4 to get the data from station 1.

Note

CDMA differs from FDMA in that only one channel occupies the entire bandwidth of the link.

Note

CDMA differs from TDMA in that all stations can send data simultaneously; there is no timesharing.

Figure 12.23 *Simple idea of communication with code*

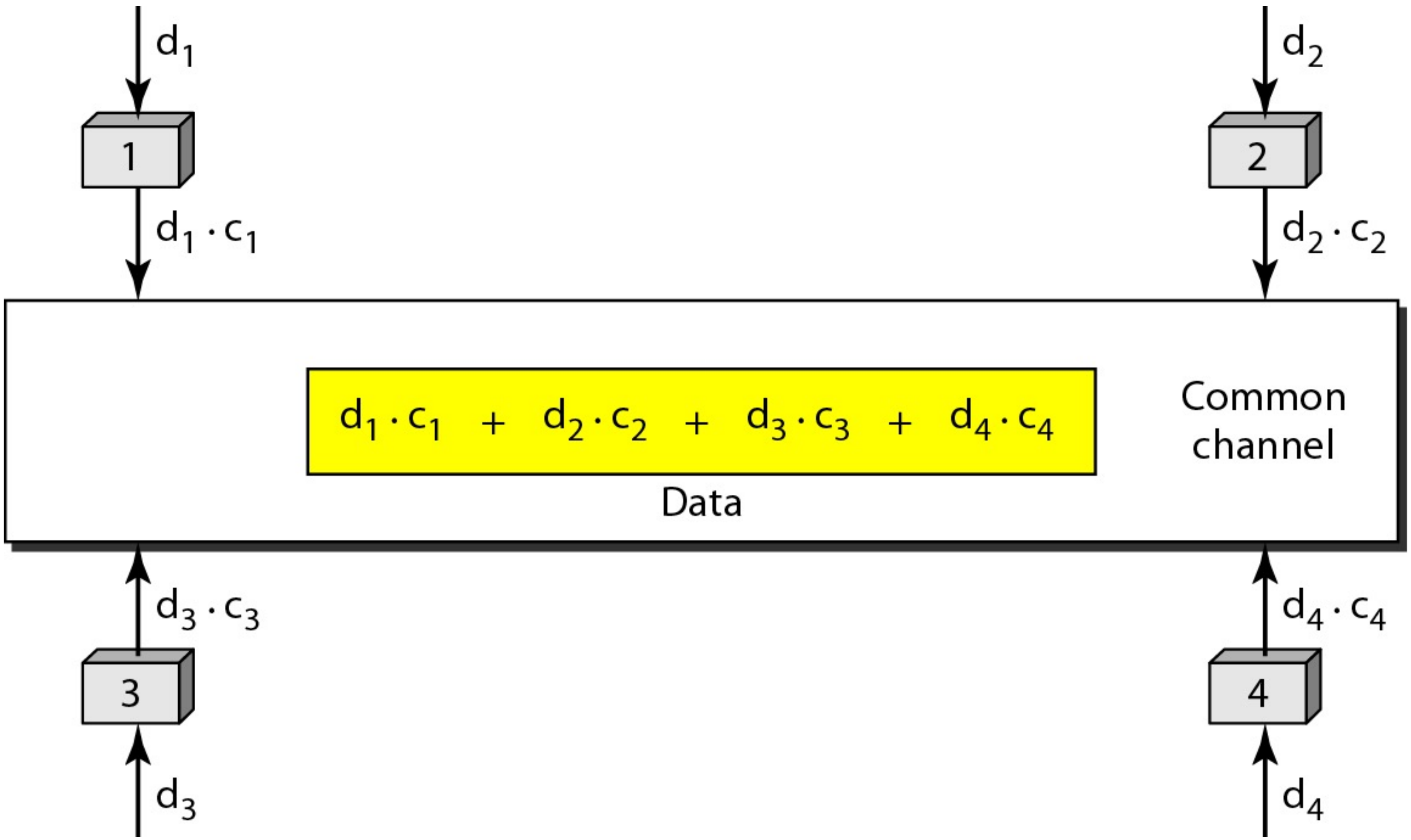


Figure 12.24 *Chip sequences*

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

- Each station is assigned a code, which is a sequence of numbers called chips.
- They are called **orthogonal sequences** and have the following properties:
  1. Each sequence is made of N elements, where N is the number of stations.
  2. If we multiply a sequence by a number, every element in the sequence is multiplied by that element. This is called multiplication of a sequence by a scalar. For example,

     2. [+1 +1-1-1]=[+2 +2 -2 -2]

  3. If we multiply two equal sequences, element by element, and add the results, we get N, where N is the number of elements in the each sequence. This is called the inner product of two equal sequences. For example,

     [+1 +1-1 -1]· [+1 +1 -1 -1] = 1 + 1 + 1 + 1 = 4

  4. If we multiply two different sequences, element by element, and add the results, we get 0. This is called inner product of two different sequences. For example,

     [+1+1-1-1]•[+1+1+1+1]=1+1- 1- 1=0

  5. Adding two sequences means adding the corresponding elements. The result is another sequence. For example,

     [+1+1-1-1]+[+1+1+1+1]=[+2 +2 0 0]

Figure 12.25 *Data representation in CDMA*

Data bit 0 → -1	Data bit 1 → +1	Silence → 0

- We assume that stations 1 and 2 are sending a 0 bit and station 4 is sending a 1 bit whereas station 3 is silent.
- The data at the sender site are translated to -1, -1, 0, and +1.
- Each station multiplies the corresponding number by its chip (its orthogonal sequence), which is unique for each station.
- The result is a new sequence which is sent to the channel. For simplicity, we assume that all stations send the resulting sequences at the same time.
- The sequence on the channel is the sum of all four sequences which is

  [-1-1-3 +1]
- Now station 3 wants to listen from station 2. Station 3 multiplies the total data on the channel by the code for station 2, which is [+1 -1 +1-1], to get

  [-1 -1 -3 +1]· [+1 -1 +1 -1] =-4/4 =-1 ...... bit 0

Figure 12.26 *Sharing channel in CDMA*

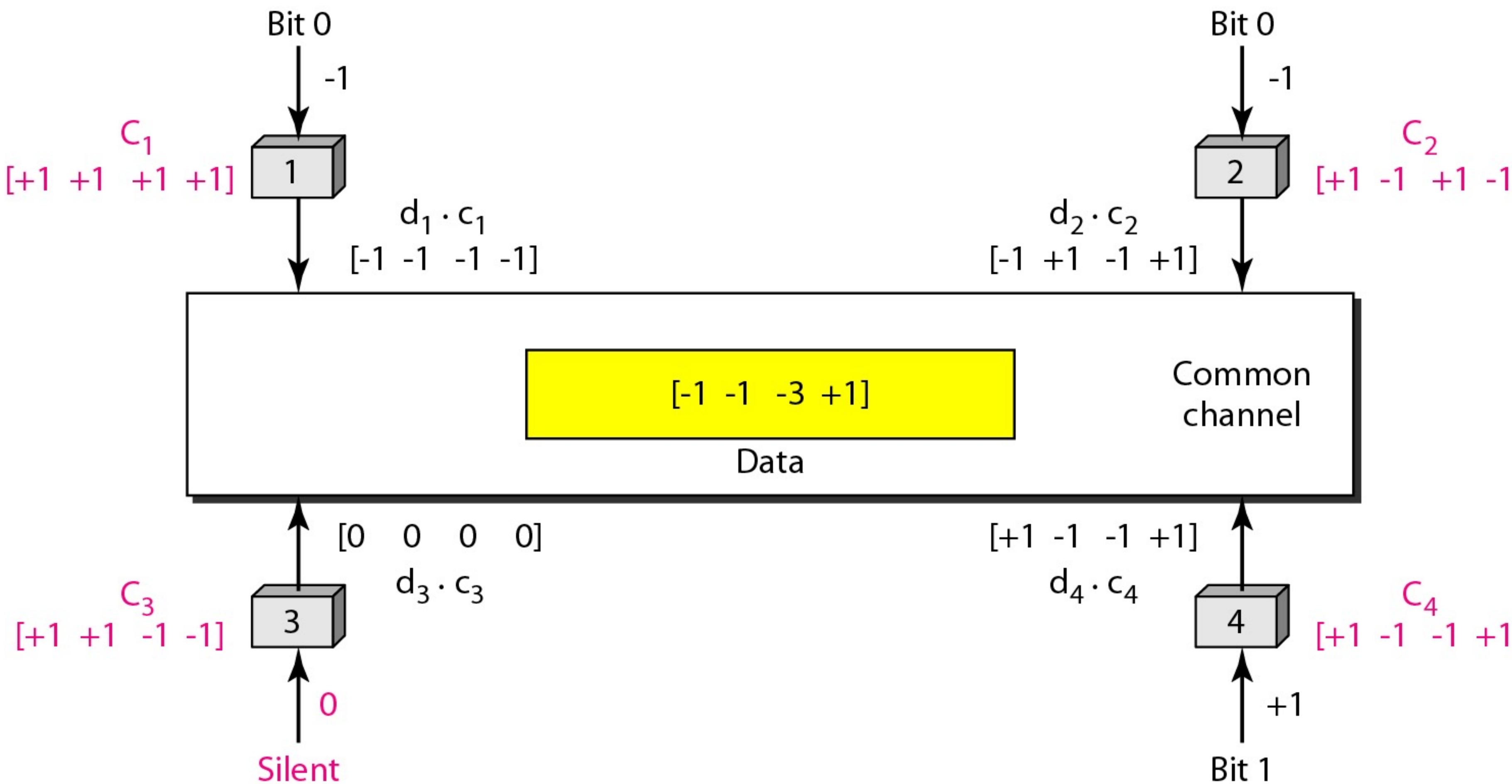


Figure 12.27 *Digital signal created by four stations in CDMA*

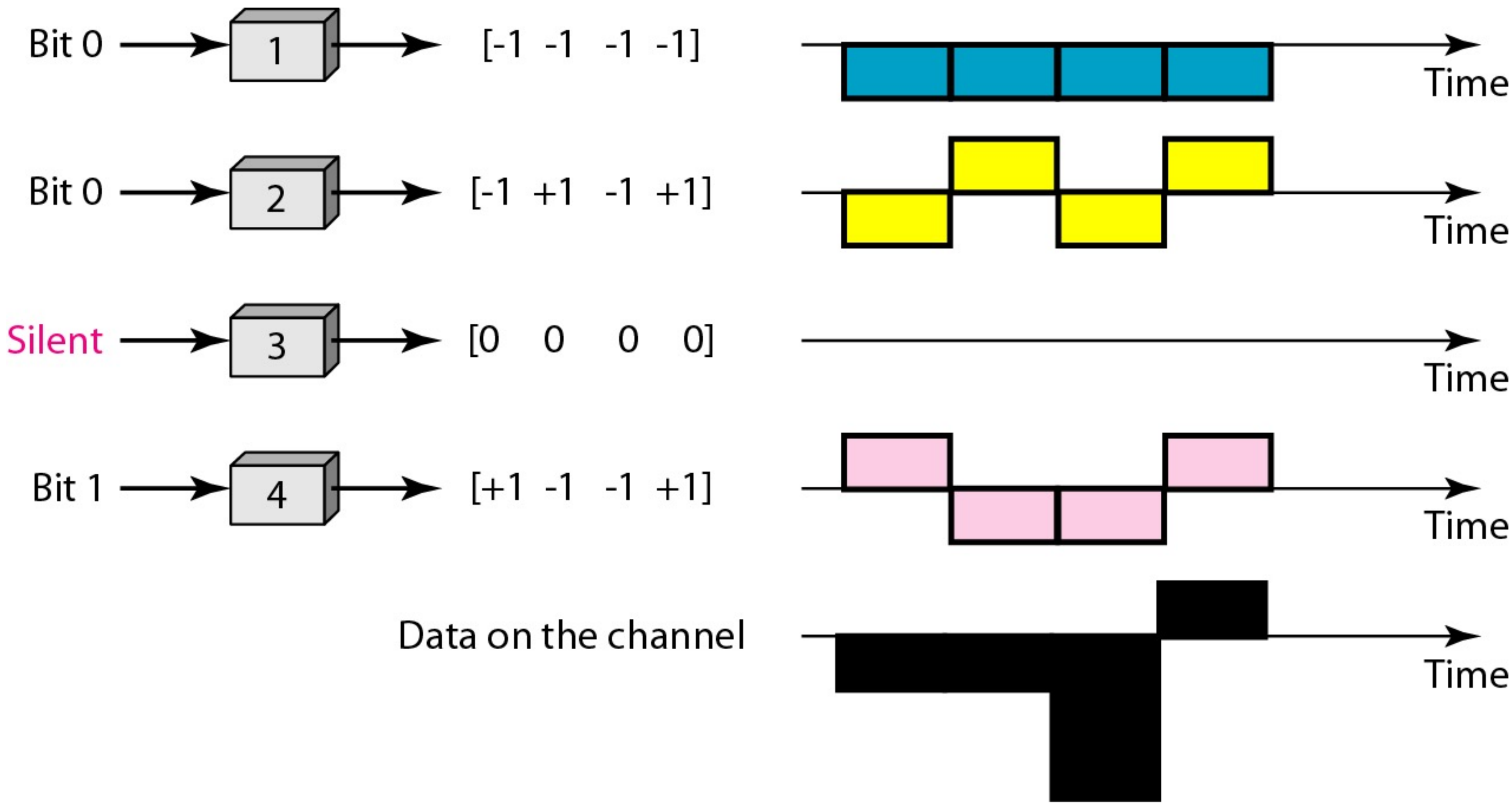


Figure 12.28 *Decoding of the composite signal for one in CDMA*

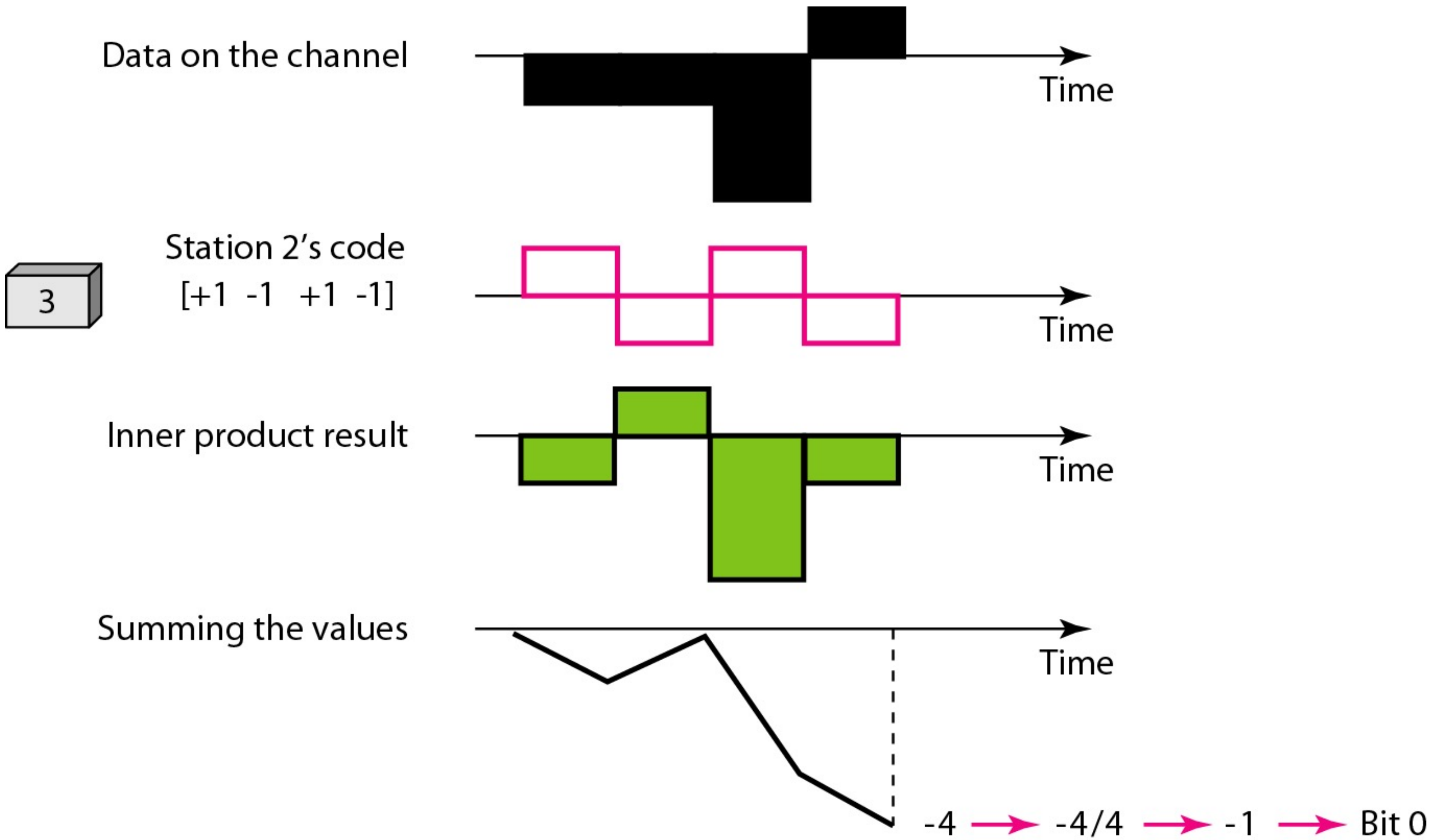


Figure 12.29 *General rule and examples of creating Walsh tables*

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

a. Two basic rules

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

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

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

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

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

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

Example 12.8 (continued)

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

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