There are different types of Media Access Control methods in a LAN, they are mentioned below :

**Ethernet**– Ethernet is a 10Mbps LAN that uses the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol to control access network. When an endstation (network device) transmits data, every endstation on the LAN receives it. Each endstation checks the data packet to see whether the destination address matches its own address. If the addresses match, the endstation accepts and processes the packet. If they do not match, it disregards the packet. If two endstations transmit data simultaneously, a collision occurs and the result is a composite, garbled message. All endstations on the network, including the transmitting endstations, detect the collision and ignore the message. Each endstation that wants to transmit waits a random amount of time and then attempts to transmit again. This method is usually used for traditional Ethernet LAN.

**Token Ring**– This is a 4-Mbps or 16-Mbps token-passing method, operating in a ring topologyA token ring network is a local area network (LAN) topology where nodes/stations are arranged in a ring topology. Data passes sequentially between nodes on the network until it returns to the source station. To prevent congestion and collision, a token ring topology uses a token to ensure that only one node/station on the line is used at a time, thereby easily denoting media users of its activity.

* A token ring LAN is physically wired as a star topology but configured as a ring topology. A token continually circulates inside the toke ring LAN
* To transmit a message, a node inserts a message and destination address inside an empty token.
* The token is examined by each successive node.
* The destination node copies the message data and returns the token to the source with the source address and a data receipt message.
* The source receives the returned token, verifies copied and received data and empties the token.
* The empty token now changes to circulation mode, and the process continues.

**Fast Ethernet** – This is an extension of 10Mbps Ethernet standard and supports speed upto 100Mbps. The access method used is CSMA/CD .For physical connections Star wiring topology is used. Fast Ethernet is becoming very popular as an upgradation from 10Mbps Ethernet LAN to Fast Ethernet LAN is quite easy.

**FDDI (Fiber Distributed Data Interface)**– FDDI provides data speed at 100Mbps which is faster than Token Ring and Ethernet LANs . FDDI comprise two independent, counter-rotating rings : a primary ring and a secondary ring. Data flows in opposite directions on the rings. The counter-rotating ring architecture prevents data loss in the event of a link failure, a node failure, or the failure of both the primary and secondary links between any two nodes. This technology is usually implemented for a backbone network.

**Random Access Protocols**

n 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. This decision depends on the state of the medium (idle or busy).

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.

 The different random access methods are as follows:

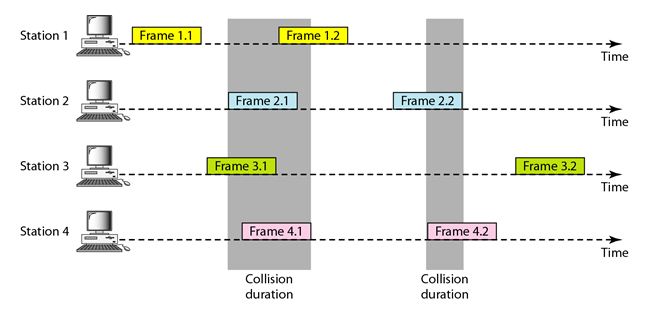
    ALOHA  
    CSMA  
    CSMA/CD  
    CSMA/CA

ALOHA, the earliest random access method was developed at the University of Hawaii in early 1970. It was designed for a radio (wireless) LAN, but it can be used on any shared medium.

It is obvious that there are potential collisions in this arrangement. 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**

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. The following figure shows an example of frame collisions in pure ALOHA.

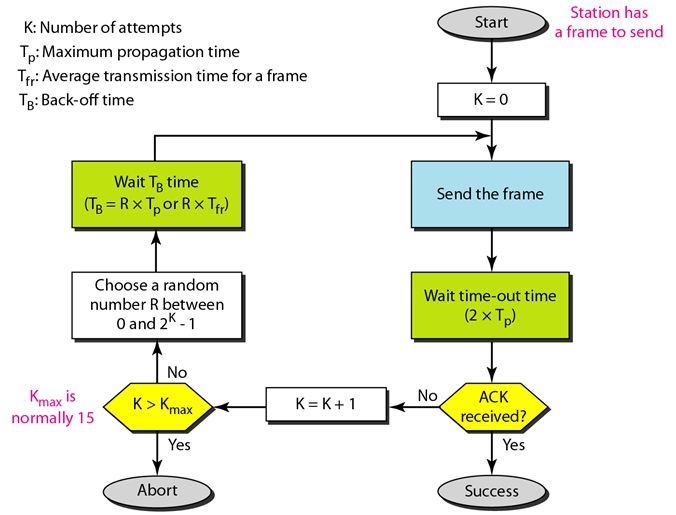


There are four stations (unrealistic assumption) that contend with one another for access to the shared channel. The figure shows that each station sends two frames; there are a total of eight frames on the shared medium. Some of these frames collide because multiple frames are in contention for the shared channel.

The above figure shows that only two frames survive: frame 1.1 from station 1 and frame 3.2 from station 3. We need to mention that 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.

It is obvious that we need to resend the frames that have been destroyed during transmission. The pure ALOHA protocol relies on acknowledgments from the receiver. 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.  
A collision involves two or more stations. If all these stations try to resend their frames after the time-out, the frames will collide again. Pure ALOHA dictates that when the time-out period passes, each station waits a random amount of time before resending its frame. The randomness will help avoid more collisions. We call this time the back-off time TB.

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 following figure shows the procedure for pure ALOHA based on the above strategy.

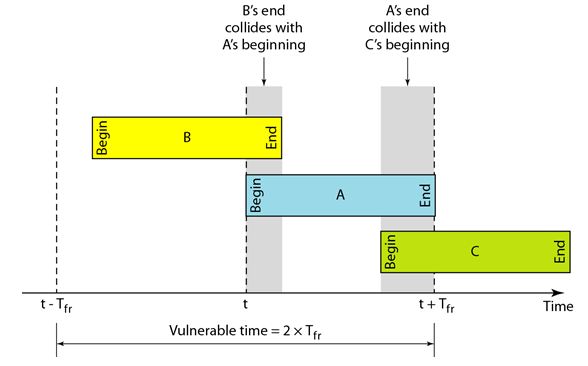


The time-out period is equal to the maximum possible round-trip propagation delay, which is twice the amount of time required to send a frame between the two most widely separated stations (2 x Tp) The back-off time TB is a random value that normally depends on K (the number of attempted unsuccessful transmissions). The formula for TB depends on the implementation. One common formula is the binary exponential back-off. 

In this method, for each retransmission, a multiplier in the range 0 to 2K - 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:**

The vulnerable time is in which there is a possibility of collision. We assume that the stations send fixed-length frames with each frame taking Tfr S to send. The following figure shows the vulnerable time for station A.



Station A sends a frame at time t. Now imagine station B has already sent a frame between t - Tfr 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 + Tfr . 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.

**Throughput:**

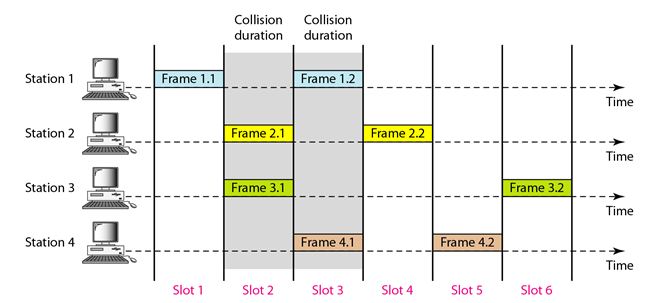
Let us call G the average number of frames generated by the system during one frame transmission time. Then it can be proved that the average number of successful transmissions for pure ALOHA is S = G x e-2G. The maximum throughput Smax is 0.184, for G = 1. In other words, if one-half a frame is generated during one frame transmission time (in other words, one frame during two frame transmission times), then 18.4 percent of these frames reach their destination successfully.

## 

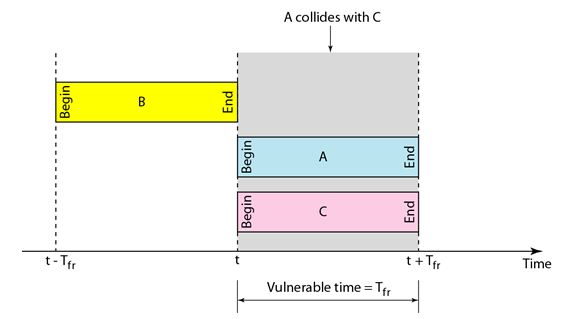
## Slotted ALOHA:

Pure ALOHA has a vulnerable time of 2 x Tfr. 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 Tfr s and force the station to send only at the beginning of the time slot. The following figure 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. But, still there is 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.  The following figure shows the situation.

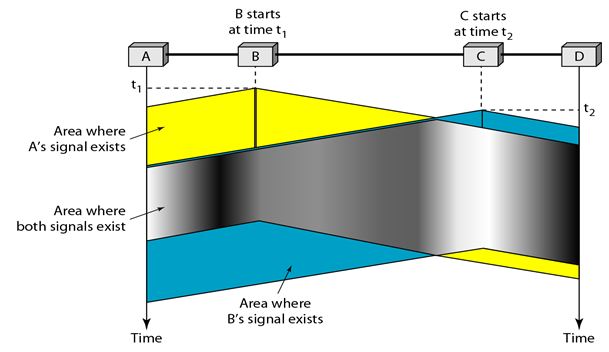


**Throughput:**

It can be proved that the average number of successful transmissions for slotted ALOHA is S = G x e-G. The maximum throughput Smax is 0.368, when G = 1. In other words, if a frame is generated during one frame transmission time, then 36.8 percent of these frames reach their destination successfully.

# Carrier Sense Multiple Access Protocol

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. CSMA can reduce the possibility of collision, but it cannot eliminate it. The following figure shows a space and time model of a CSMA network. Stations are connected to a shared channel (usually a dedicated medium).

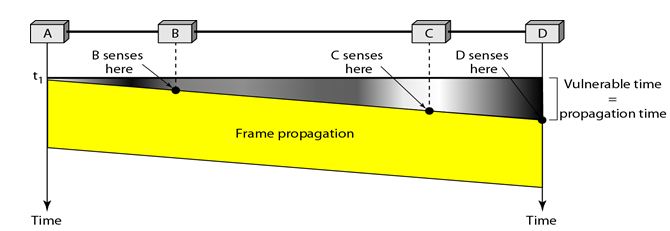


The possibility of collision still exists because of propagation delay, when a station sends a frame, it still takes time (although very short) for the first bit to reach every station and for every station to sense it. In other words, a station may sense the medium and find it idle, only because the 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.

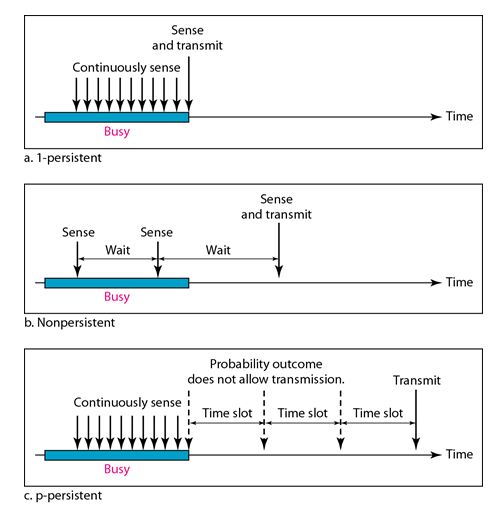
**Vulnerable Time:**

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 following figure shows the worst case. 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.



**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: the 1-persistent method, the nonpersistent method, and the p-persistent method. The following figure shows the behavior of three persistence methods when a station finds a channel busy.



• **1-Persistent:** The 1-persistent method is simple and straightforward. 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.

**• Nonpersistent:** In the 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. 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:** The p-persistent method is used if the channel has time slots with a 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.

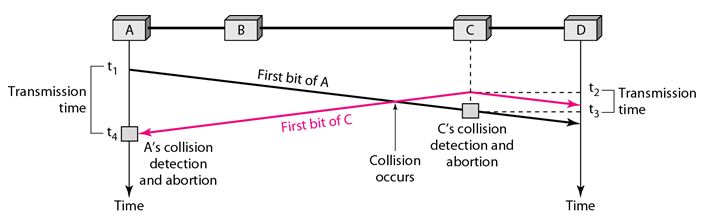
1. If the line is idle, it goes to step 1.

2. If the line is busy, it acts as though a collision has occurred and uses the back off procedure. 

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 the following 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. The protocol to work, the length of any frame divided by the bit rate in this protocol must be more than either of these durations. At time t4, the transmission of A’s frame, though incomplete, is aborted. At time t3, the transmission of C's frame, though incomplete, is aborted.

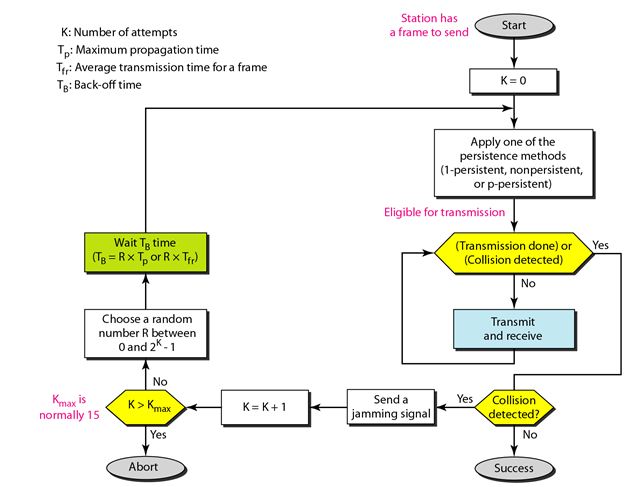
**Minimum Frame Size:**

For CSMA/CD 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 Tfr 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 .

**Procedure**

Now let us look at the flow diagram for CSMA/CD in the following figure.  It is similar to the one for the ALOHA protocol, but there are differences.



• The first difference is the addition of the persistence process. We need to sense the channel before we start sending the frame by using one of the persistence processes we discussed previously (nonpersistent, I-persistent, or p-persistent).

• The second difference is the frame transmission. In ALOHA, we first transmit the entire frame and then wait for an acknowledgment. In CSMA/CD, transmission and collision detection is a continuous process. We constantly monitor in order to detect one of two conditions: either transmission is finished or a collision is detected. Either event stops transmission.

**Energy Level:**

We can say that the level of energy in a channel can have three values: zero, normal, and abnormal. At the zero level, the channel is idle. At the normal level, a station has successfully captured the channel and is sending its frame. At the abnormal level, there is a collision and the level of the energy is twice the normal level. A station that has a frame to send or is sending a frame needs to monitor the energy level to determine if the channel is idle, busy, or in collision mode.

**Throughput**

The throughput of CSMA/CD is greater than that of pure or slotted ALOHA. The maximum throughput occurs at a different value of G and is based on the persistence method and the value of p in the p-persistent approach. For 1-persistent method the maximum throughput is around 50 percent when G =1. For nonpersistent method, the maximum throughput can go up to 90 percent when G is between 3 and 8.

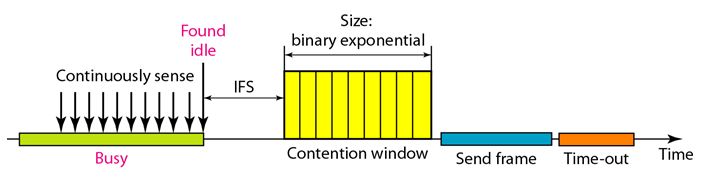
**Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)**

he basic idea behind CSMA/CD is that a station needs to be able to receive while transmitting to detect a collision. When there is no collision, the station receives one signal: its own signal. When there is a collision, the station receives two signals: its own signal and the signal transmitted by a second station. To distinguish between these two cases, the received signals in these two cases must be significantly different. In other words, the signal from the second station needs to add a significant amount of energy to the one created by the first station.

In a wired network, the received signal has almost the same energy as the sent signal because either the length of the cable is short or there are repeaters that amplify the energy between the sender and the receiver. This means that in a collision, the detected energy almost doubles.

 However, in a wireless network, much of the sent energy is lost in transmission. The received signal has very little energy. Therefore, a collision may add only 5 to 10 percent additional energy. This is not useful for effective collision detection.

Carrier sense multiple access with collision avoidance (CSMA/CA) was invented to avoid collisions on wireless networks. Collisions are avoided through the use of CSMAICA's three strategies: the interframe, space, the contention window, and acknowledgments, as shown in the following figure.



**Interframe 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 interframe 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 a shorter IFS has a higher priority.

**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**

The following figure shows the procedure. Note that the channel needs to be sensed before and after the IFS. The channel also needs to be sensed during the contention time. For each time slot of the contention window, the channel is sensed. If it is found idle, the timer continues; if the channel is found busy, the timer is stopped and continues after the timer becomes idle again.



**Controlled Access Protocols**

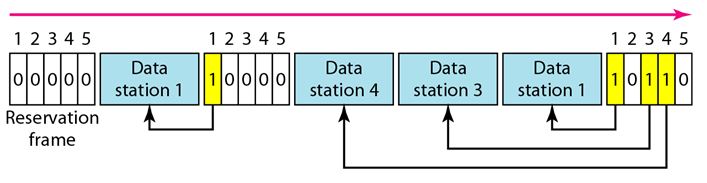
## Controlled access:

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. The three popular controlled-access methods are as follows.

**1. 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.

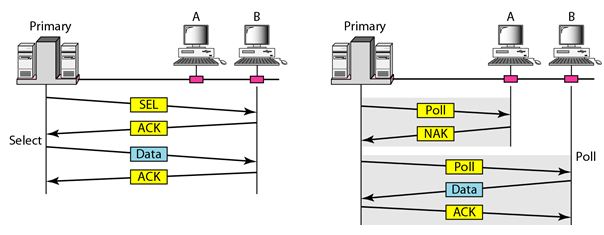
If there are N stations in the system, there are exactly N reservation minislots in the reservation frame. Each minislot belongs to a station. When a station needs to send a data frame, it makes a reservation in its own minislot. The stations that have made reservations can send their data frames after the reservation frame.  
The following figure shows a situation with five stations and a five-minislot 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 works with topologies in which one device is designated as a primary station and the other devices are secondary stations. 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. It is up to the primary device to determine which device is allowed to use the channel at a given time. The primary device, therefore, is always the initiator of a session. Consider the following figure.



If the primary wants to receive data, it asks the secondaries if they have anything to send, this is called poll function. If the primary wants to send data, it tells the secondary to get ready to receive; this is called select function.

**Select:**

The select function is used whenever the primary device has something to send. If it has something to send, the primary device sends it. It has to know whether the target device is prepared to receive or not. So 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 or with data (in the form of a data frame) if it does. 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.

**3. Token Passing:**

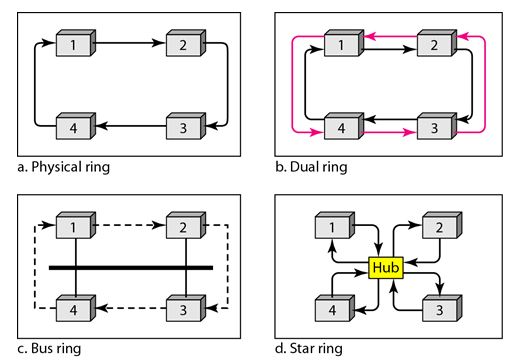
In the token-passing method, the stations in a network are organized in a logical ring. In other words, for each station, there is a predecessor and a successor. The predecessor is the station which is logically before the station in the ring; the successor is the station which is after the station in the ring. The current station is the one that is accessing the channel now. The right to this access has been passed from the predecessor to the current station. The right will be passed to the successor when the current station has no more data to send.

In this method, a special packet called a token circulates through the ring. The possession of the token gives the station the right to access the channel and send its data. When a station has some data to send, it waits until it receives the token from its predecessor. It then holds the token and sends its data. When the station has no more data to send, it releases the token, passing it to the next logical station in the ring. The station cannot send data until it receives the token again in the next round.

Token management is needed for this access method. Stations must be limited in the time they can have possession of the token. The token must be monitored to ensure it has not been lost or destroyed. For example, if a station that is holding the token fails, the token will disappear from the network. Another function of token management is to assign priorities to the stations and to the types of data being transmitted. And finally, token management is needed to make low- priority stations release the token to high priority stations.

## Logical Ring:

In a token-passing network, stations do not have to be physically connected in a ring; the ring can be a logical one. The following figure show four different physical topologies that can create a logical ring.



• In the physical ring topology, when a station sends the token to its successor, the token cannot be seen by other stations; the successor is the next one in line. This means that the token does not have to have the address of the next successor. The problem with this topology is that if one of the links-the medium between two adjacent stations fails, the whole system fails.

• The dual ring topology 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. After the failed link is restored, the auxiliary ring becomes idle again.

• In the bus ring topology, also called a token bus, the stations are connected to a single cable called a bus. They, however, make a logical ring, because each station knows the address of its successor (and also predecessor for token management purposes). When a station has finished sending its data, it releases the token and inserts the address of its successor in the token. Only the station with the address matching the destination address of the token gets the token to access the shared media. The Token Bus LAN, standardized by IEEE, uses this topology.

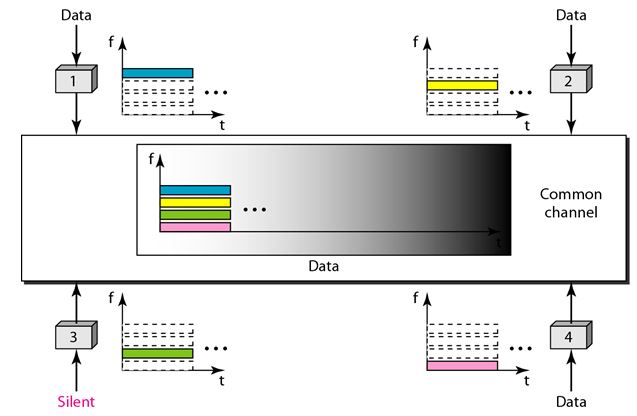
• In a star ring topology, the physical topology is a star. There is a hub, however, that acts as the connector. The wiring inside the hub makes the ring; the stations are connected to this ring through the two wire connections. This topology makes the network less prone to failure because if a link goes down, it will be bypassed by the hub and the rest of the stations can operate. Also adding and removing stations from the ring is easier. This topology is still used in the Token Ring LAN designed by IBM.

**Channelization Protocols**

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.

## ****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 bandpass 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.

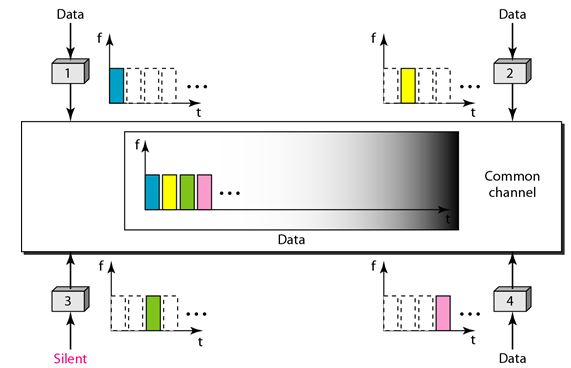


The differences between FDM and FDMA are as follows:

FDM, is a physical layer technique that combines the loads from low-bandwidth channels and transmits them by using a high-bandwidth channel. The channels that are combined are low-pass. The multiplexer modulates the signals, combines them, and creates a bandpass signal. The bandwidth of each channel is shifted by the multiplexer.

## 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. 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 at the beginning of each slot. 

**The differences between TDMA and TDM are :**

• TDM is a physical layer technique that combines the data from slower channels and transmits them by using a faster channel. The process uses a physical multiplexer that interleaves data units from each channel.

• TDMA, on the other hand, is an access method in the data link layer. The data link layer in each station tells its physical layer to use the allocated time slot. There is no physical multiplexer at the physical layer.

## 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.

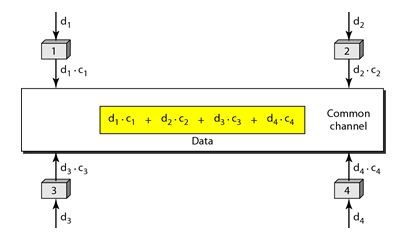
**Implementation:**

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.



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]

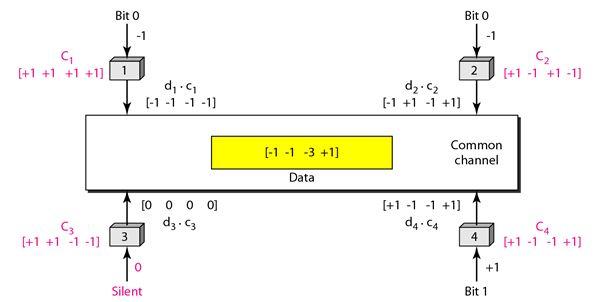
**Data Representation:**

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. 

**Encoding and Decoding:**

As a simple 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.

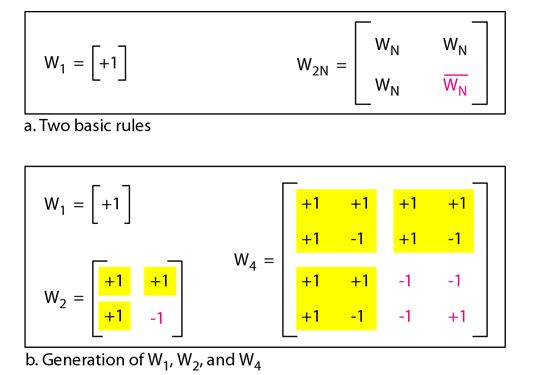
 

Now imagine station 3, which we said is silent, is listening to station 2. Station 3 multiplies the total data on the channel by the code for station 2, which is [+1 -1 +1-1],  
 to get 

[-1-1-3 +1]• [+1-1 +1-1] =-4/4 =-1 ---- --> bit 1

**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.



In the Walsh table, each row is a sequence of chips. W1 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 WN we can create the table for 2N sequences W2N, as shown in Figure. The WN with the overbar WN stands for the complement of WN  where each +1 is changed to -1 and vice versa. 

The above figure also shows how we can create W2 and W4 from W1.  After we select W1, W2 can be made from four W1 's, with the last one the complement of W1 After W2 is generated, W4 can be made of four W2's, with the last one the complement of W2. Of course, W8 is composed of four W4's, and so on. Note that after WN 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 = 2m.