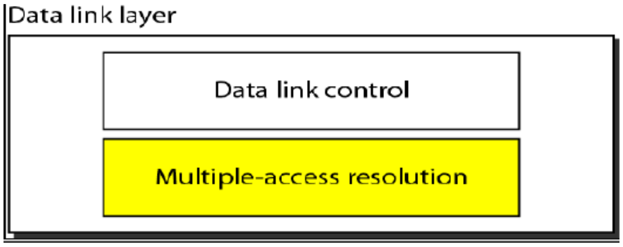
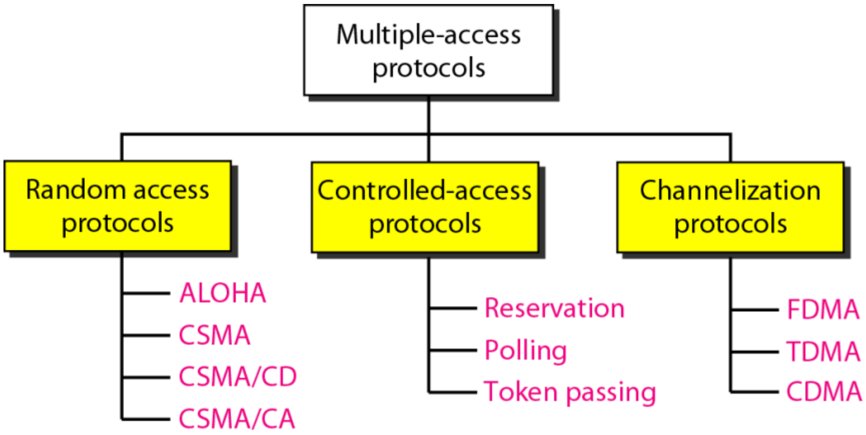
RANDOM ACCESS PROTOCOLS

We can consider the data link layer as two sub layers. The upper sub layer is responsible for data link control, and the lower sub layer is responsible for resolving access to the shared media.



The upper sub layer that is responsible for flow and error control is called the logical link control (LLC) layer; the lower sub layer that is mostly responsible for multiple access resolution is called the media access control (MAC) layer. When nodes or stations are connected and use a common link, called a multipoint or broadcast link, we need a multiple-access protocol to coordinate access to the link.

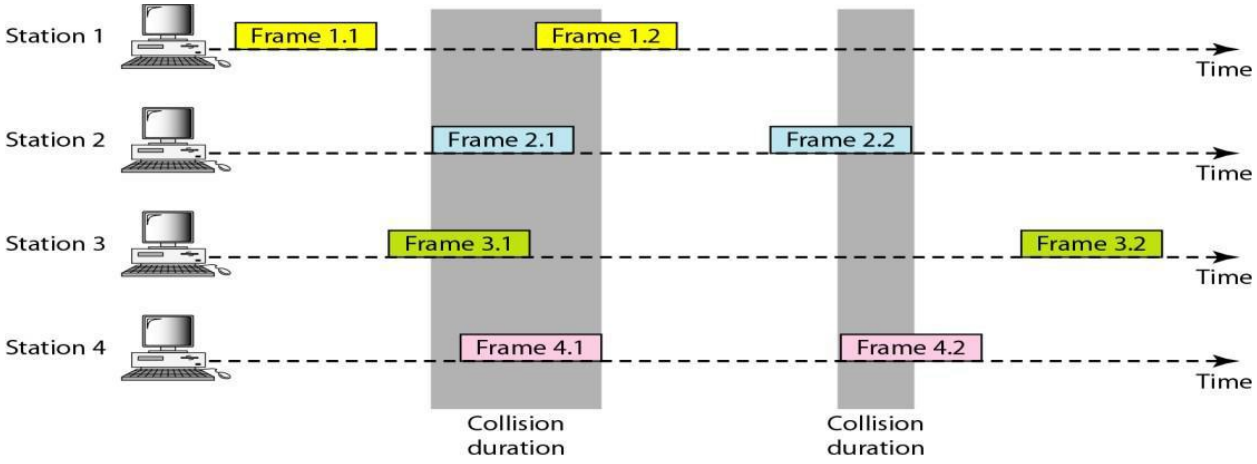


RANDOM ACCESS METHOD

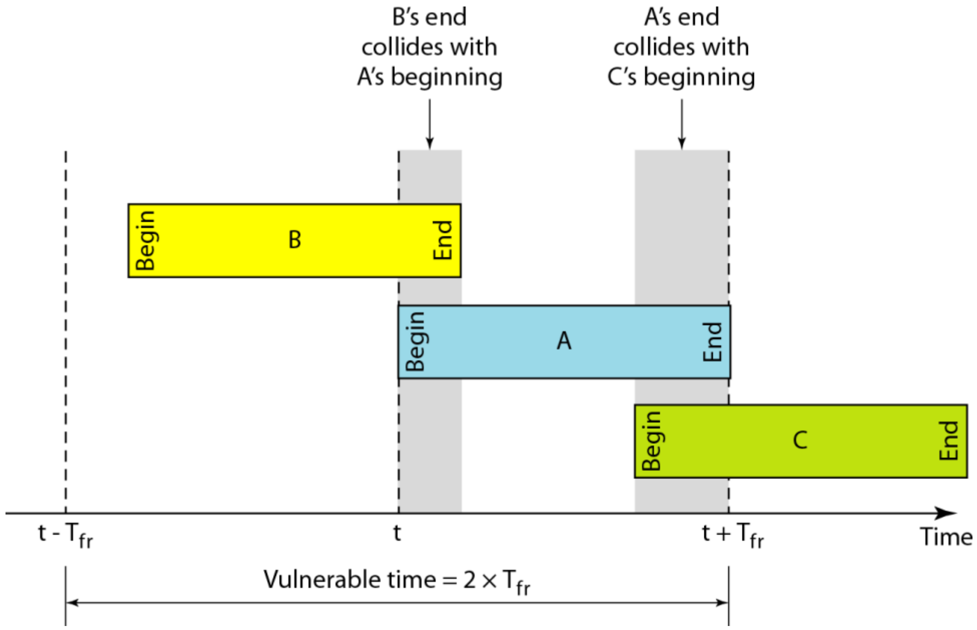
- In random access or contention methods, no station is superior to another station and none is assigned the control over another.
- Two features give this method its name. First, there is no scheduled time for a station to transmit. Transmission is random among the stations. That is why these methods are called *random access.*
- Second, no rules specify which station should send next. Stations compete with one another to access the medium. That is why these methods are also called *contention* methods.

1. Pure ALOHA

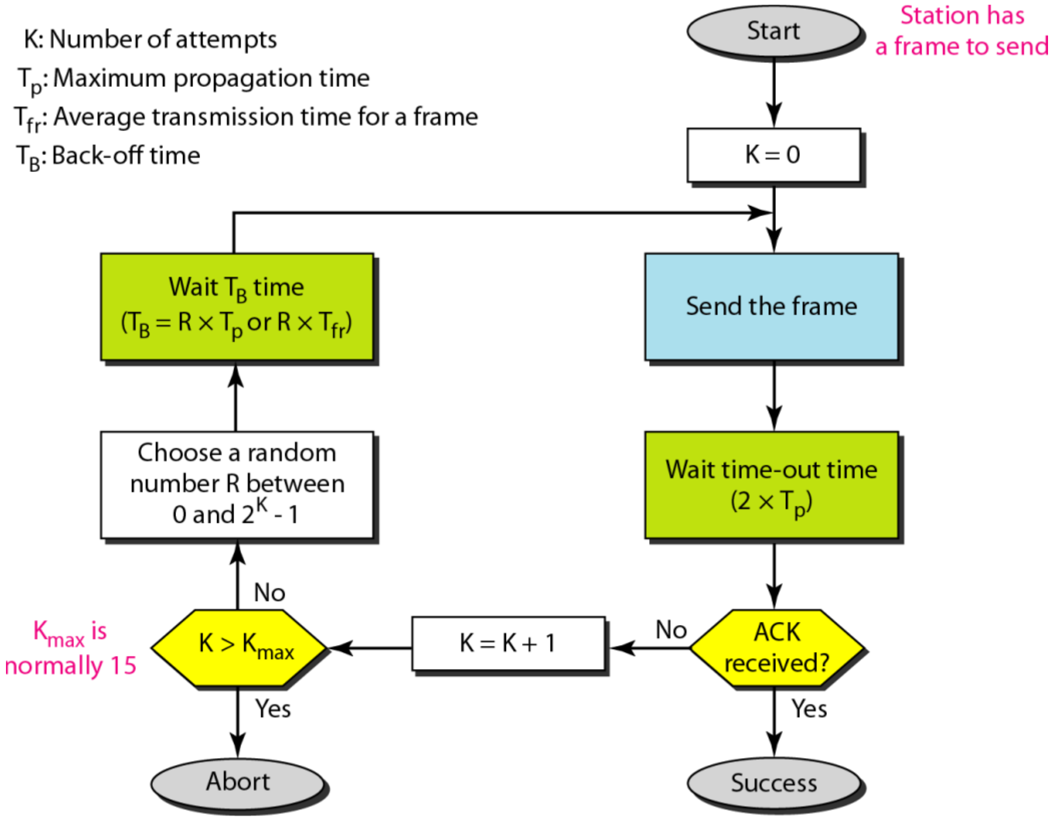
- The original ALOHA protocol is called pure ALOHA. This is a simple, but elegant protocol. The idea is that each station sends a frame whenever it has a frame to send. However, since there is only one channel to share, there is the possibility of collision between frames from different stations.
- Below Figure shows an example of frame collisions in pure ALOHA.



- In pure ALOHA, the stations transmit frames whenever they have data to send.
- When two or more stations transmit simultaneously, there is collision and the frames are destroyed.
- In pure ALOHA, whenever any station transmits a frame, it expects the acknowledgement from the receiver.
- If acknowledgement is not received within specified time, the station assumes that the frame (or acknowledgement) has been destroyed.
- If the frame is destroyed because of collision the station waits for a random amount of time and sends it again. This waiting time must be random otherwise same frames will collide again and again.
- Therefore pure ALOHA dictates that when time-out period passes, each station must wait for a random amount of time before resending its frame. This randomness will help avoid more collisions.
- **Vulnerable time :** The **vulnerable time** is the length of time in which there is a possibility of collision. We assume that the stations send fixed length frames with each frame taking *Tfr* time to send. Below figure shows the vulnerable time for station A.

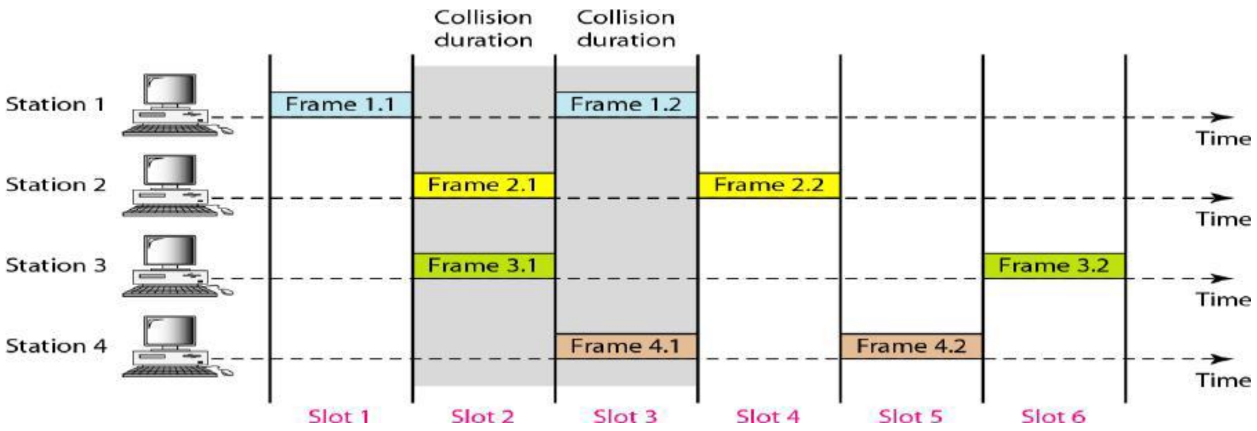

B's end collides with A's beginning
A's end collides with C's beginning
Begin
B
End
Begin
A
End
Begin
C
End
t - T_fr
t
t + T_fr
Time
Vulnerable time = 2 × T_fr

- Station A sends a frame at time *t.* Now imagine station B has already sent a frame between *t* -T_{fr} and *t.* This leads to a collision between the frames from station A and station B. The end of B's frame collides with the beginning of A's frame. On the other hand, suppose that station C sends a frame between *t* and *t*+T_{fr} .
- Here, there is a collision between frames from station A and station C. The beginning of C's frame collides with the end of A's frame
- Looking at Figure, we see that the vulnerable time, during which a collision may occur in pure ALOHA, is 2 times the frame transmission time.
- Pure ALOHA vulnerable time = 2 x T_{fr}


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

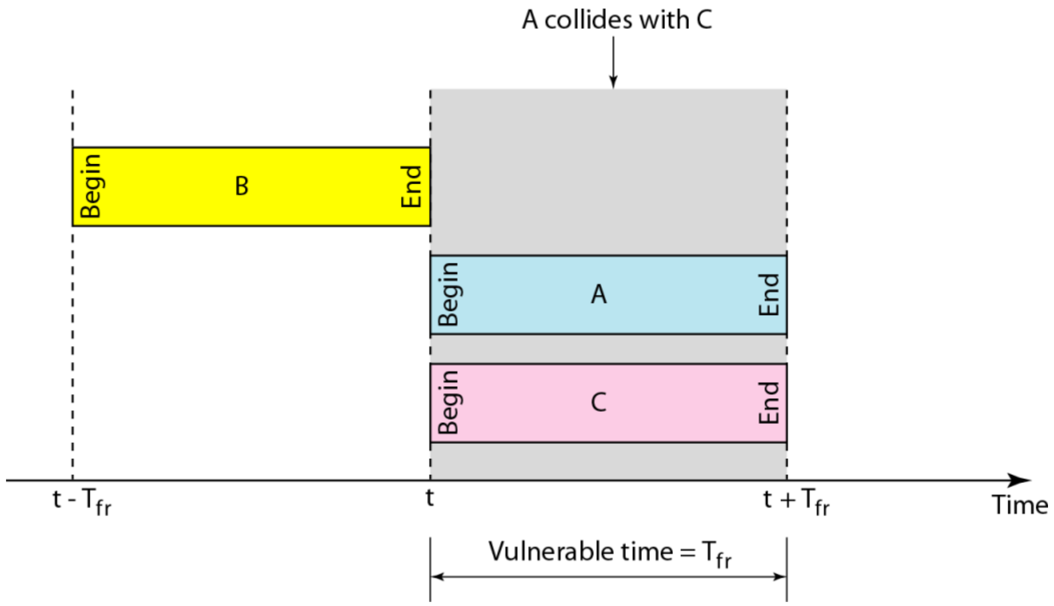
2. ***Slotted ALOHA***

- Pure ALOHA has a vulnerable time of 2 x T_{fr} . This is so because there is no rule that defines when the station can send. A station may send soon after another station has started or soon before another station has finished.
- Slotted ALOHA was invented to improve the efficiency of pure ALOHA. In slotted ALOHA we divide the time into slots of T_{fr} s and force the station to send only at the beginning of the time slot. Figure 3 shows an example of frame collisions in slotted ALOHA



- Because a station is allowed to send only at the beginning of the synchronized time slot, if a station misses this moment, it must wait until the beginning of the next time slot. This means that the station which started at the beginning of this slot has already finished sending its frame. 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* Figure 4 shows the situation Below fig shows that the vulnerable time for slotted ALOHA is one-half that of pure ALOHA.

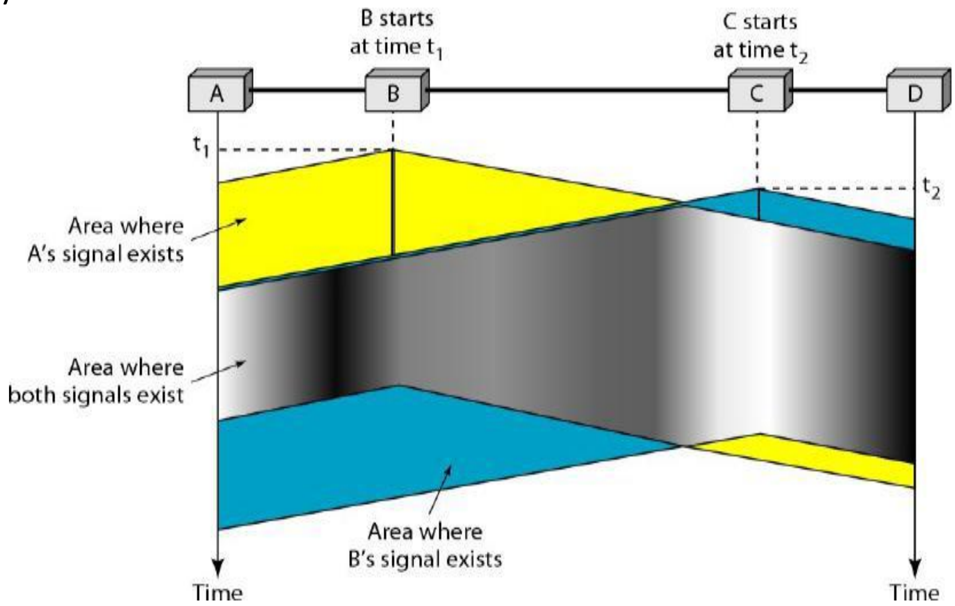
  Slotted ALOHA vulnerable time = Tfr



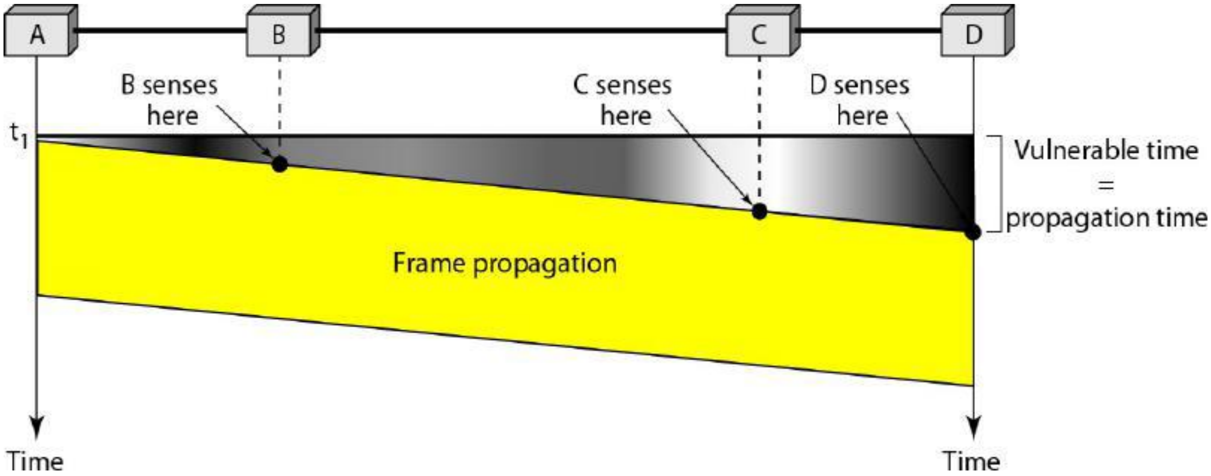
3. Carrier Sense Multiple Access (CSMA)

- To minimize the chance of collision and, therefore, increase the performance, the CSMA method was developed. The chance of collision can be reduced if a station senses the medium before trying to use it.
- Carrier sense multiple access (CSMA) requires that each station first listen to the medium (or check the state of the medium) before sending. In other words, CSMA is based on the principle "sense before transmit" or "listen before talk."
- CSMA can reduce the possibility of collision, but it cannot eliminate it. The reason for this is shown in below Figure. Stations are connected to a shared channel (usually a dedicated medium).

- The possibility of collision still exists because of propagation delay; 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.


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

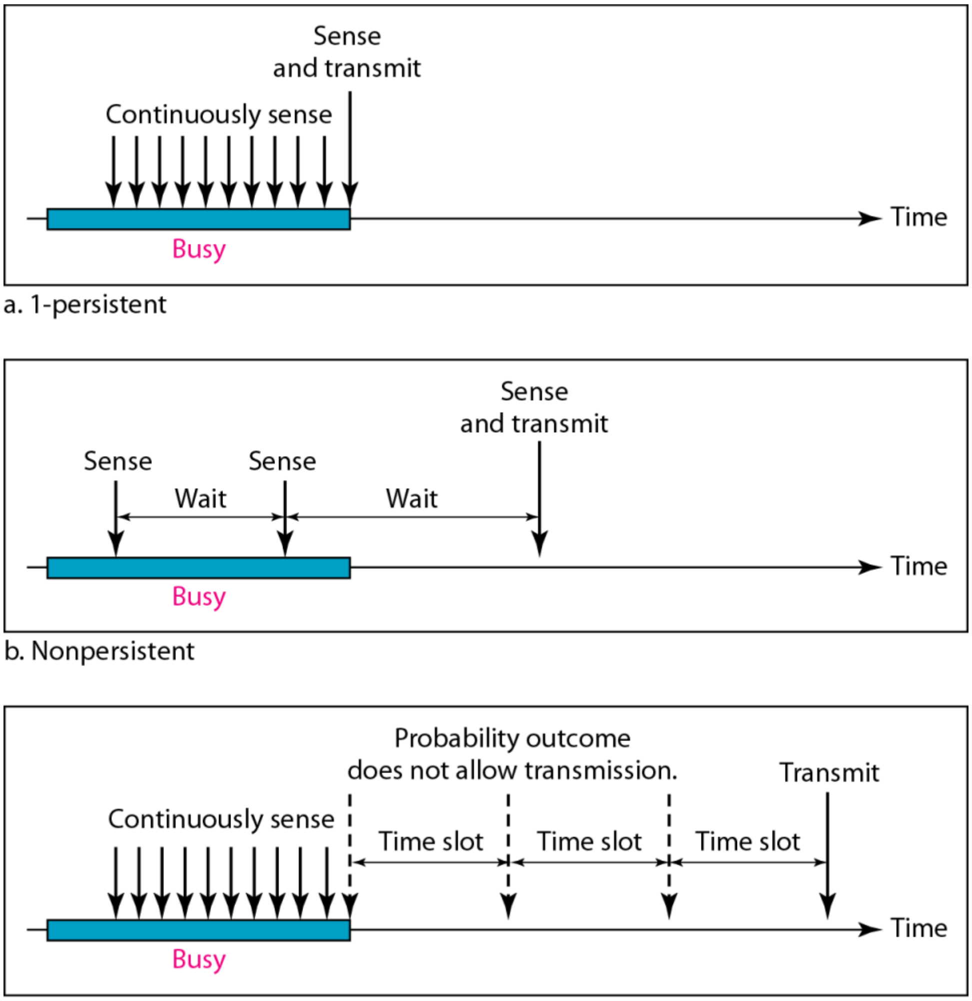
- ***Vulnerable Time***
  - The vulnerable time for CSMA is the propagation time Tp . This is the time needed for a signal to propagate from one end of the medium to the other.
  - When a station sends a frame, and any other station tries to send a frame during this time, a collision will result. But 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



- ***Persistence Methods***

  What should a station do if the channel is busy? What should a station do if the channel is idle? Three methods have been devised to answer these questions:

1. 1-persistent method
2. non-persistent method
3. p-persistent method



a. 1-persistent

b. Nonpersistent

c. p-persistent

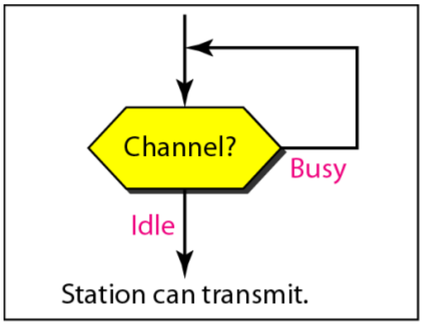
- **1-Persistent:** In this method, after the station finds the line idle, it sends its frame immediately (with probability 1). This method has the highest chance of collision because two or more stations may find the line idle and send their frames immediately.
- **Non-persistent:** a station that has 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. This 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. However, this method reduces the efficiency of the network because the medium remains idle when there may be stations with frames to send.
- **P-Persistent:** This is used if the channel has time slots with slot duration equal to or greater than the maximum propagation time. The p-persistent approach combines the advantages of the other two strategies. It reduces the chance of collision and improves efficiency.

  In this method, after the station finds the line idle it follows these steps:

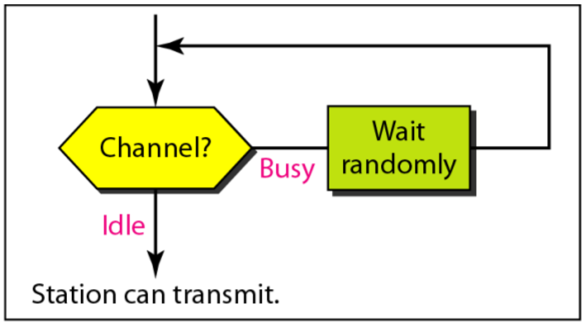
  1. With probability *p,* the station sends its frame.

  2. With probability $q = 1 - P$, the station waits for the beginning of the next time slot and checks the line again.

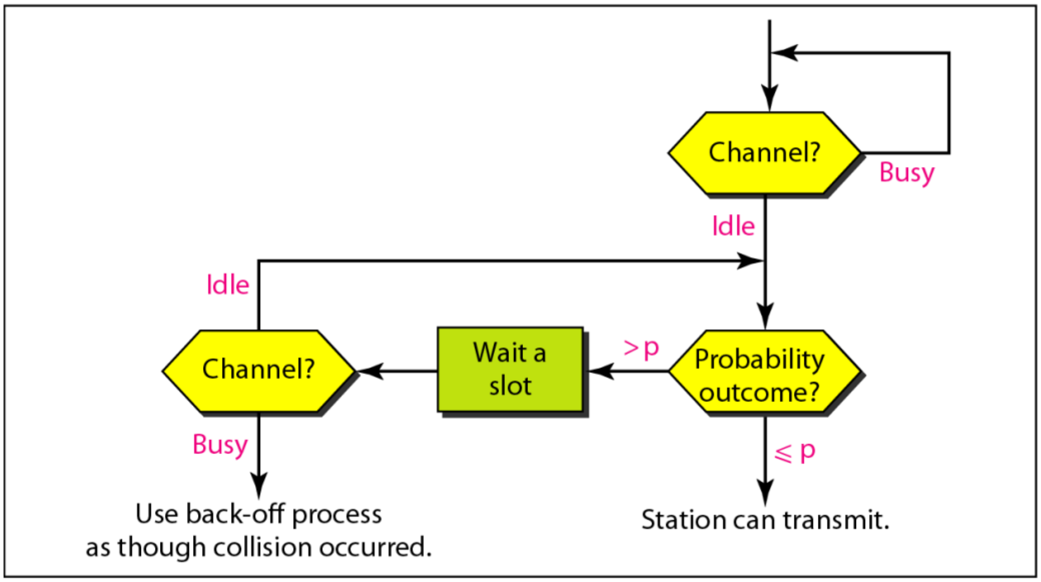
a. If the line is idle, it goes to step 1.
b. If the line is busy, it acts as though a collision has occurred and uses the back-off procedure.



a. 1-persistent



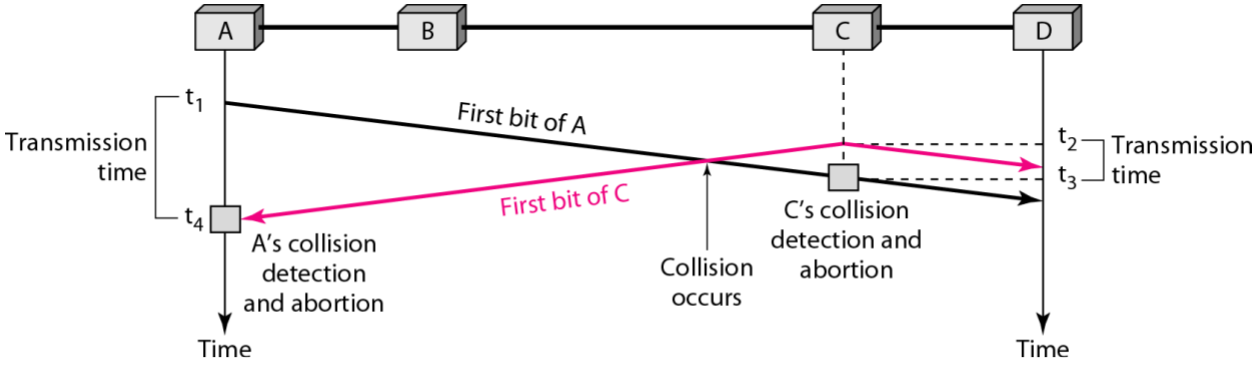
b. Nonpersistent



c. p-persistent

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

- The CSMA method does not specify the procedure following a collision. Carrier sense multiple access with collision detection (CSMA/CD) augments the algorithm to handle the collision.
- In this method, a station monitors the medium after it sends a frame to see if the transmission was successful. If so, the station is finished. If, however, there is a collision, the frame is sent again.
- To better understand CSMA/CD, let us look at the first bits transmitted by the two stations involved in the collision. Although each station continues to send bits in the frame until it detects the collision, we show what happens as the first bits collide. In below Figure, stations A and C are involved in the collision.



At time *t*1, station A has executed its persistence procedure and starts sending the bits of its frame. At time *t2,* station C has not yet sensed the first bit sent by A. Station C executes its persistence procedure and starts sending the bits in its frame, which propagate both to the left and to the right.

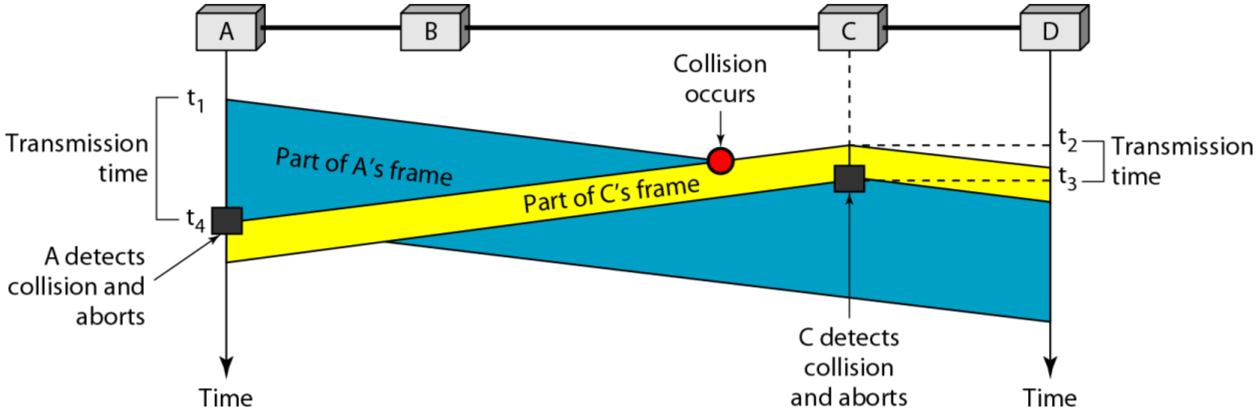
The collision occurs sometime after time *t2.*

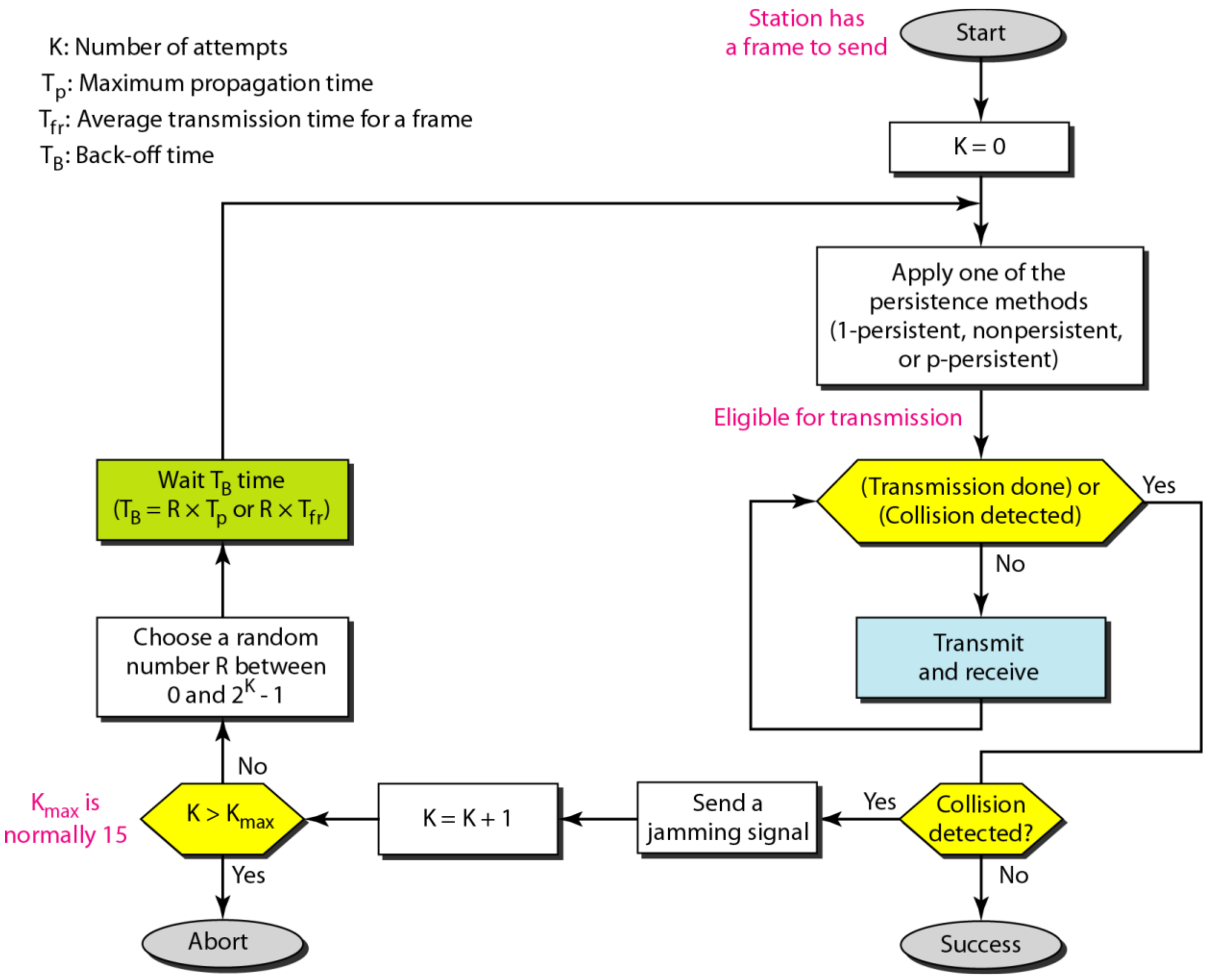
Station C detects a collision at time *t3* when it receives the first bit of A's frame. Station C immediately (or after a short time, but we assume immediately) aborts transmission.

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.*

Minimum Frame Size

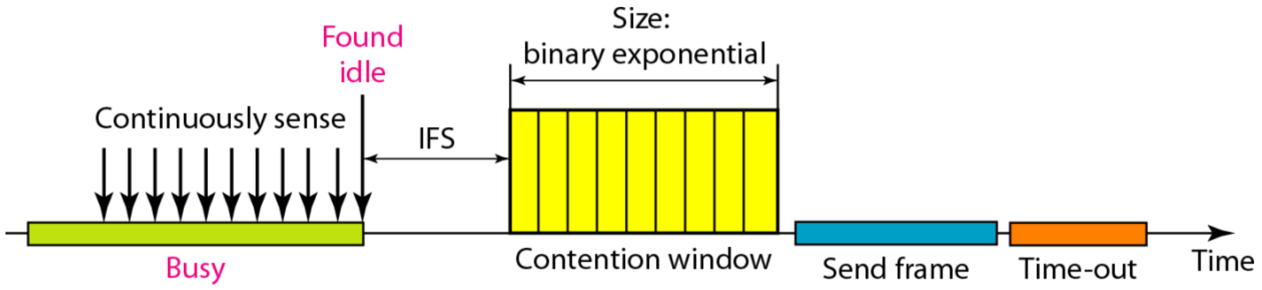
- For *CSMAlCD* to work, we need a restriction on the frame size. Before sending the last bit of the frame, the sending station must detect a collision, if any, and abort the transmission. This is so because the station, once the entire frame is sent, does not keep a copy of the frame and does not monitor the line for collision detection. Therefore, the frame transmission time T_{fr} must be at least two times the maximum propagation time *Tp.*
- To understand the reason, let us think about the worst-case scenario. If the two stations involved in a collision are the maximum distance apart, the signal from the first takes time *Tp* to reach the second and the effect of the collision takes another time *Tp* to reach the first. So the requirement is that the first station must still be transmitting after *2Tp.*




K: Number of attempts
T_p: Maximum propagation time
T_{fr}: Average transmission time for a frame
T_B: Back-off time
Station has a frame to send
Start
K = 0
Apply one of the persistence methods (1-persistent, nonpersistent, or p-persistent)
Eligible for transmission
(Transmission done) or (Collision detected)
Yes
No
Transmit and receive
Wait T_B time ($T_B = R \times T_p$ or $R \times T_{fr}$)
Choose a random number R between 0 and $2^K - 1$
No
K_{max} is normally 15
$K > K_{max}$
Yes
Abort
K = K + 1
Send a jamming signal
Yes
Collision detected?
No
Success

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

We need to avoid collisions on wireless networks because they cannot be detected. Carrier sense multiple access with collision avoidance *(CSMAICA)* was invented for wirelesss network. Collisions are avoided through the use of CSMA/CA's three strategies: the inter frame space, the contention window, and acknowledgments.



Inter frame Space (IFS)

- First, 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 inter frame space or IFS.

- Even though the channel may appear idle when it is sensed, a distant station may have already started transmitting. The distant station's signal has not yet reached this station. The IFS time allows the front of the transmitted signal by the distant station to reach this station. If after the IFS time the channel is still idle, the station can send, but it still needs to wait a time equal to the contention time.
- The IFS variable can also be used to prioritize stations or frame types. For example, a station that is assigned shorter IFS has a higher priority. In CSMA/CA, the IFS can also be used to define the priority of a station or a frame.

Contention Window

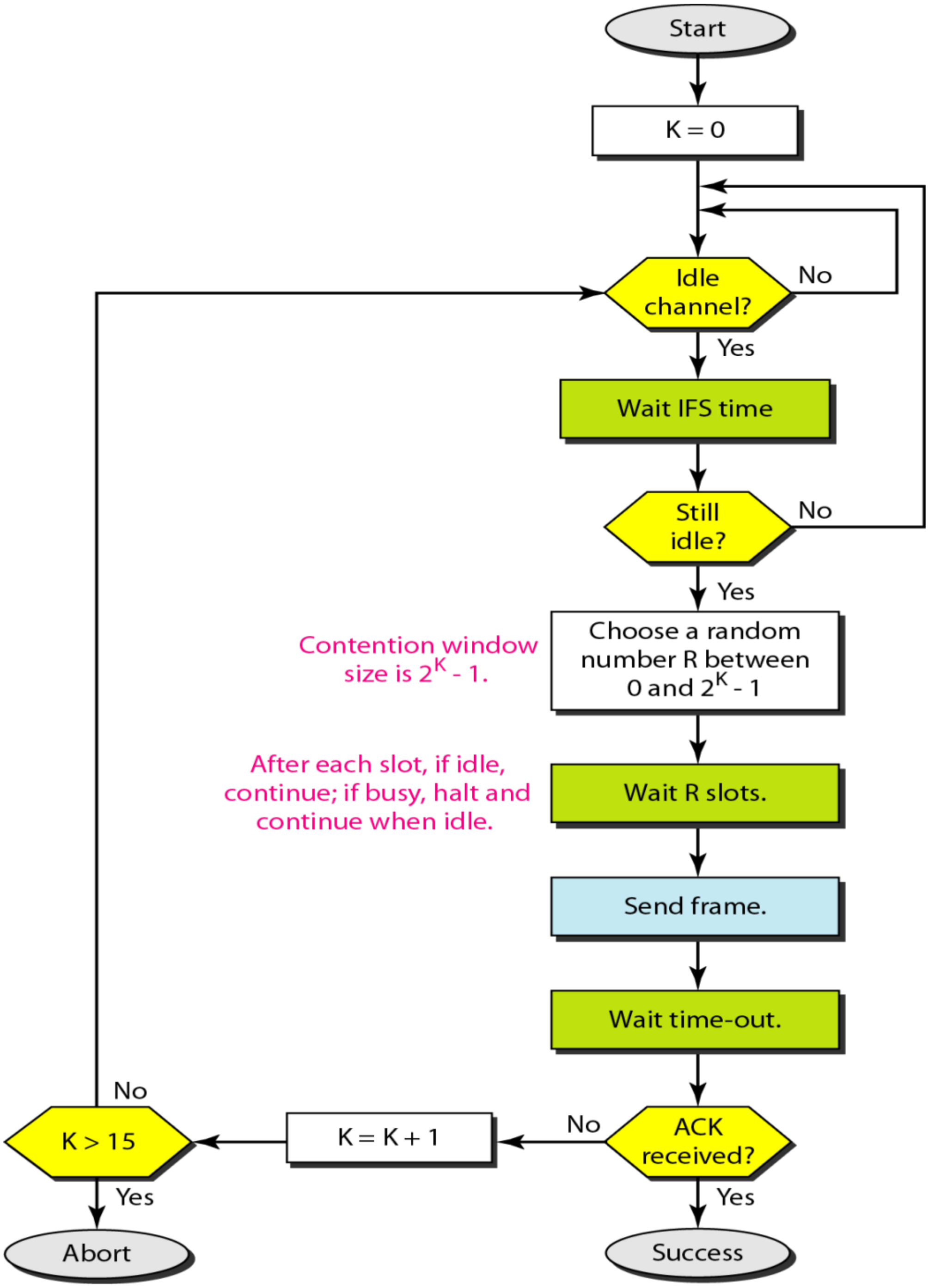
- The contention window is an amount of time divided into slots. A station that is ready to send chooses a random number of slots as its wait time. The number of slots in the window changes according to the binary exponential back-off strategy. This means that it is set to one slot the first time and then doubles each time the station cannot detect an idle channel after the IFS time. This is very similar to the p-persistent method except that a random outcome defines the number of slots taken by the waiting station.
- One interesting point about the contention window is that the station needs to sense the channel after each time slot. However, 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. This gives priority to the station with the longest waiting time.

Acknowledgment

- With all these precautions, there still may be a collision resulting in destroyed data. In addition, the data may be corrupted during the transmission. The positive acknowledgment and the time-out timer can help guarantee that the receiver has received the frame.

Procedure for CSMA protocol with collision avoidance:

- The station ready to transmit, senses the line by using one of the persistent strategies.
- As soon as it finds the line to be idle, the station waits for IFS (Inter frame space) amount of time.
- If then waits for some random time and sends the frame.
- After sending the frame, it sets a timer and waits for the acknowledgement from the receiver.
- If the acknowledgement is received before expiry of the timer, then the transmission is successful.
- But if the transmitting station does not receive the expected acknowledgement before the timer expiry then it increments the back-off parameter, waits for the back off time and re senses the line


Start
K = 0
Idle channel?
No
Yes
Wait IFS time
Still idle?
No
Yes
Choose a random number R between 0 and 2^K - 1
Contention window size is 2^K - 1.
After each slot, if idle, continue; if busy, halt and continue when idle.
Wait R slots.
Send frame.
Wait time-out.
ACK received?
No
Yes
K = K + 1
K > 15
No
Yes
Abort
Success

CONTROLLED ACCESS PROTOCOLS

In controlled access, the stations seek information from one another to find which station has the right to send. It allows only one node to send at a time, to avoid collision of messages on shared medium.

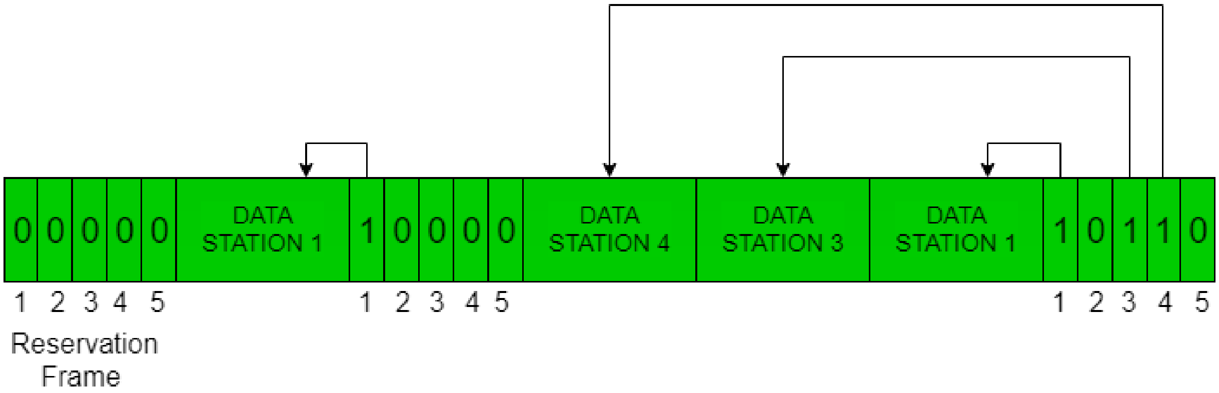
The three controlled-access methods are:

1. Reservation
2. Polling
3. Token Passing

1. Reservation

- In the reservation method, a station needs to make a reservation before sending data. The time line has two kinds of periods:
  1. Reservation interval of fixed time length
  2. Data transmission period of variable frames.
- If there are M stations, the reservation interval is divided into M slots, and each station has one slot. Suppose if station 1 has a frame to send, it transmits 1 bit during the slot 1. No other station is allowed to transmit during this slot.
- In general, i th station may announce that it has a frame to send by inserting a 1 bit into i th slot. After all N slots have been checked, each station knows which stations wish to transmit.
- The stations which have reserved their slots transfer their frames in that order. After data transmission period, next reservation interval begins. Since everyone agrees on who goes next, there will never be any collisions.

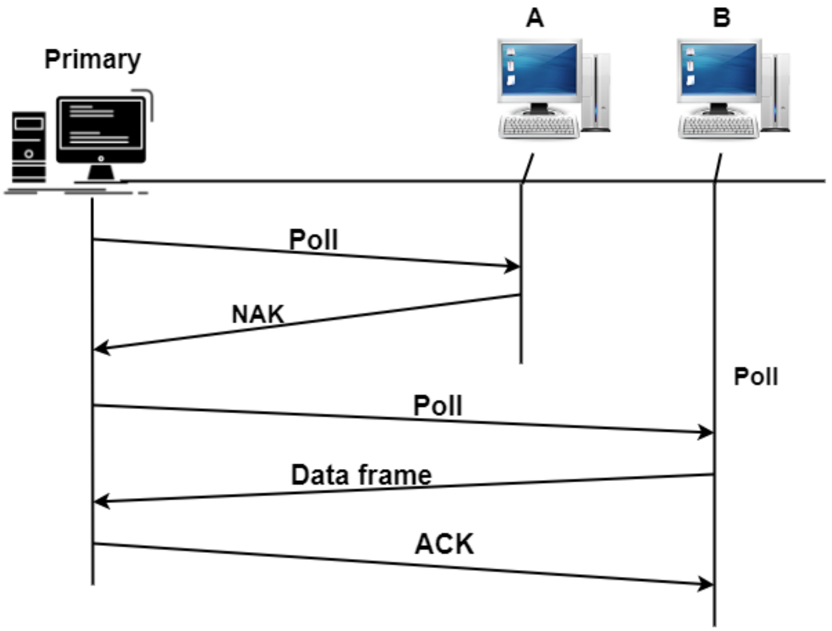
The following figure shows a situation with five stations and a five slot reservation frame. In the first interval, only stations 1, 3, and 4 have made reservations. In the second interval, only station 1 has made a reservation.



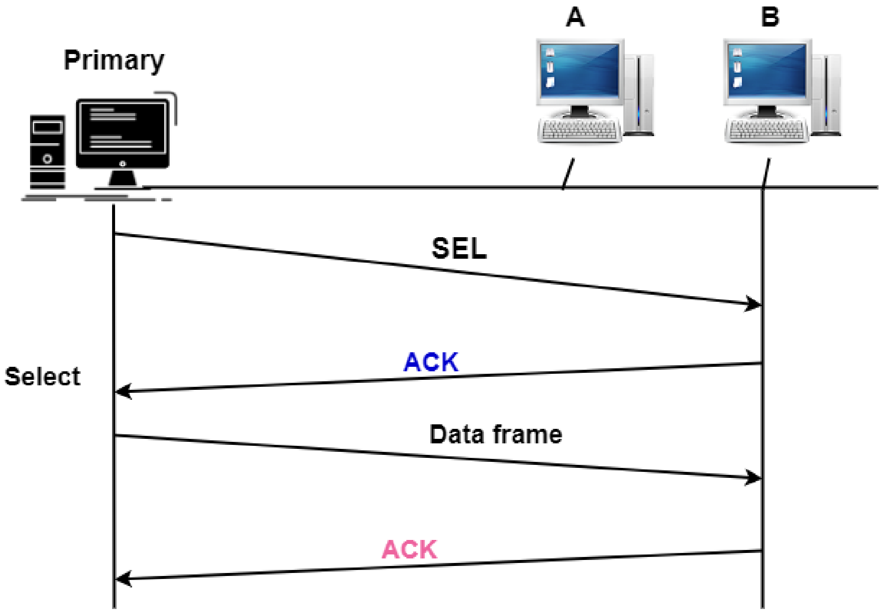
2. Polling

- Polling process is similar to the roll-call performed in class. Just like the teacher, a controller sends a message to each node in turn.
- In this, one act as a primary station (controller) and the others are secondary stations. All data exchanges must be made through the controller.
- The message sent by the controller contains the address of the node being selected for granting access.
- Although all nodes receive the message but the addressed one responds to it and sends data, if any. If there is no data, usually a “poll reject” (NAK) message is sent back.
- Problems include high overhead of the polling messages and high dependence on the reliability of the controller.

Polling method:



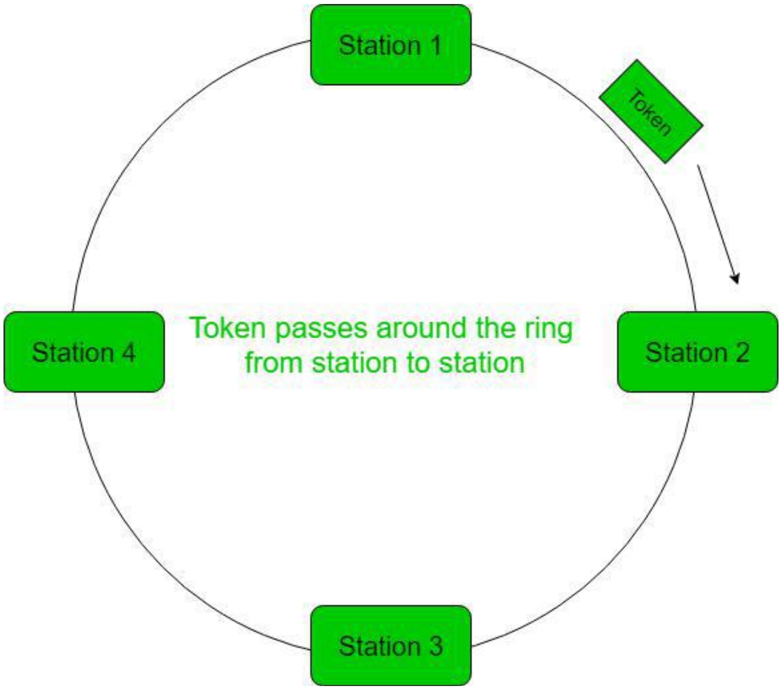
Selection method:



<u>3.Token Passing</u>

- In token passing scheme, the stations are connected logically to each other in form of ring and access of stations is governed by tokens.
- A token is a special bit pattern or a small message, which circulate from one station to the next in the some predefined order.
- In Token ring, token is passed from one station to another adjacent station in the ring whereas in case of Token bus, each station uses the bus to send the token to the next station in some predefined order.

- In both cases, token represents permission to send. If a station has a frame queued for transmission when it receives the token, it can send that frame before it passes the token to the next station. If it has no queued frame, it passes the token simply.
- After sending a frame, each station must wait for all N stations (including itself) to send the token to their neighbors and the other N – 1 stations to send a frame, if they have one.
- There exists problem like duplication of token or token is lost or insertion of new station, removal of a station, which need be tackled for correct and reliable operation of this scheme.

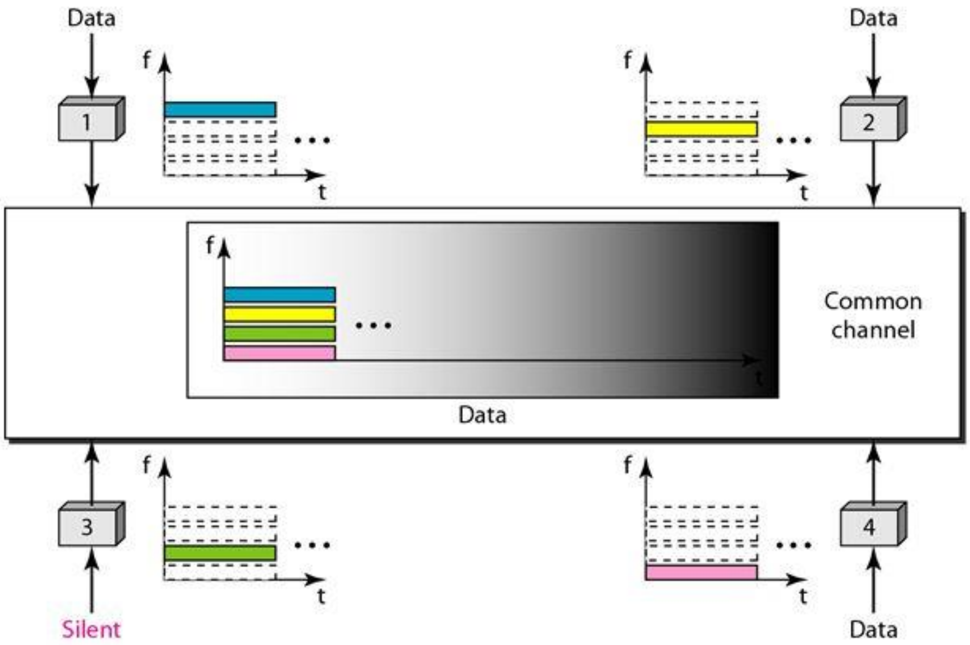


CHANNELIZATION METHOD

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. The three channelization protocols are FDMA, TDMA, and CDMA.

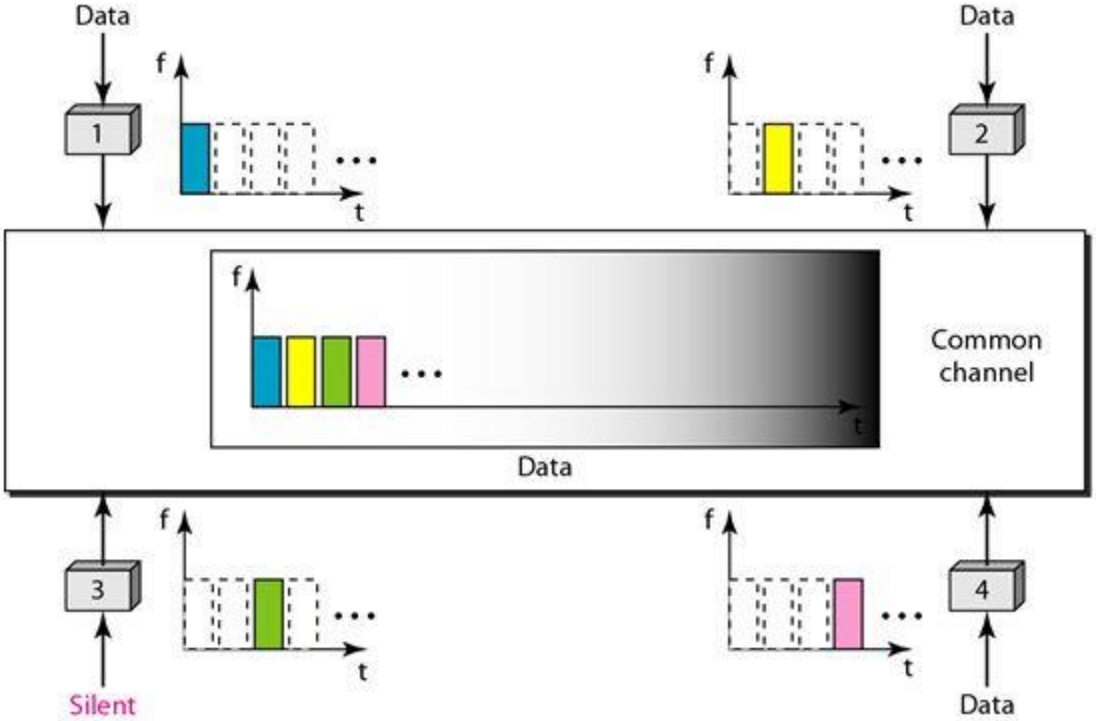
1. The Frequency-Division Multiple Access (FDMA):

- 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 band-pass filter to confine the transmitter frequencies. To prevent station interferences, the allocated bands are separated from one another by small guard bands. The following figure shows the idea of FDMA.



2.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 is assigned time slot. The following figure shows the idea behind TDMA.
- 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.
- Synchronization is normally accomplished by having some synchronization bits at the beginning of each slot.



3.Code-Division Multiple Access (CDMA):

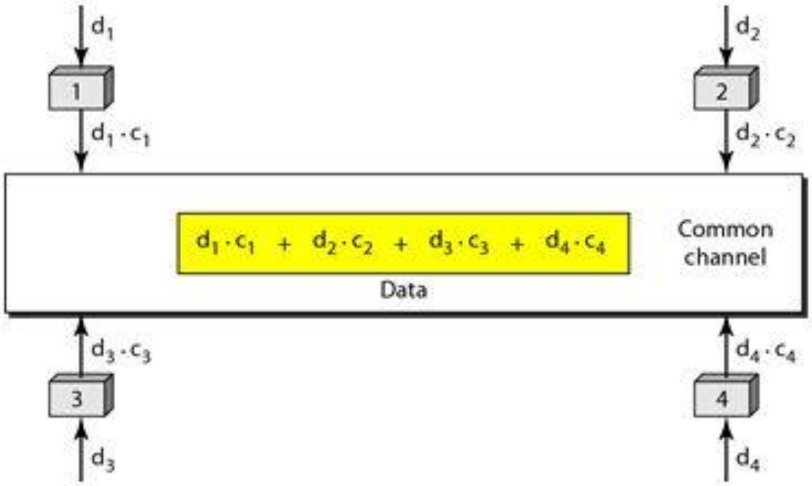
CDMA simply means communication with different codes. CDMA differs from FDMA because only one channel occupies the entire bandwidth of the link. It differs from TDMA because all stations can send data simultaneously; there is no timesharing.

Implementation:

- Let us assume we have four stations 1, 2, 3, and 4 connected to the same channel. The data from station 1 are d1 , from station 2 are d2, and so on. The code assigned to the first station is c1, to the second is c2, and so on. We assume that the assigned codes have two properties.

    1. If we multiply each code by another, we get 0.
    2. If we multiply each code by itself, we get 4 (the number of stations).

- With these two properties in mind, how the above four stations can send data using the same common channel, as shown in the following figure.

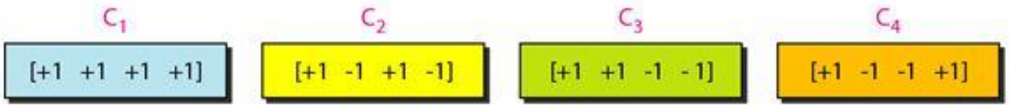


- Station 1 multiplies (a special kind of multiplication, as we will see) its data by its code to get d1.c1. Station 2 multiplies its data by its code to get d2.c2. And so on. The data that go on the channel are the sum of all these terms, as shown in the box.

- Any station that wants to receive data from one of the other three multiplies the data on the channel by the code of the sender. For example, suppose stations 1 and 2 are talking to each other. Station 2 wants to hear what station 1 is saying. It multiplies the data on the channel by c1 the code of station1.
  Because (c1.c1) is 4, but (c2 . c1), (c3. c1), and (c4 .c1) are all 0s, station 2 divides the result by 4 to get the data from station1.
  Data = (d1.c1+d2.c2+d3.c3+d4.c4).c1 = c1. d1. c1+ c1. d2. c2+ c1. d3. c3+ c1. d4. c4= 4d1

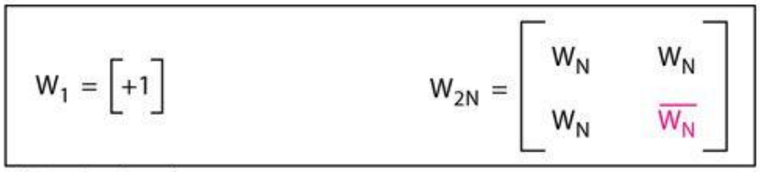
Chips:

CDMA is based on coding theory. Each station is assigned a code, which is a sequence of numbers called chips, as shown in the following figure. The codes are for the previous example.

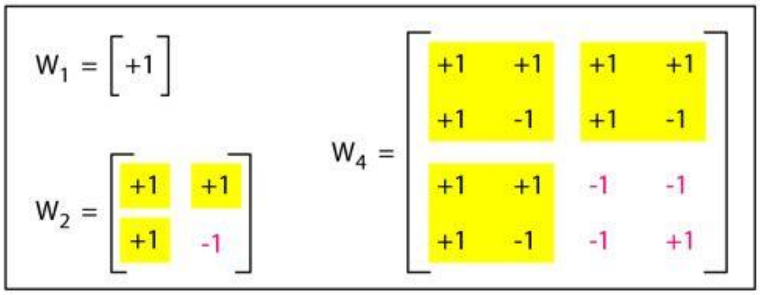


Sequence Generation:

To generate chip sequences, we use a Walsh table, which is a two-dimensional table with an equal number of rows and columns, as shown in the following figure.


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

a. Two basic rules


$W_1 = [+1]$
$W_2 = \begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix}$
$W_4 = \begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{bmatrix}$

b. Generation of W_1, W_2, and W_4

In the ***Walsh table***, each row is a sequence of chips. W_1 for a one-chip sequence has one row and one column. We can choose –1 or +1 for the chip for this trivial table (we chose +1).

- According to Walsh, if we know the table for N sequences W_N we can create the table for 2N sequences W_{2N}, as shown in Figure. The W_N with the overbar of W_N stands for the complement of W_N , where each +1 is changed to -1 and vice versa.
- The above figure also shows how we can create W_2 and W_4 from W_1. After we select W_1, W_2 can be made from four W_1's, with the last one the complement of W_1. After W_2 is generated, W_4 can be made of four W_2's, with the last one the complement of W_2. Of course, W_8 is composed of four W_4's, and so on. Note that after W_N is made, each station is assigned a chip corresponding to a row.
- Something we need to emphasize is that the number of sequences N needs to be a power of 2. In other words, we need to have $N = 2^m$.

We need to know that we did not choose the sequences randomly; they were carefully selected. 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,

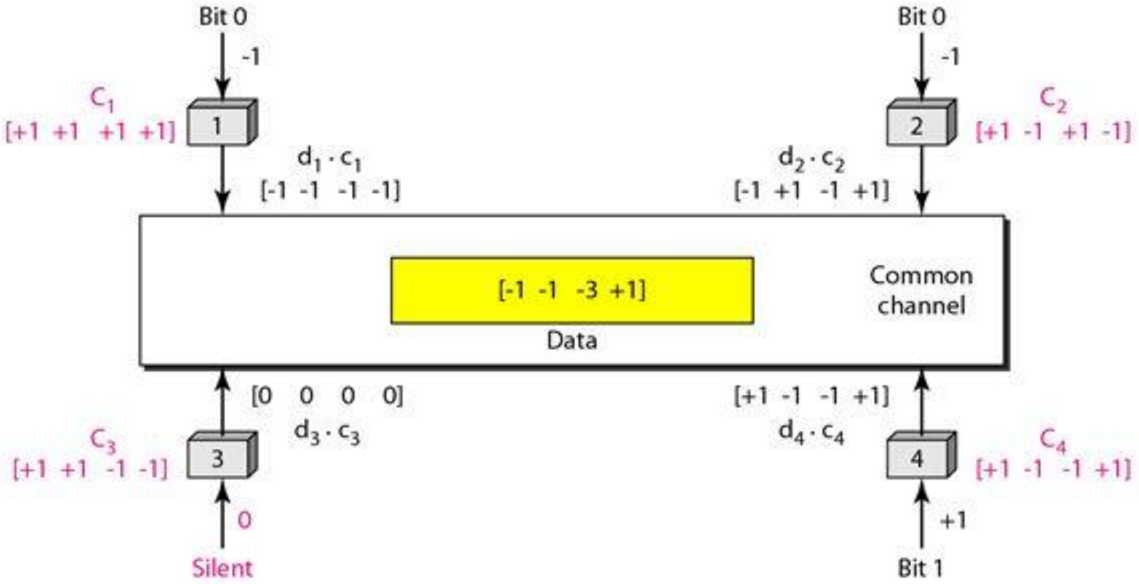
[+1+1-1-1]+[+1+1+1+1]=[+2+2 +0 +0]

Data Representation: (Encoding)

We follow the following rules for encoding: If a station needs to send a 0 bit, it encodes it as -1, if it needs to send a 1 bit, it encodes it as +1. When a station is idle, it sends no signal, which is interpreted as a 0.

Example,

- We show how four stations share the link during a 1-bit interval. The procedure can easily be repeated for additional intervals.
- We assume that stations 1 and 2 are sending a 0 bit and channel 4 is sending a 1 bit. 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 as defined before. The following figure shows the situation.



- Now imagine station 3, which we said is silent, is listening to 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 1