Unit VI

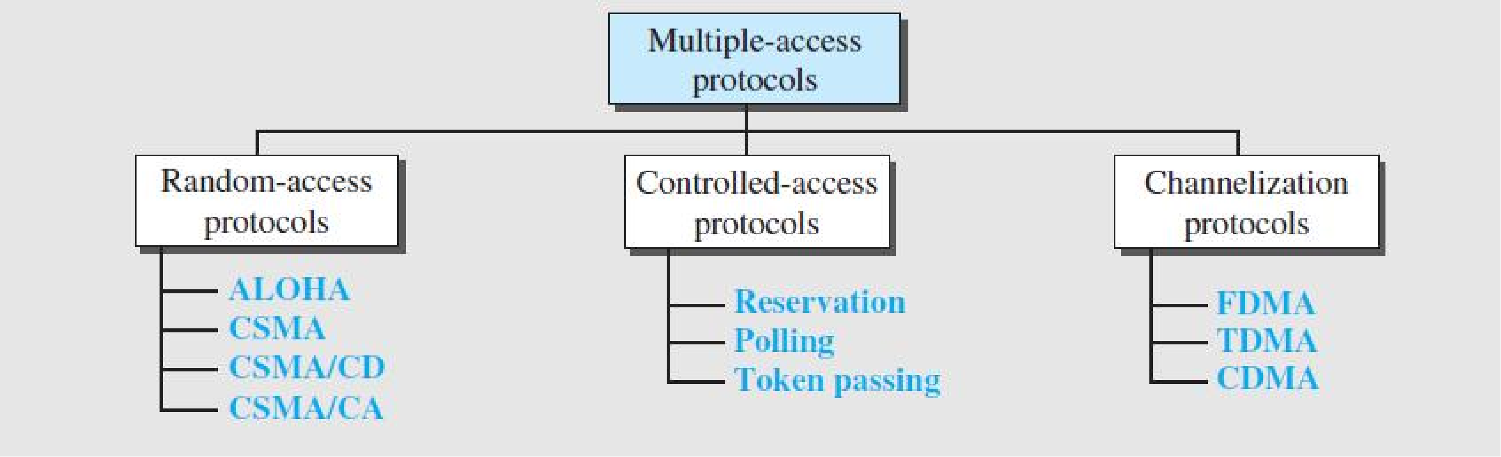
(Part I)

Media Access Control (MAC)

Vinit R Tribhuvan
vrt22.comp@coep.ac.in

Department of Computer Engineering and Information Technology
College of Engineering Pune (COEP)
Forerunners in Technical Education

Multiple Access Protocols



Random Access

- Two features –
- **Random Access** - no scheduled time for a station to transmit.
- **Contention -** Stations compete with one another to access the medium.

- A station that has data to send uses a procedure defined by the protocol to make a **decision** on whether or not to send.

- Depends on the state of the medium (idle or busy).

- Each station has the right to the medium without being controlled by any other station.
- if more than one station tries to send, there is an access conflict—***collision***

- To **avoid access conflict** or resolve it, each station follows a procedure that answers the following questions:
  - When can the station access the medium?
  - What can the station do if the medium is busy?
  - How can the station determine the success or failure of the transmission?
  - What can the station do if there is an access conflict?

ALOHA

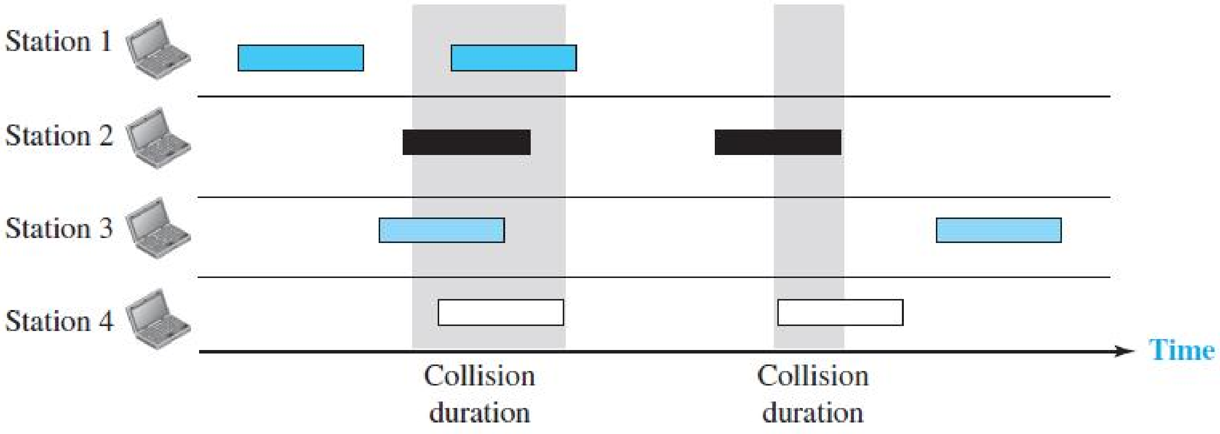
- Developed at the University of Hawaii in early 1970.
- It was designed for a radio (wireless) LAN.
- The medium is shared between the stations.
- When a station sends data, another station may attempt to do so at the same time.
- The data from the two stations **collide** and become garbled.

Pure ALOHA

- Original ALOHA
- Each station sends a frame whenever it has a frame to send (multiple access).
- Since there is only one channel to share, there is the possibility of collision between frames from different stations.

Frames in ALOHA network

Figure 12.2 *Frames in a pure ALOHA network*



- Even if one bit of a **frame coexists on the channel** with one bit from another frame, there is a collision and both will be destroyed.
- we need to **resend the frames** that have been destroyed during transmission.

- When a station sends a frame, it expects the receiver to send an acknowledgment.

- If the **acknowledgment does not arrive** after a time-out period, the station assumes that the frame (or the acknowledgment) has been destroyed and **resends the frame**.

- After collision, the stations resend their frames after ***time-out*** period.
- There is a possibility that the frames may collide again.
- In Pure ALOHA, each station waits a ***random amount of time*** before resending its frame.
- The randomness will help avoid more collisions. And this time is called as the ***backoff time*** $\boldsymbol{T_B}$**.**

backoff time T_B

- Pure ALOHA has a second method to prevent congesting the channel with retransmitted frames.
- After a maximum number of retransmission attempts ***Kmax,*** a station must give up and try later.

- The time-out period is equal to the maximum possible round-trip propagation delay, **which is twice the amount of time required to send a frame between the two most widely separated stations (2 x *Tp*)**

- The backoff time *TB* is a random value that normally depends on *K* (the number of attempted unsuccessful transmissions).

binary exponential backoff

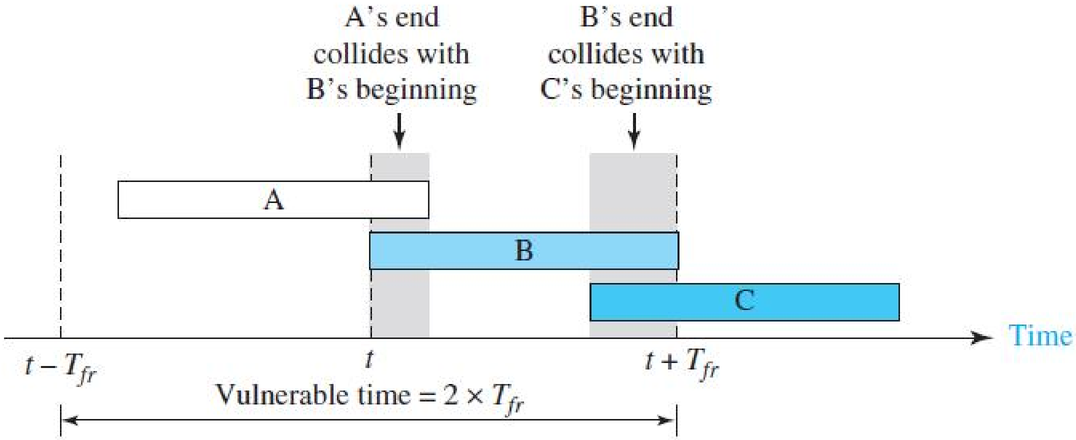
- for each retransmission, a multiplier $R = 0$ to $2^K - 1$ is randomly chosen and multiplied by Tp (maximum propagation time) or Tfr (the average time required to send out a frame) to find TB.

- Note that in this procedure, the range of the random numbers increases after each collision. The value of $Kmax$ is usually chosen as 15.

Vulnerable time

- It is the length of time in which there is a possibility of collision.

Figure 12.4 *Vulnerable time for pure ALOHA protocol*



- Station B starts to send a frame at time t.
- Now imagine station A has started to send its frame after $t - Tfr$.
- This leads to a collision between the frames from station B and station A
- suppose that station C starts to send a frame before time $t + Tfr$.
- there is also a collision between frames from station B and station C.

Vulnerable time

we see that the vulnerable time during which a collision may occur in pure ALOHA is 2 times the frame transmission time

$$\text{Pure ALOHA vulnerable time} = 2 \times T_{fr}$$

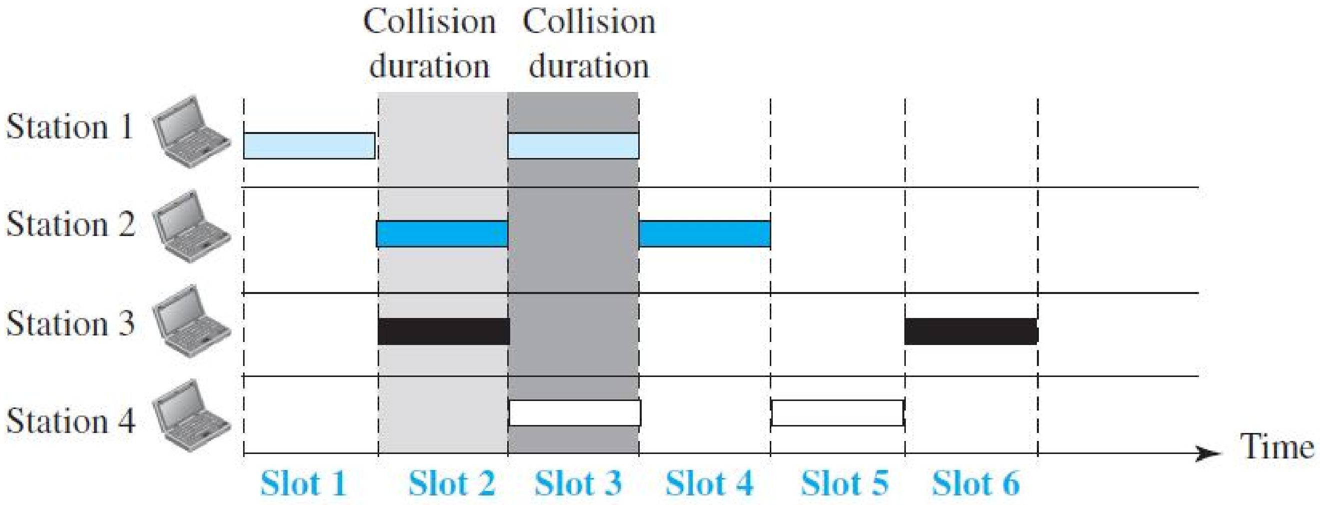
Shared ALOHA

- there is no rule that defines when the station can send.
- A station may send soon after another station has started or just before another station has finished.

- Slotted ALOHA improves the efficiency of pure ALOHA.

- In ***slotted ALOHA*** we divide the time into slots of *Tfr* seconds and force the station to **send only at the beginning of the time slot.**

Figure 12.5 *Frames in a slotted ALOHA network*



- if a station misses this moment, it must wait until the beginning of the next time slot.

- Of course, there is still the possibility of collision if two stations try to send at the beginning of the **same time slot**.

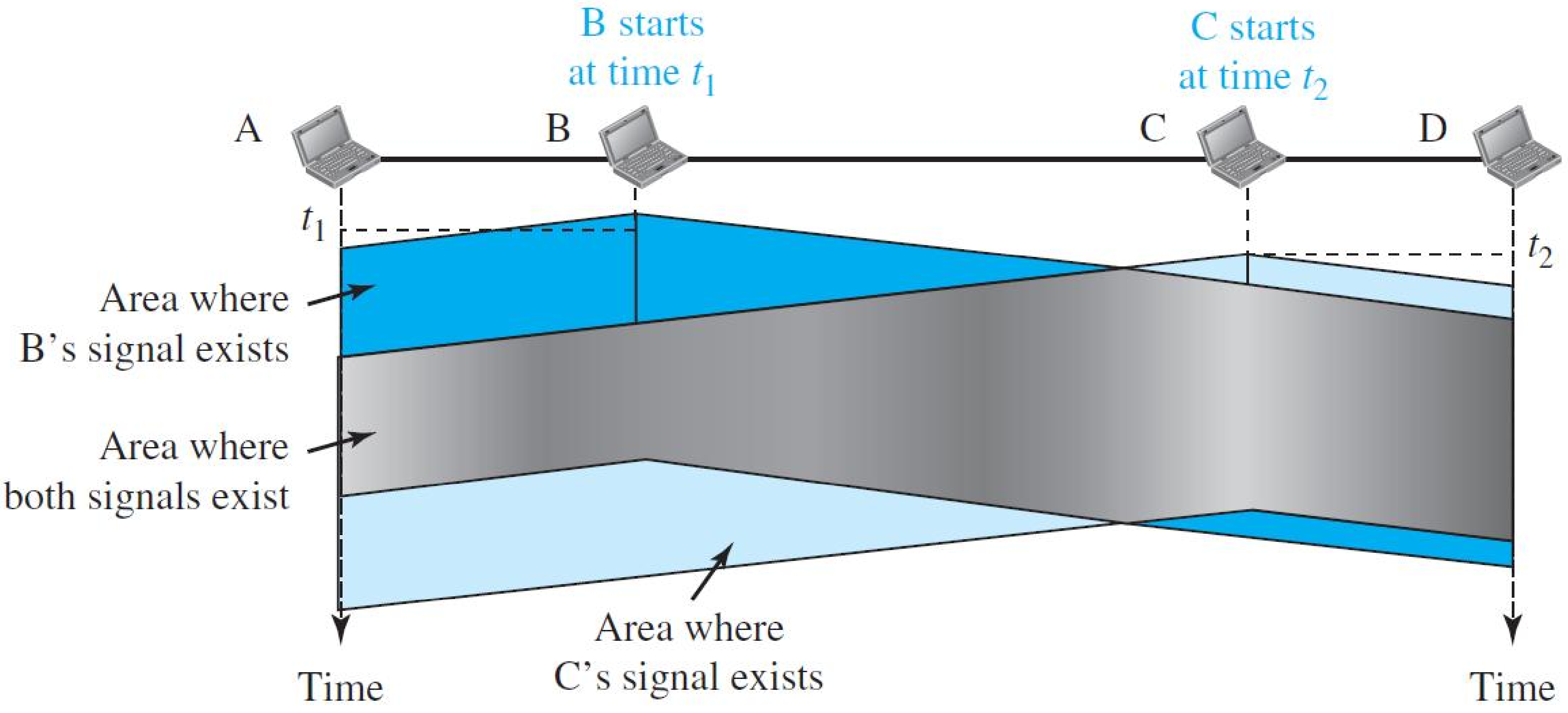
- However, the vulnerable time is now reduced to one-half, equal to Tfr.

$$\text{Slotted ALOHA vulnerable time} = T_{fr}$$

Carrier sense multiple access (CSMA)

- Minimizes the chance of collision.
- each station first listen (sense) to the medium (or check the state of the medium) before sending.
- sense before transmit – listen before talk.
- CSMA can reduce the possibility of collision, but it cannot eliminate it.

Space/time model of a collision in CSMA


B starts at time t_1
C starts at time t_2
A
B
C
D
t_1
t_2
Area where B's signal exists
Area where both signals exist
Area where C's signal exists
Time
Time

- The possibility of collision still exists because of propagation delay.
- when a station sends a frame, it still takes time (although very short) for the first bit to reach every station and for every station to sense it.
- a station may sense the medium and find it idle, only because the first bit sent by another station has not yet been received.
- 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.

Vulnerable Time

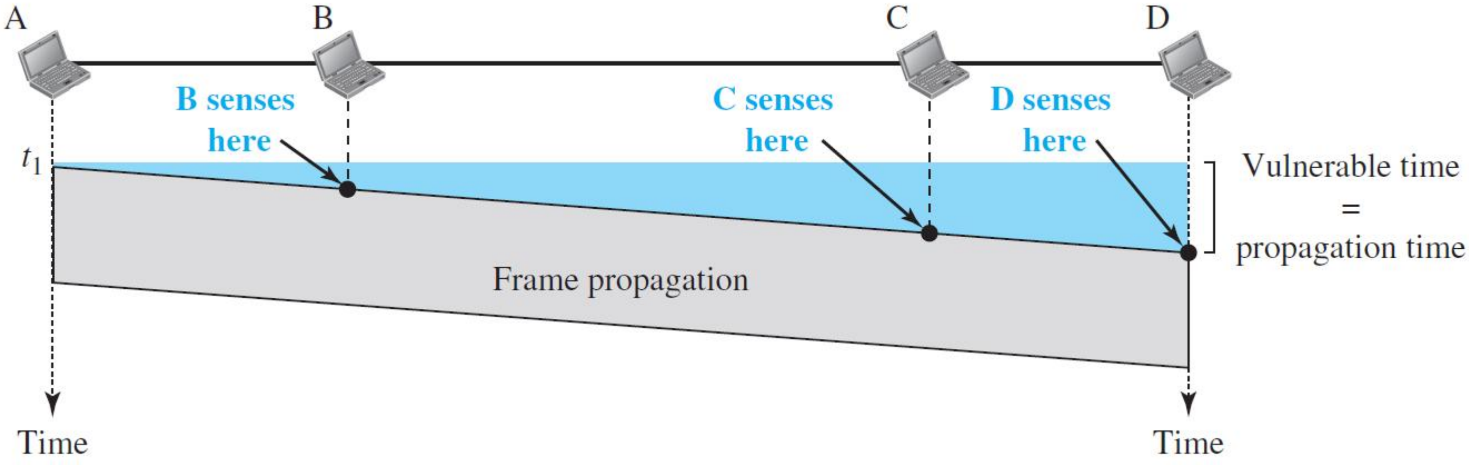
- The vulnerable time for CSMA is the *propagation time Tp*

- 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.



Questions

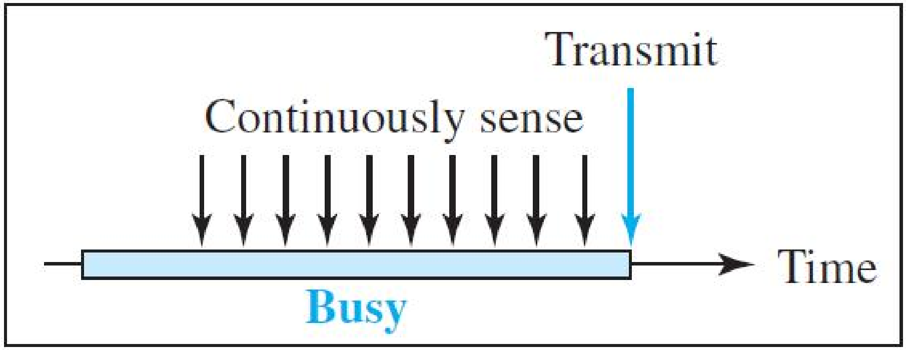
- What should a station do if the channel is busy?
- What should a station do if the channel is idle?

Persistent methods

- Three methods
  1. 1-persistent method,
  2. nonpersistent method
  3. *p*-persistent method

1-persistent method

- After the station finds the line idle, it sends its frame immediately (with probability 1).
- This method has the highest chance of collision because two or more stations may find the line idle and send their frames immediately.
- Ethernet uses this method.



a. 1-Persistent

Nonpersistent method

- a station that has a frame to send senses the line. If the line is idle, it sends immediately.

- If the line is not idle, it waits a random amount of time and then senses the line again.

- The nonpersistent approach ***reduces the chance of collision*** because it is unlikely that two or more stations will wait the same amount of time and retry to send simultaneously.

- reduces the efficiency of the network because the medium remains idle when there may be stations with frames to send.

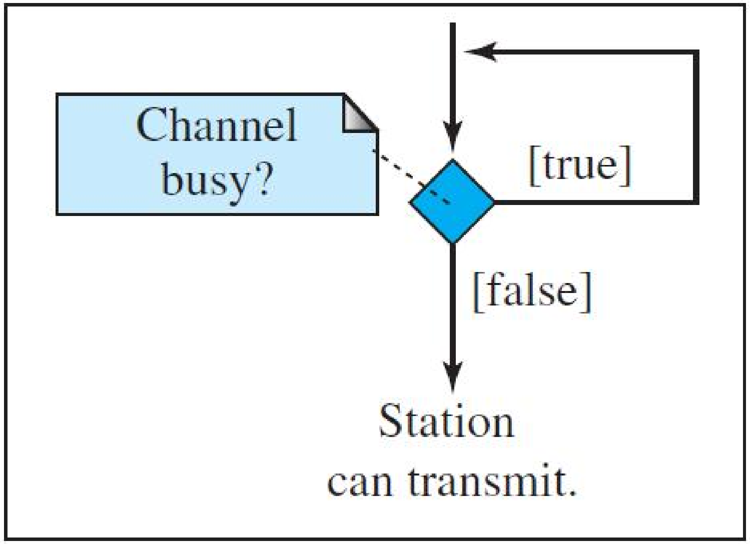
p-Persistent

- used if the channel has time slots with a slot duration equal to or greater than the maximum propagation time.
- Combines the advantages of the other two strategies.
- It reduces the chance of collision and improves efficiency

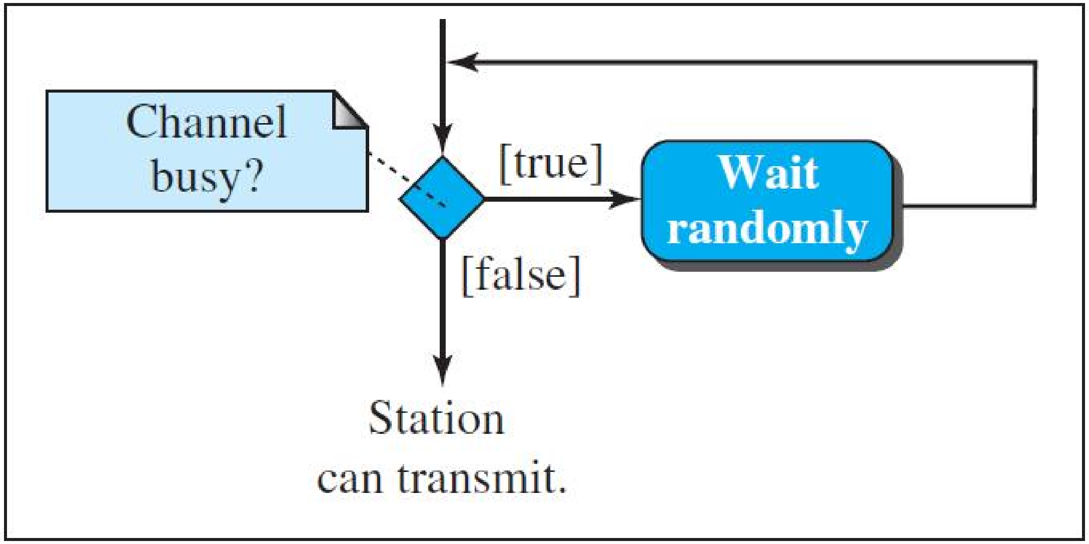
p-Persistent

- after the station finds the line idle it follows these steps:

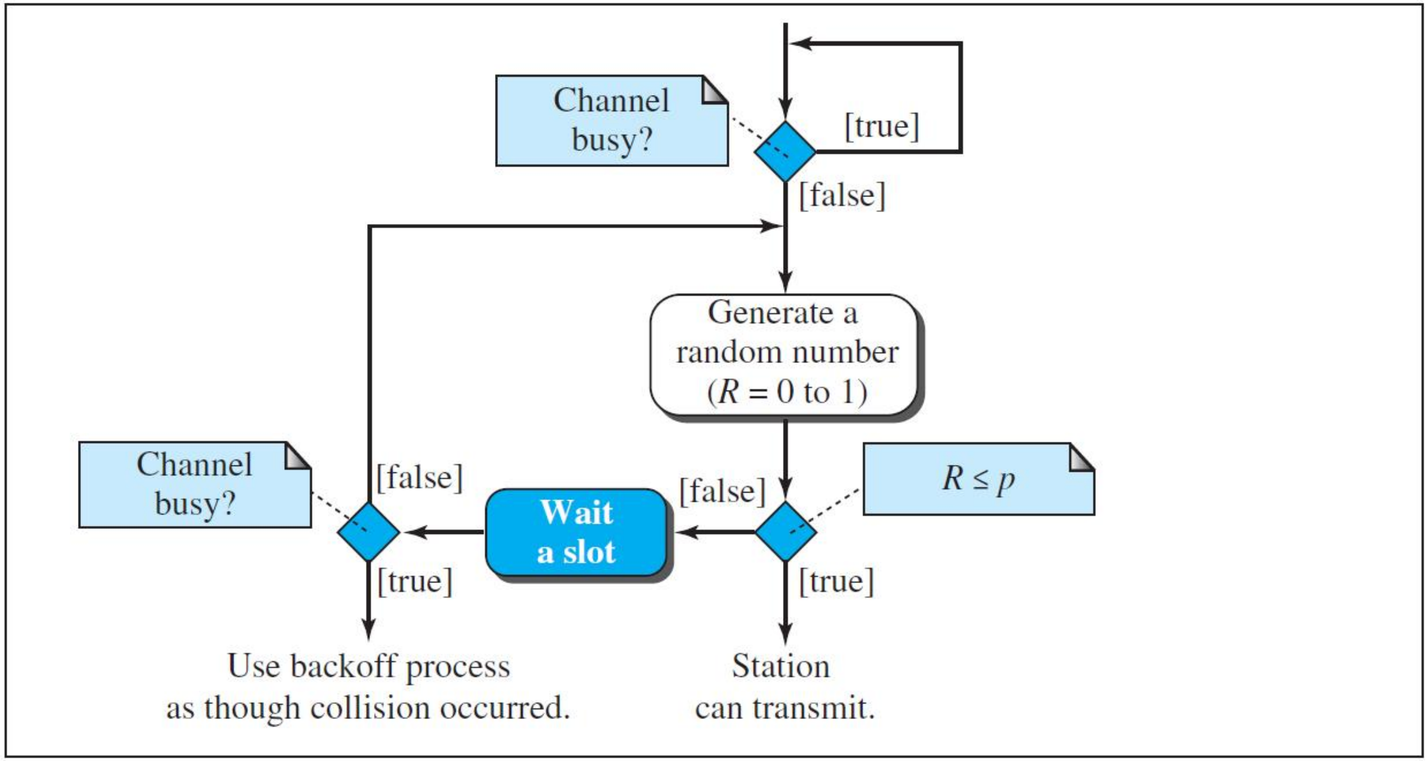
1. With probability p, the station sends its frame.
2. With probability $q = 1 - p$, the station waits for the beginning of the next time slot and checks the line again.
   a. If the line is idle, it goes to step 1.
   b. If the line is busy, it acts as though a collision has occurred and uses the backoff procedure.



a. 1-Persistent



b. Nonpersistent



c. *p*-Persistent

Carrier sense multiple access with collision avoidance (CSMA/CA)

- For wireless networks.
- Collisions are avoided through the use of CSMA/CA’s three strategies:
  - the interframe space
  - the contention window
  - acknowledgments

Interframe Space (IFS)

- collisions are avoided by deferring transmission even if the channel is found idle
- When an idle channel is found, the station does not send immediately.
- It waits for a period of time called the *interframe space* or *IFS.*

- a distant station may have already started transmitting

- The IFS time allows the front of the transmitted signal by the distant station to
- reach this station.

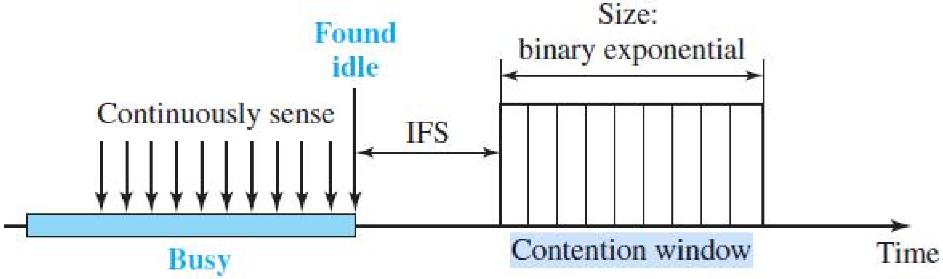
- After waiting an IFS time, if the channel is still idle, the station can send, but it still needs to wait a time equal to the contention window

Contention Window

- The **contention window** is an amount of time divided into slots.

- The station that is ready to send chooses a random number of slots as its wait time.
- The number of slots in the window changes according to the binary exponential backoff strategy.
- a random outcome defines the number of slots taken by the waiting station.
- the station needs to sense the channel after each time slot.
- if the station finds the channel busy, it does not restart the process; it just stops the timer and restarts it when the channel is sensed as idle
- Priority to the station with the longest waiting time.

Figure 12.16 *Contention window*



Acknowledgment

- There still may be a collision resulting in destroyed data.
- However, The positive acknowledgment and the time-out timer can help guarantee that the receiver has received the frame

the exchange of data and control frames in time

1. Before sending a frame, the source station senses the medium by checking the energy level at the carrier frequency.

a. The channel uses a persistence strategy with backoff until the channel is idle.

b. After the station is found to be idle, the station waits for a period of time called the *DCF interframe space (DIFS);* then the station sends a control frame called the *request to send (RTS).*

2. After receiving the RTS and waiting a period of time called the ***short interframe***

space (SIFS), the destination station sends a control frame, called the *clear to send (CTS),* to the source station. This control frame indicates that the destination station is ready to receive data

3. The source station sends data after waiting an amount of time equal to SIFS.

4. The destination station, after waiting an amount of time equal to SIFS, sends an acknowledgment to show that the frame has been received. Acknowledgment is needed in this protocol because the station does not have any means to check for the successful arrival of its data at the destination. On the other hand, the lack of collision in CSMA/CD is a kind of indication to the source that data have arrived.

how is the *collision avoidance* aspect of this protocol accomplished?

network allocation vector (NAV)

- When a station sends an RTS 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.
- Each time a station accesses the system and sends an RTS frame, other stations start their NAV.
- In other words, each station, before sensing the physical medium to see if it is idle, first checks its NAV to see if it has expired.

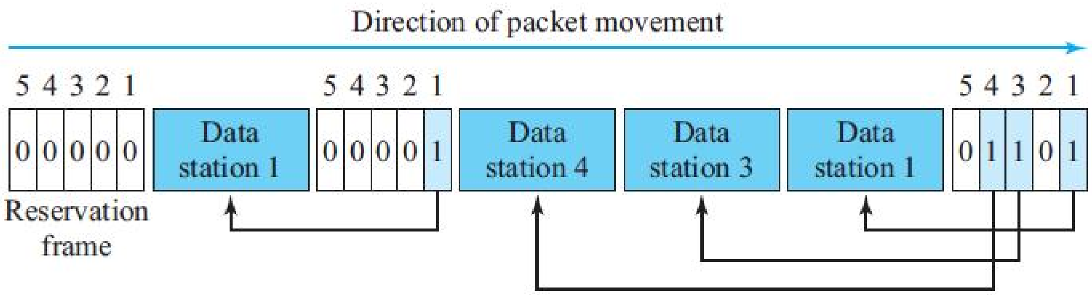
CONTROLLED ACCESS

- 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

Reservation

- In the ***reservation*** method, a station needs to make a reservation before sending data.
- Time is divided into intervals. In each interval, a reservation frame precedes the data frames sent in that interval.
- If there are *N* stations in the system, there are exactly *N* reservation minislots in the reservation frame.
- When a station needs to send a data frame, it makes a reservation in its own minislot.
- The stations that have made reservations can send their data frames after the reservation frame

Figure 12.18 *Reservation access method*



Polling

- one device is designated as a *primary station* and the other devices are *secondary station.*
- All data exchanges must be made through the primary device even when the ultimate destination is a secondary device.
- The primary device controls the link; the secondary devices follow its instructions.
- Primary device initiates the sessions.
- This method uses **poll** and **select** functions to prevent collisions.

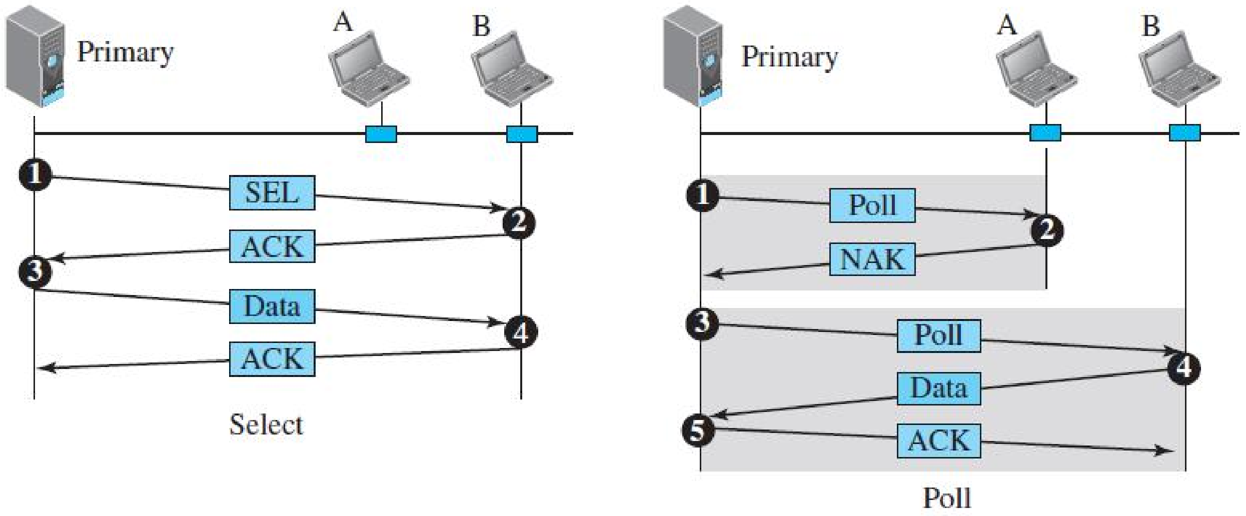
Select

- The *select* function is used whenever the primary device has something to send.
- If the primary is neither sending nor receiving data, it knows the link is available.
- If it has something to send, the primary device sends it.
- It does not know whether the target device is prepared to receive.
- the primary must alert the secondary to the upcoming transmission and wait for an acknowledgment of the secondary's ready status
- Before sending data, the primary creates and transmits a select (SEL) frame, one field of which includes the address of the intended secondary.

Poll

- The *poll* function is used by the primary device to solicit transmissions from the secondary devices.
- When the primary is ready to receive data, it must ask (poll) each device in turn if it has anything to send.
- When the first secondary is approached, it responds either with a NAK frame if it has nothing to send. It responds with data (in the form of a data frame) if it has data to send.
- If the response is negative (a NAK frame), then the primary polls the next secondary in the same manner until it finds one with data to send.
- When the response is positive (a data frame), the primary reads the frame and returns an acknowledgment (ACK frame), verifying its receipt.

Figure 12.19 *Select and poll functions in polling-access method*

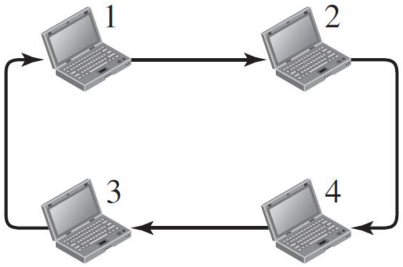


Token Passing

- the stations in a network are organized in a logical ring.
- for each station, there is a *predecessor* and a *successor*.
- The current station is the one that is accessing the channel now.
- a special packet called a ***token*** circulates through the ring.
- The possession of the token gives the station the right to access the channel.

- When a station has some data to send, it waits until it receives the token from its predecessor.
- The station cannot send data until it receives the token again in the next round.
- Token management is needed for this access method.
- The token must be monitored to ensure it has not been lost or destroyed.
- Another function of token management is to assign priorities to the stations and to the types of data being transmitted.
- token management is needed to make low-priority stations release the token to high-priority stations.

Physical ring

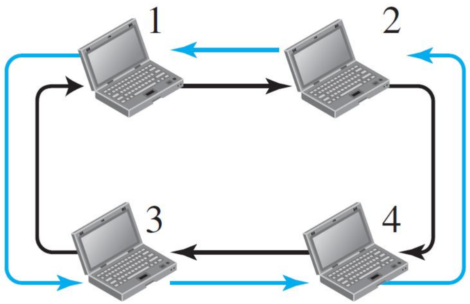


a. Physical ring

- when a station sends the token to its successor, the token cannot be seen by other stations.
- the token does not have to have the address of the next successor as token comes next to successor itself.

- if one of the links—the medium between two adjacent stations—fails, the whole system fails.

Dual Ring



b. Dual ring

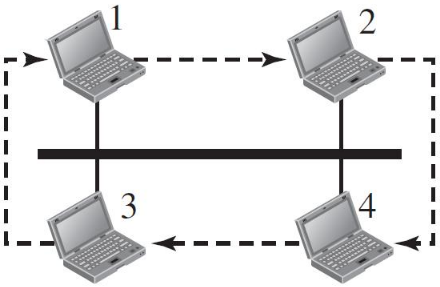
- uses a second (auxiliary) ring which operates in the reverse direction compared with the main ring.

- The second ring is for emergencies only.

- If one of the links in the main ring fails, the system automatically combines the two rings to form a temporary ring.

- for this topology to work, each station needs to have two transmitter ports and two receiver ports

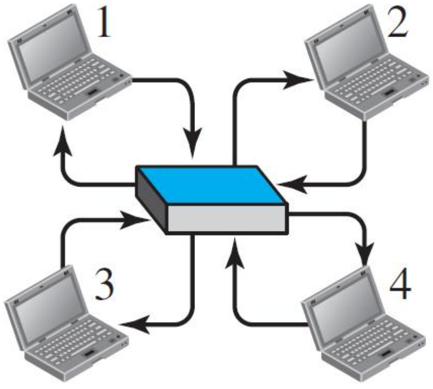
Bus-Ring



c. Bus ring

- also called a token bus.
- The bus, makes a logical ring, because each station knows the address of its successor.
- When a station has finished sending its data, it releases the token and inserts the address of its successor in the token which gets access to the token.

Star Ring



d. Star ring

- the physical topology is a star.
- There is a hub which acts as a connector.

- The wiring inside the hub makes the ring; the stations are connected to this ring through the two wire connections.

- less prone to failure because if a link goes down, it will be bypassed by the hub and the rest of the stations can operate.

- adding and removing stations from the ring is easier.

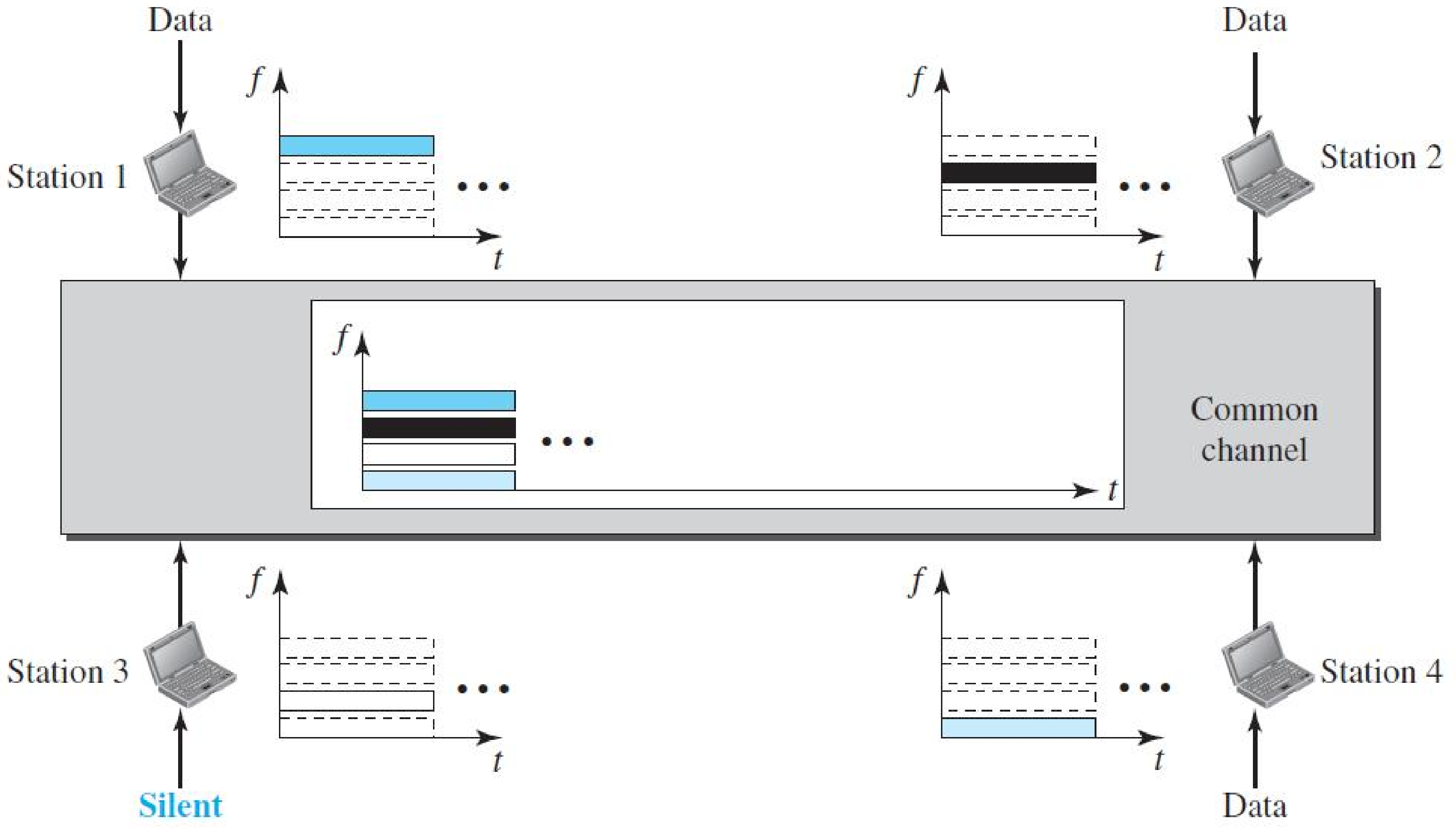
CHANNELIZATION

- **Channelization** (or *channel partition,* as it is sometimes called) is a multiple-access method in which the available bandwidth of a link is shared in time, frequency, or through code, among different stations

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.
- each band is reserved for a specific station
- Each station also uses a bandpass filter to confine the transmitter frequencies.

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



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

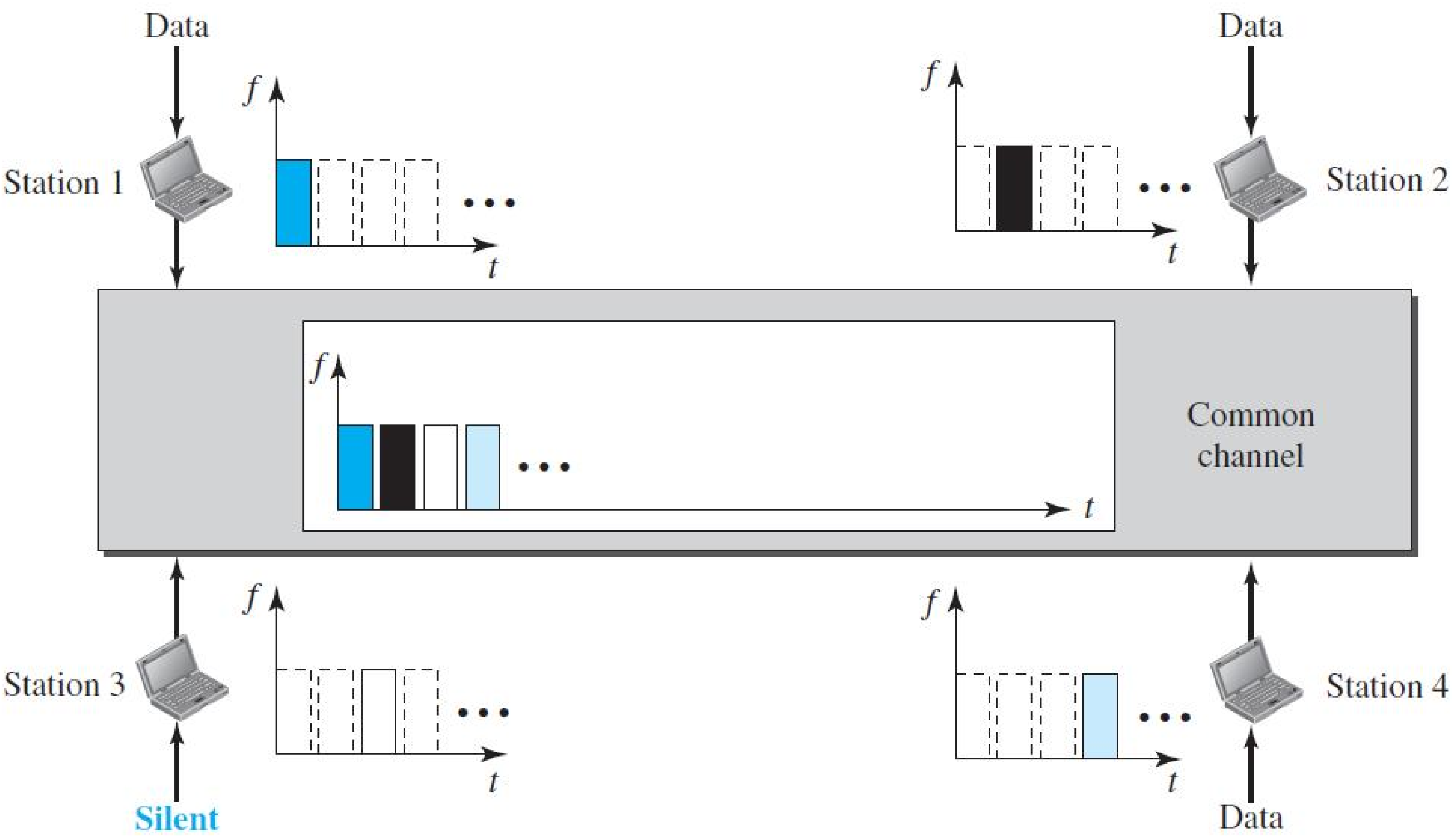
- FDMA, on the other hand, is an access method in the data-link layer.

- There is no physical multiplexer at the physical layer as in FDM

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.
- Each station needs to know the beginning of its slot and the location of its slot.
- To compensate for the delays, we can insert *guard times*.
- Synchronization is normally accomplished by having some synchronization bits.

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



Code-division multiple access (CDMA)

- CDMA differs from FDMA in that only one channel occupies the entire bandwidth of the link.
- It differs from TDMA in that all stations can send data simultaneously.
- CDMA simply means communication with different codes.
- CDMA is based on coding theory.
- 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.
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 each sequence. This is called the ***inner product*** of two equal sequences.
4. If we multiply two different sequences, element by element, and add the results, we get 0. This is called the *inner product* of two different sequences

Figure 12.26 *Sharing channel in CDMA*

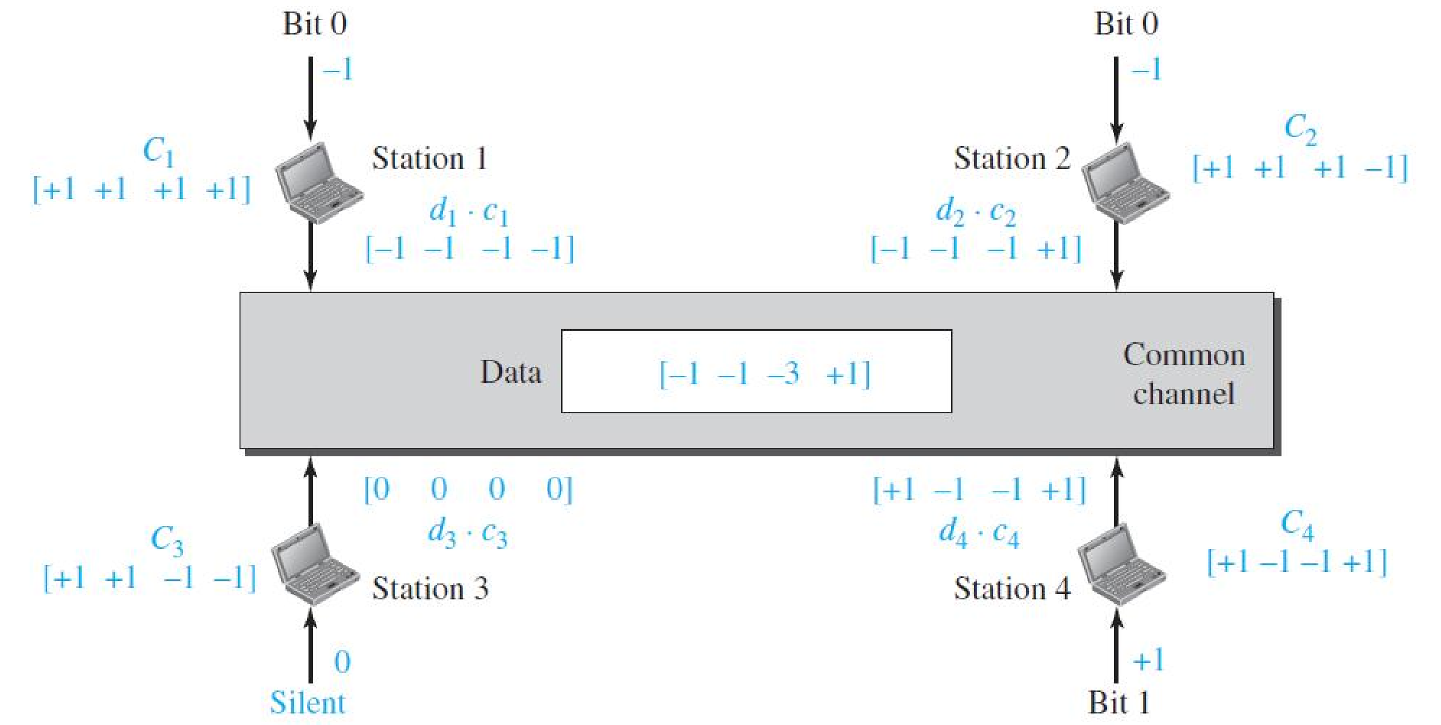
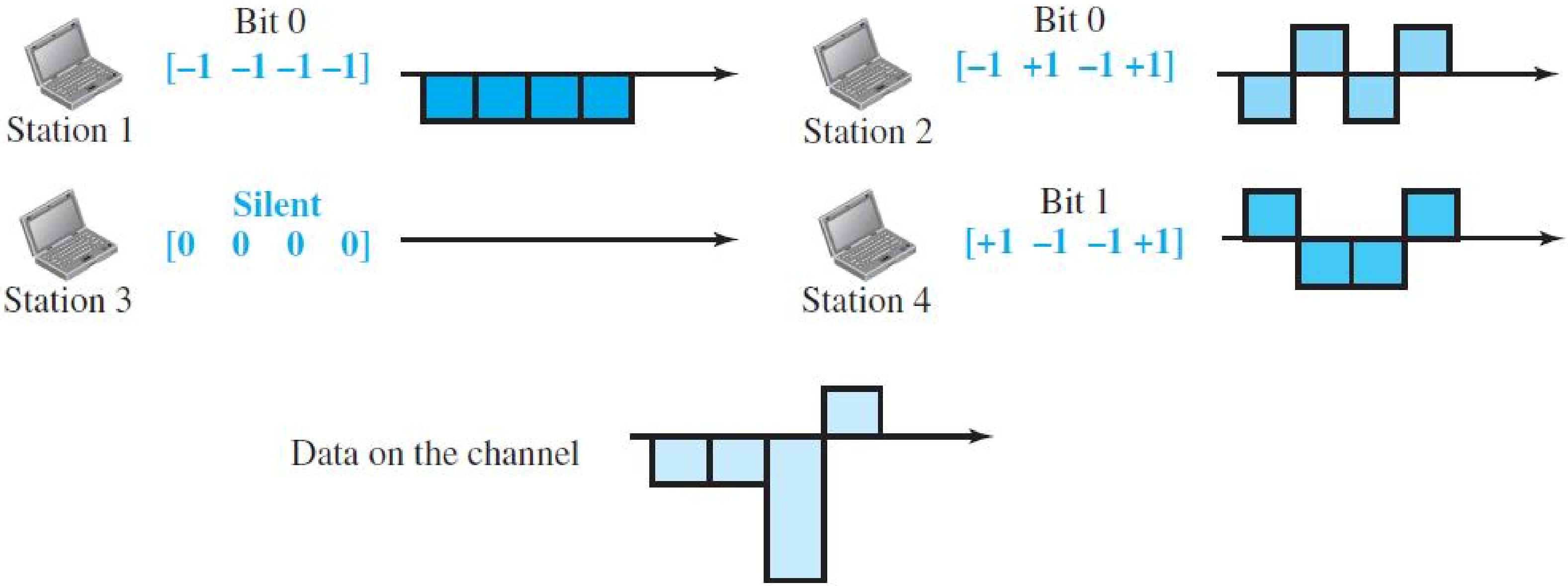


Figure 12.27 *Digital signal created by four stations in CDMA*



Walsh Tables

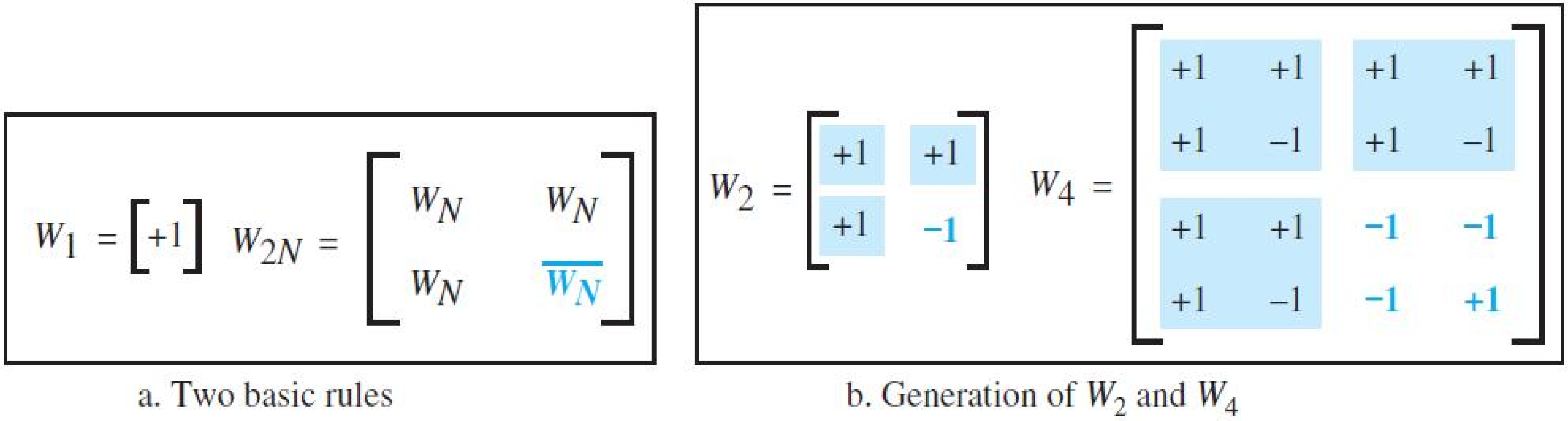
- To generate chip sequences.
- is a two-dimensional table with an equal number of rows and columns
- each row is a sequence of chips
- W_1 for a one-chip sequence has one row and one column. (choose –1 or +1)
- if we know the table for N sequences W_N, we can create the table for 2N sequences W_{2N}

- W2 can be made from four W1s

- W4 can be made of four W2s, with the last one the complement of W2

- W8 is composed of four W4s

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



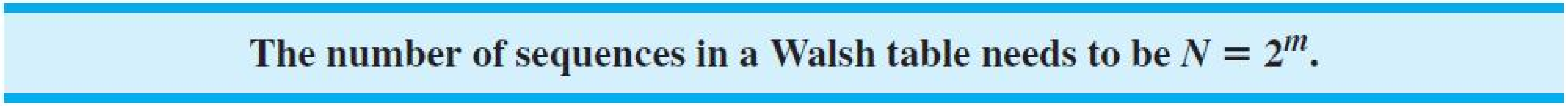
$$W_1 = \begin{bmatrix} +1 \end{bmatrix} \quad W_{2N} = \begin{bmatrix} W_N & W_N \\ W_N & \overline{W_N} \end{bmatrix}$$

a. Two basic rules

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

b. Generation of W_2 and W_4

- the number of sequences, *N*, needs to be a power of 2.



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