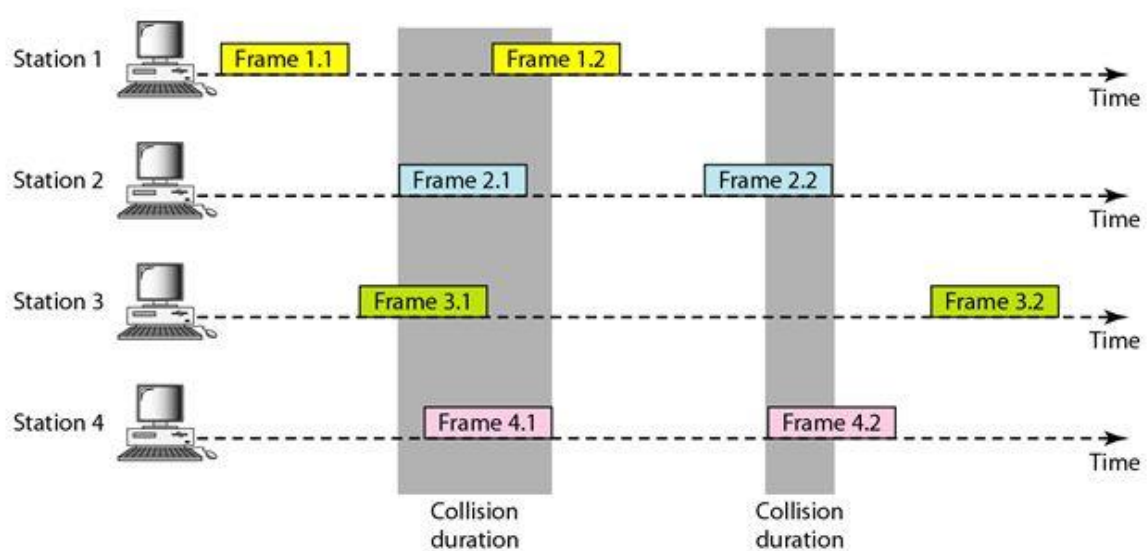


Random Access

ALOHA

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.



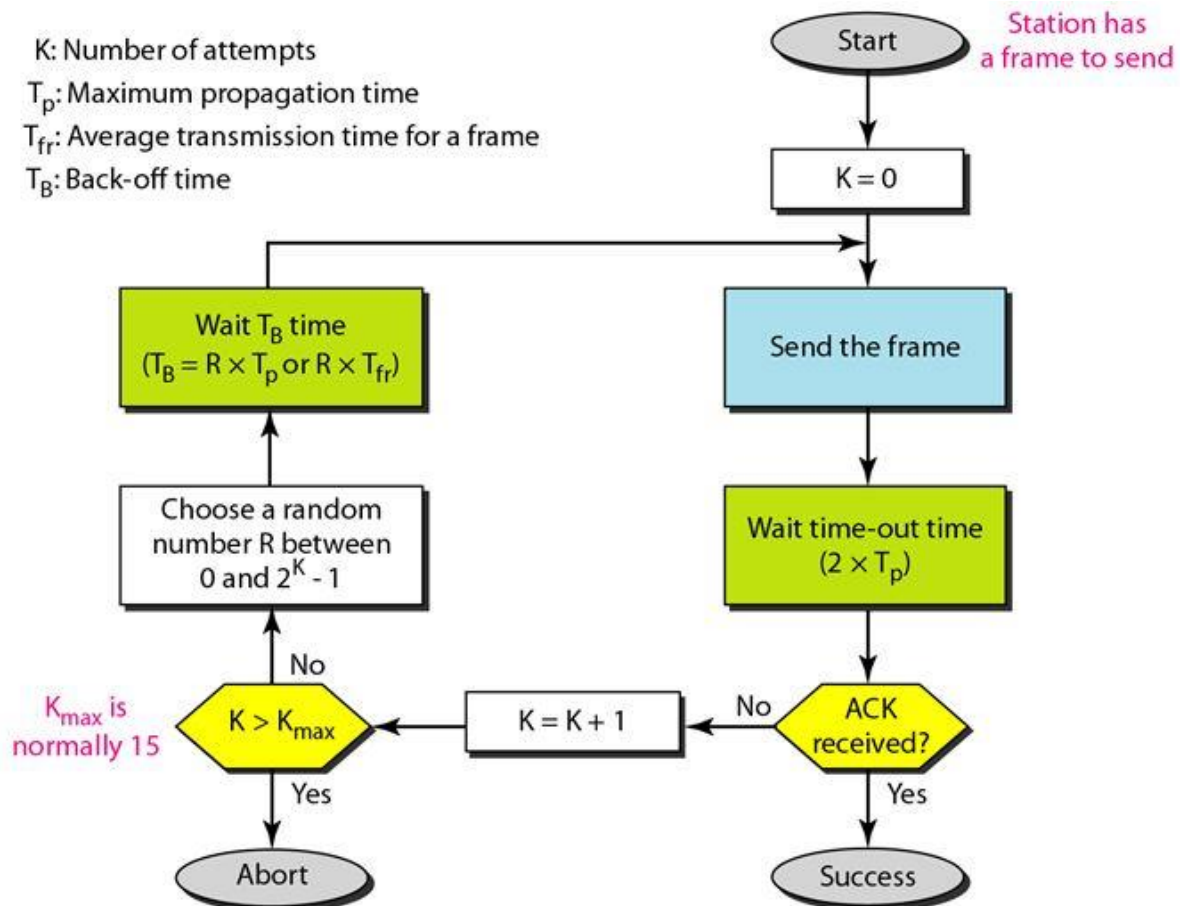
There are four stations that fight with one another for access to the shared channel. The figure shows that each station sends two frames; there are a total of eight frames on the shared medium. Some of these frames collide because multiple frames are in contention for the shared channel.

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 T_B .

Pure ALOHA has a second method to prevent congesting the channel with retransmitted frames. After a maximum number of retransmission attempts K_{max} a station must give up and try later. The following figure shows the procedure for pure ALOHA based on the above strategy.



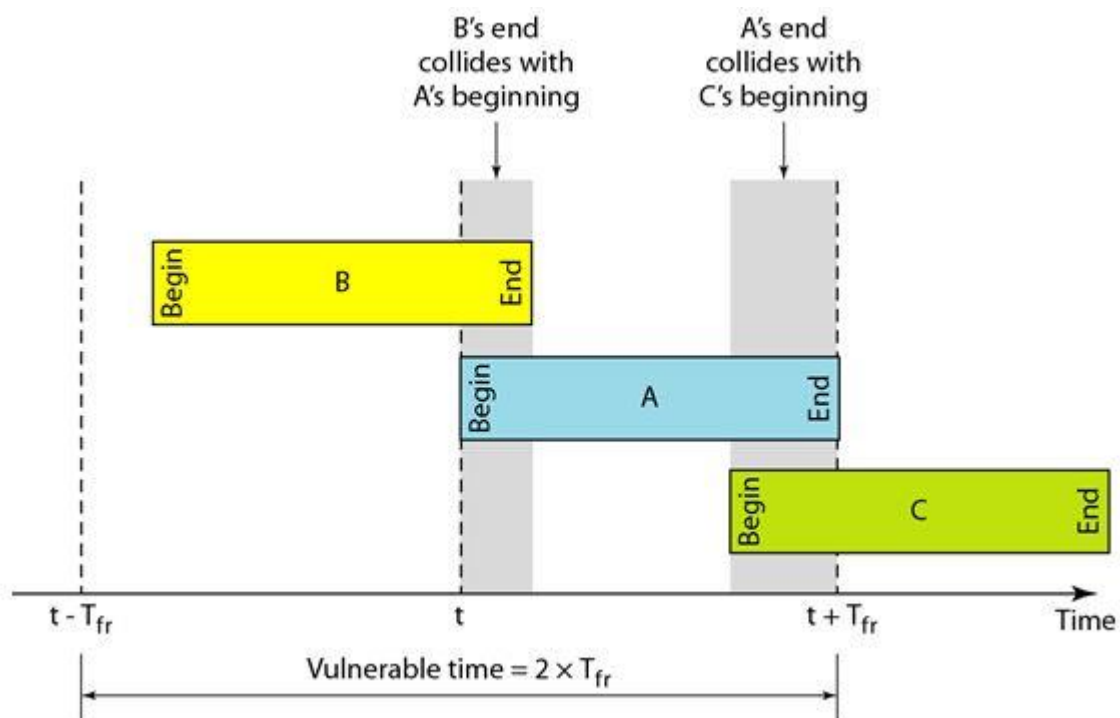
Explanation:

- A station which has a frame ready will send it.
- Then it waits for some time. ($2 \times T_p$)
- If it receives the acknowledgement then the transmission is successful.
- Otherwise the station uses a backoff strategy, and sends the packet again.
- After many times (K_{max}) if there is no acknowledgement then the station aborts the idea of transmission.

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 ($2 \times T_p$). The back-off time T_B is a random value that normally depends on K (the number of attempted unsuccessful transmissions). The formula for T_B depends on the implementation.

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 T_{fr} to send. The following figure shows the vulnerable time for station A.

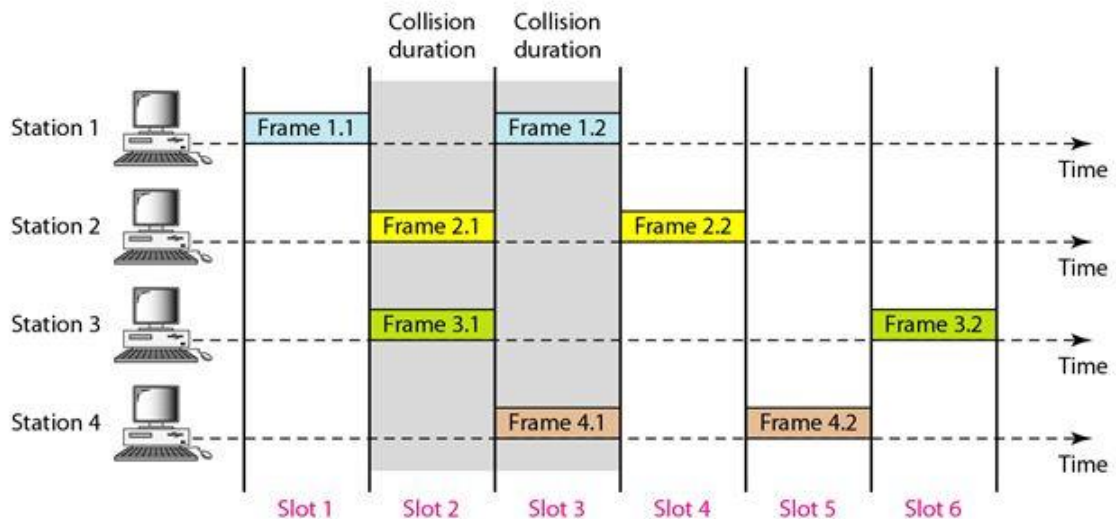


The end of B's frame collides with the beginning of A's frame and The beginning of C's frame collides with the end of A's frame.

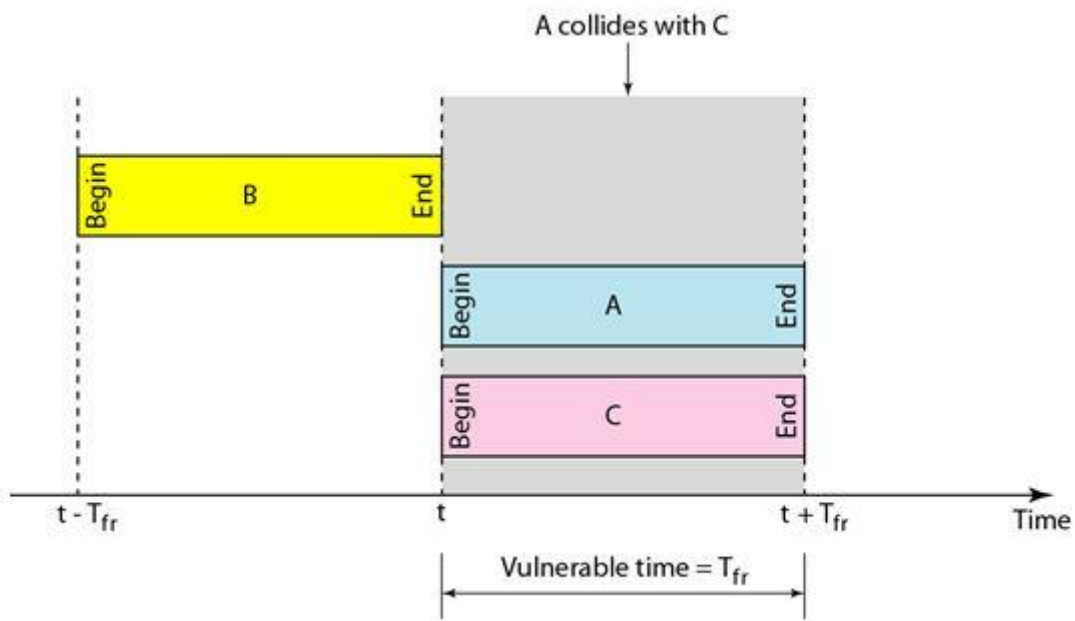
Slotted ALOHA:

Pure ALOHA has a vulnerable time of $2 \times T_{fr}$. This is so because there is no rule that defines when the station can send. 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. The following figure shows an example of frame collisions in slotted ALOHA.

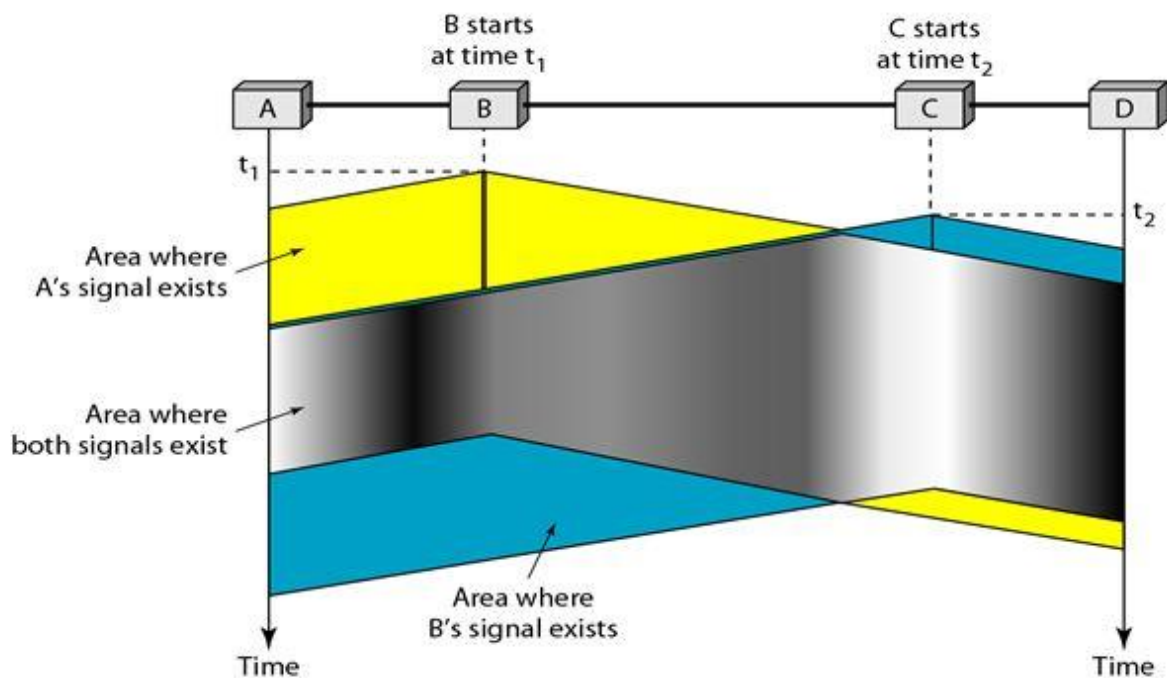


Because a station is allowed to send only at the beginning of the synchronized time slot, if a station misses this moment, it must wait until the beginning of the next time slot. This means that the station which started at the beginning of this slot has already finished sending its frame. But, still 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 T_{fr} . The following figure shows the situation.



Carrier Sense Multiple Access (CSMA) 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).

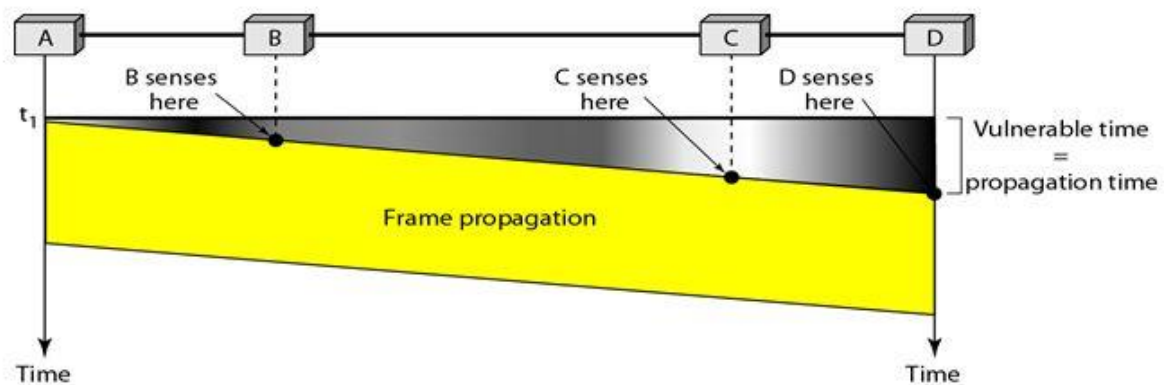


The possibility of collision still exists because of propagation delay, when a station sends a frame, it still takes time (although very short) for the first bit to reach every station and for every station to sense it. In other words, a station may sense the medium and find it idle, only because the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

At time t_1 station B senses the medium and finds it idle, so it sends a frame. At time t_2 ($t_2 > t_1$) 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.

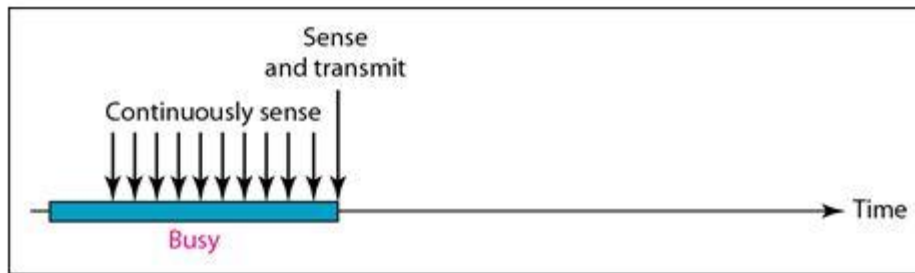
Vulnerable Time:

The vulnerable time for CSMA is the propagation time T_p . 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 t_1 which reaches the rightmost station D at time $t_1 + T_p$. The gray area shows the vulnerable area in time and space.

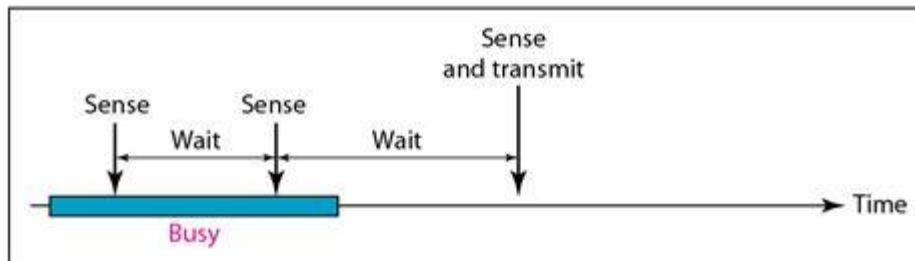


Persistence Methods:

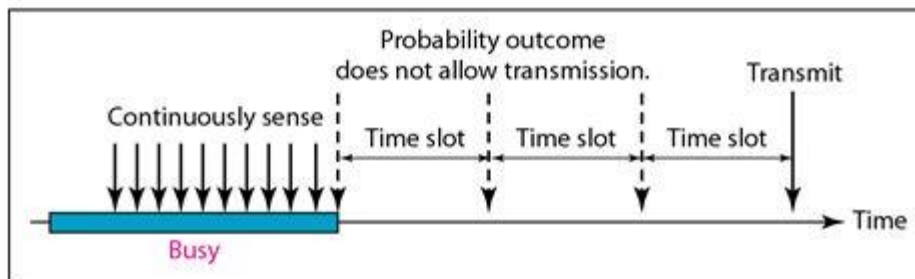
What should a station do if the channel is busy? What should a station do if the channel is idle? Three methods have been devised to answer these questions: the 1-persistent method, the nonpersistent method, and the p-persistent method. The following figure shows the behavior of three persistence methods when a station finds a channel busy.



a. 1-persistent



b. Nonpersistent



c. p-persistent

- **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:

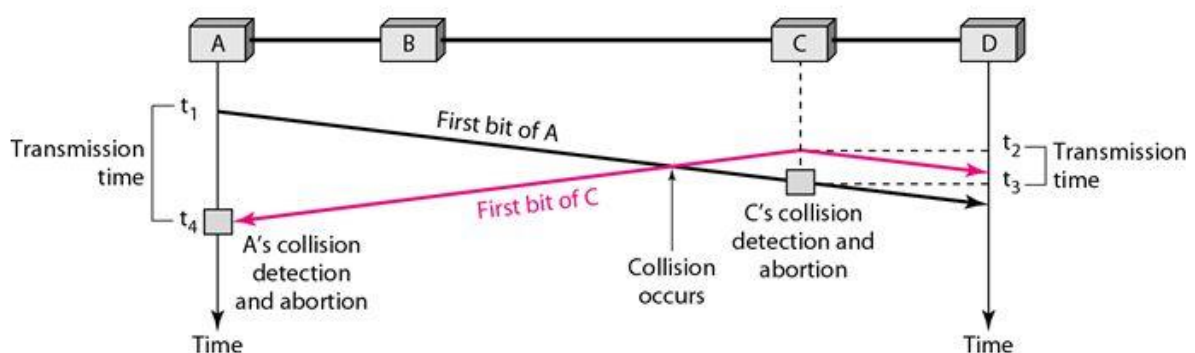
1. With probability p , the station sends its frame.
 2. With probability $q = 1 - p$, the station waits for the beginning of the next time slot and checks the line again.
-
1. If the line is idle, it goes to step 1.
 2. If the line is busy, it acts as though a collision has occurred and uses the back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

Carrier Sense Multiple Access with Collision Detection

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 t_2 , 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 t_2 . Station C detects a collision at time t_3 when it receives the first bit of A's frame. • Station A detects collision at time t_4 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 $t_4 - t_1$. C transmits for the duration $t_3 - t_2$. 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 t_4 , the transmission of A's frame, though incomplete, is aborted. At time t_3 , the transmission of C's frame, though incomplete, is aborted.

Minimum Frame Size:

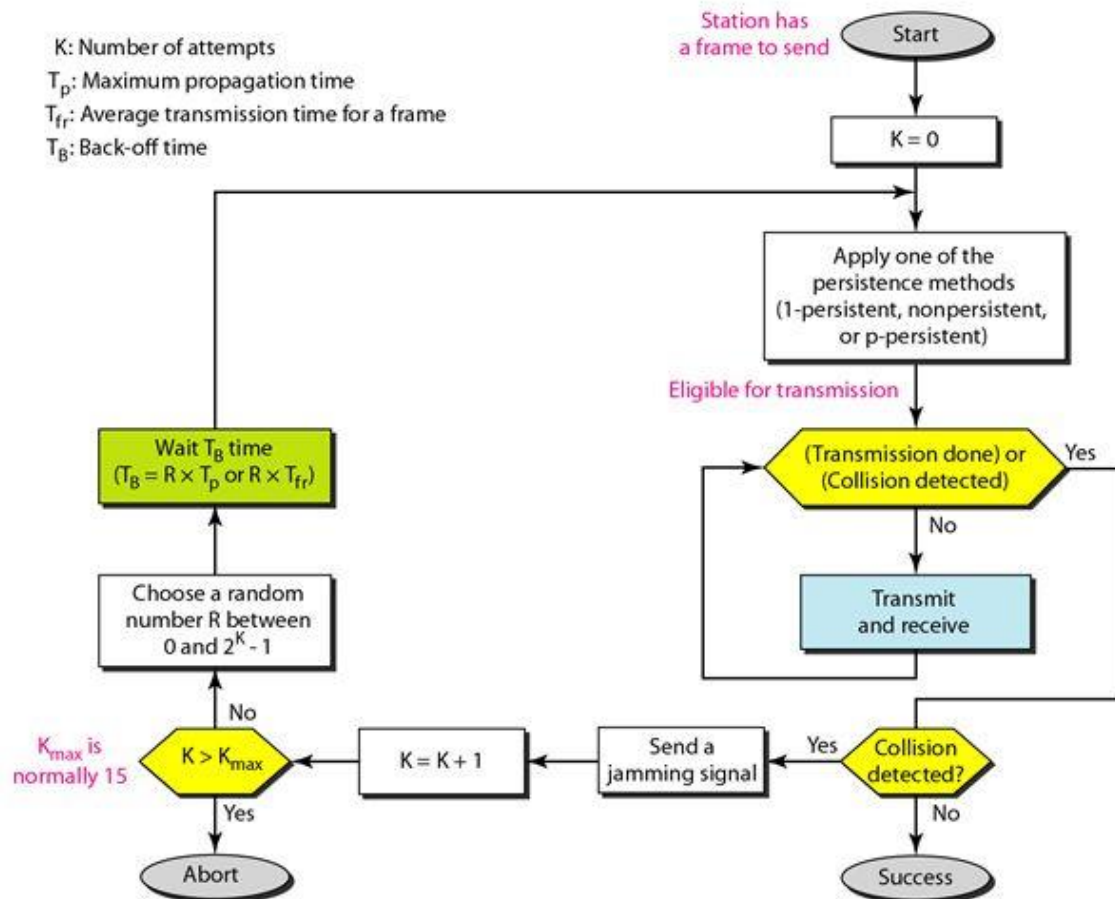
For CSMA/CD to work, we need a restriction on the frame size. Before sending the last bit of the frame, the sending station must detect a collision, if any, and abort the transmission.

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 T_{fr} must be at least two times the maximum propagation time T_p .

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 T_p to reach the second and the effect of the collision takes another time T_p to reach the first. So the requirement is that the first station must still be transmitting after $2T_p$.

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, 1-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.

Controlled Access Protocols

Controlled access:

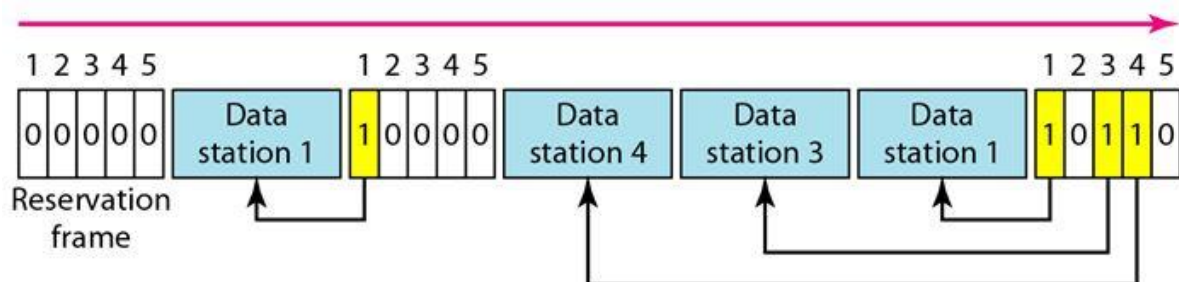
In controlled access, the stations consult one another to find which station has the right to send. A station cannot send unless it has been authorized by other stations. The three popular controlled-access methods are as follows.

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.

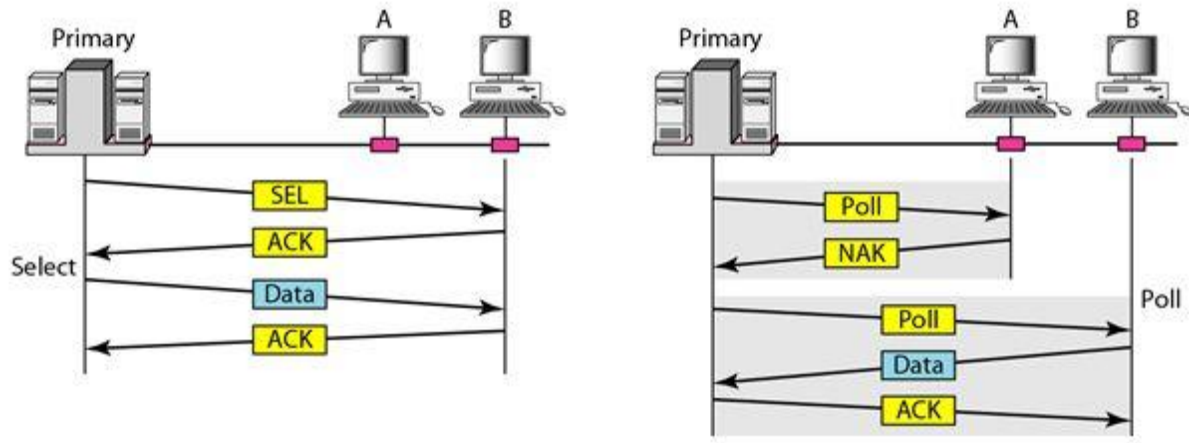
The following figure shows a situation with five stations and a five-minislot reservation frame. In the first interval, only stations 1, 3, and 4 have made reservations. In the second interval, only station 1 has made a reservation.



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:

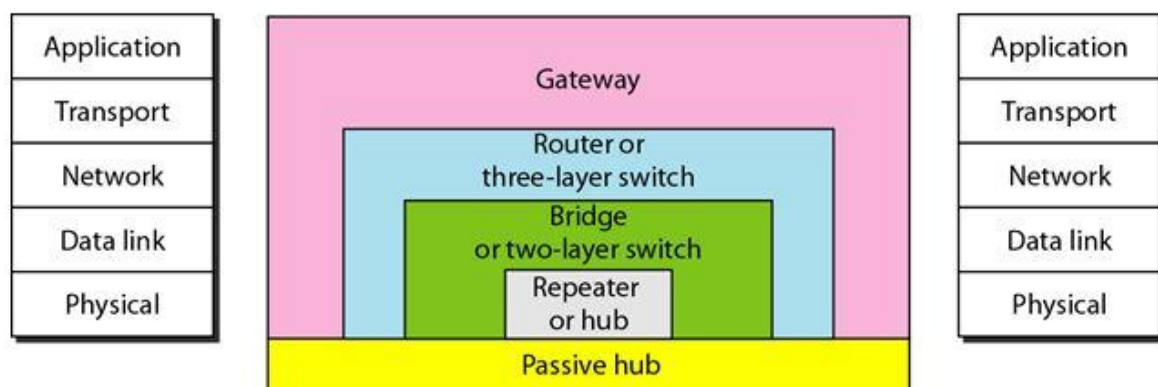
In the token-passing method, the stations in a network are organized in a logical ring. In other words, for each station, there is a predecessor and a successor. The predecessor is the station which is logically before the station in the ring; the successor is the station which is after the station in 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.

Different Connecting Devices

We divide connecting devices into five different categories based on the layer in which they operate in a network, as shown in the following figure.



he five categories contain devices which can be defined as:

- Those which operate below the physical layer such as a passive hub.
- Those which operate at the physical layer (a repeater or an active hub).

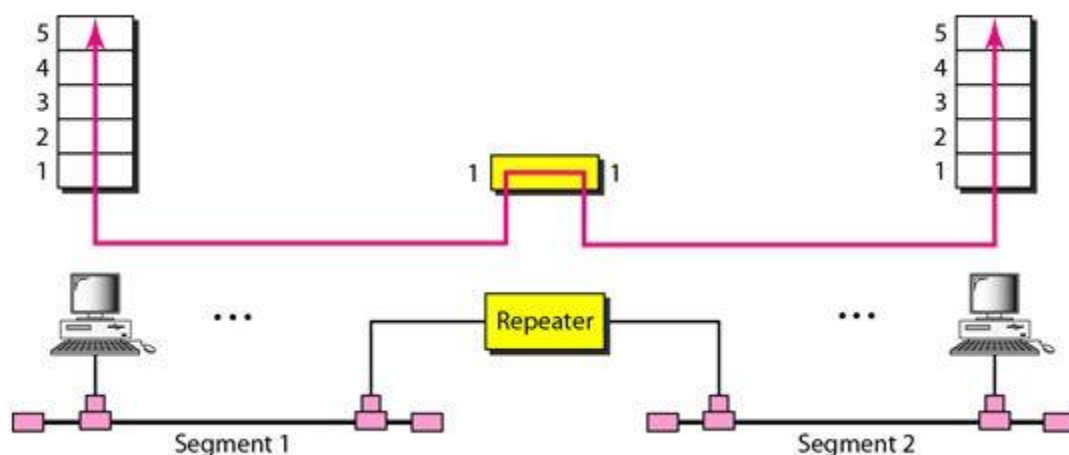
- Those which operate at the physical and data link layers (a bridge or a two-layer switch).
- Those which operate at the physical, data link, and network layers (a router or a three-layer switch).
- Those which can operate at all five layers (a gateway).

Passive Hubs:

A passive hub is just a connector. It connects the wires coming from different branches. In a star-topology Ethernet LAN, a passive hub is just a point where the signals coming from different stations collide; the hub is the collision point. Its location in the Internet model is below the physical layer.

Repeaters

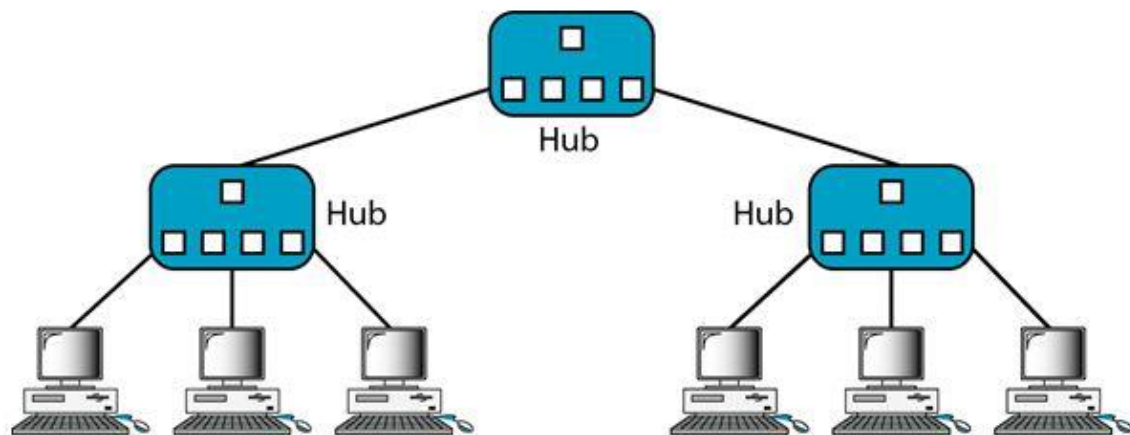
A repeater is a device that operates only in the physical layer. Signals that carry information within a network can travel a fixed distance before attenuation endangers the integrity of the data. A repeater receives a signal and, before it becomes too weak or corrupted, regenerates the original bit pattern. The repeater then sends the refreshed signal. A repeater can extend the physical length of a LAN, as shown in the following figure.



A repeater does not actually connect two LANs; it connects two segments of the same LAN. The segments connected are still part of one single LAN. A repeater is not a device that can connect two LANs of different protocols.

Active Hubs:

An active hub is actually a multipart repeater. It is normally used to create connections between stations in a physical star topology. However, hubs can also be used to create multiple levels of hierarchy, as shown in Figure



The difference in the functionality of a bridge and a repeater is, a bridge has filtering capability. It can check the destination address of a frame and decide if the frame should be forwarded or dropped. If the frame is to be forwarded, the decision must specify the port. A bridge has a table that maps addresses to ports.

Two-Layer Switches:

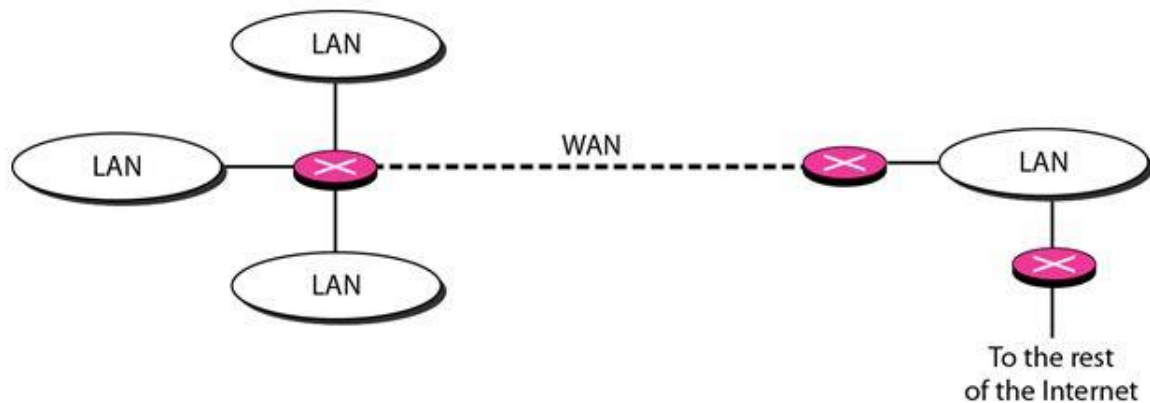
A switch can be a two-layer switch or a three-layer switch. A three-layer switch is used at the network layer; it is a kind of router. The two-layer switch performs at the physical and data link layers.

A two-layer switch is a bridge, a bridge with many ports and a design that allows better (faster) performance. A bridge with a few ports can connect a few LANs together. A bridge with many ports may be able to allocate a unique port to each station, with each station on its own independent entity. This means no competing traffic (no collision, as we saw in Ethernet).

A two-layer switch, as a bridge does, makes a filtering decision based on the MAC address of the frame it received. However, a two-layer switch can be more sophisticated. It can have a buffer to hold the frames for processing. It can have a switching factor that forwards the frames faster. Some new two-layer switches, called cut-through switches, have been designed to forward the frame as soon as they check the MAC addresses in the header of the frame.

Routers

A router is a three-layer device that routes packets based on their logical addresses (host-to-host addressing). A router normally connects LANs and WANs in the Internet and has a routing table that is used for making decisions about the route. The routing tables are normally dynamic and are updated using routing protocols.



Three-Layer Switches:

A three-layer switch is a router, but a faster and more sophisticated. The switching fabric in a three-layer switch allows faster table lookup and forwarding.

Gateway:

Although some textbooks use the terms gateway and router interchangeably, most of the literature distinguishes between the two. A gateway is normally a computer that operates in all five layers of the Internet or seven layers of OSI model. A gateway takes an application message, reads it, and interprets it. This means that it can be used as a connecting device between two internetworks that use different models. For example, a network designed to use the OSI model can be connected to another network using the Internet model. The gateway connecting the two systems can take a frame as it arrives from the first system, move it up to the OSI application layer, and remove the message. Gateways can provide security.