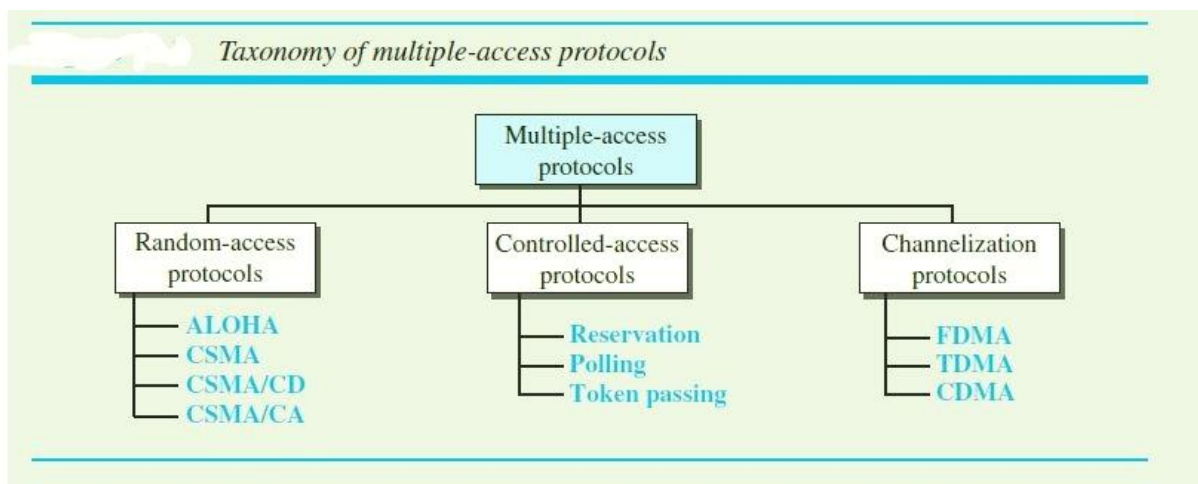


MODULE 2

Media Access Control (MAC)

- 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.
- Many protocols have been devised to handle access to a shared link. All of these protocols belong to a sub layer in the data-link layer called media access control (MAC).



RANDOM ACCESS

- In **random-access** or **contention** methods, no station is superior to another station and none is assigned control over another.
- 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). In other words, each station can transmit when it desires on the condition that it follows the predefined procedure, including testing the state of the medium.
- 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**.

- In a random-access method, each station has the right to the medium without being controlled by any other station. However, if more than one station tries to send, there is an access conflict “**collision**” and the frames will be either destroyed or modified.
- The random-access methods have evolved from a very interesting protocol known as ALOHA, which used a very simple procedure called **multiple access (MA)**.
- The method was improved with the addition of a procedure that forces the station to sense the medium before transmitting. This was called **carrier sense multiple access (CSMA)**.
- CSMA method later evolved into two parallel methods: **carrier sense multiple access with collision detection (CSMA/CD)**, which tells the station what to do when a collision is detected, and **carrier sense multiple access with collision avoidance (CSMA/CA)**, which tries to avoid the collision.

ALOHA

- 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.
- 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 (multiple access) there is only one channel to share, there is the possibility of collision between frames from different stations.

Figure 1 below shows an example of frame collisions in pure ALOHA

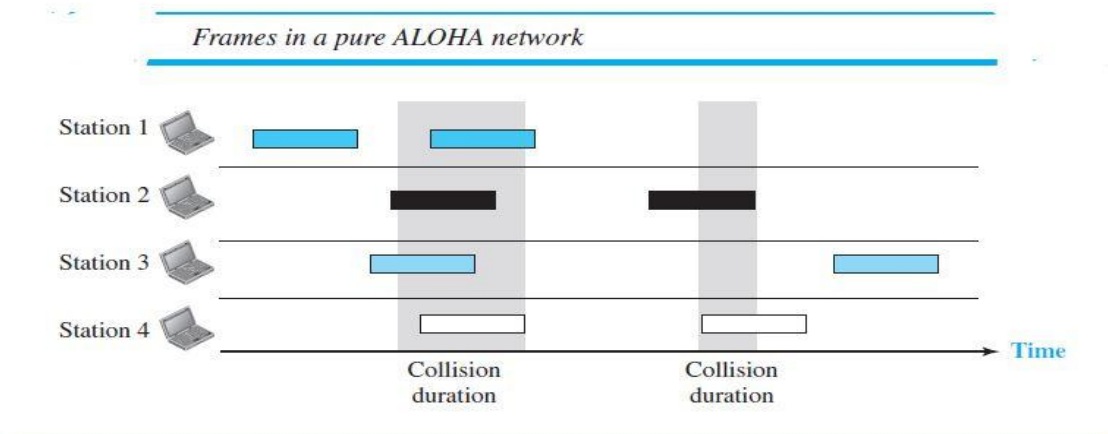


Figure 1: Frames in pure ALOHA network

- There are four stations (unrealistic assumption) that contend with one another for access to the shared channel. The above 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. Figure 1 shows that only two frames survive: one frame from station 1 and one frame from station 3.
- 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 the frames have to be resend that have been destroyed during transmission.
- The pure ALOHA protocol relies on acknowledgments from the receiver. When a station sends a frame, it expects the receiver to send an acknowledgment. If the acknowledgment does not arrive after a time-out period, the station assumes that the frame (or the acknowledgment) has been destroyed and resends the frame.
- 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. This time is called as 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. Figure 1 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 ($2 \times T_p$).
- The backoff time T_B is a random value that normally depends on K (the number of attempted unsuccessful transmissions).
- In this method, for each retransmission, a multiplier $R = 0$ to 2^K is randomly chosen and multiplied by T_p (maximum propagation time) or T_{fr} (the average time required to send out a frame) to find T_B .

Note: The range of the random numbers increases after each collision. The value of K_{max} is usually chosen as 15.

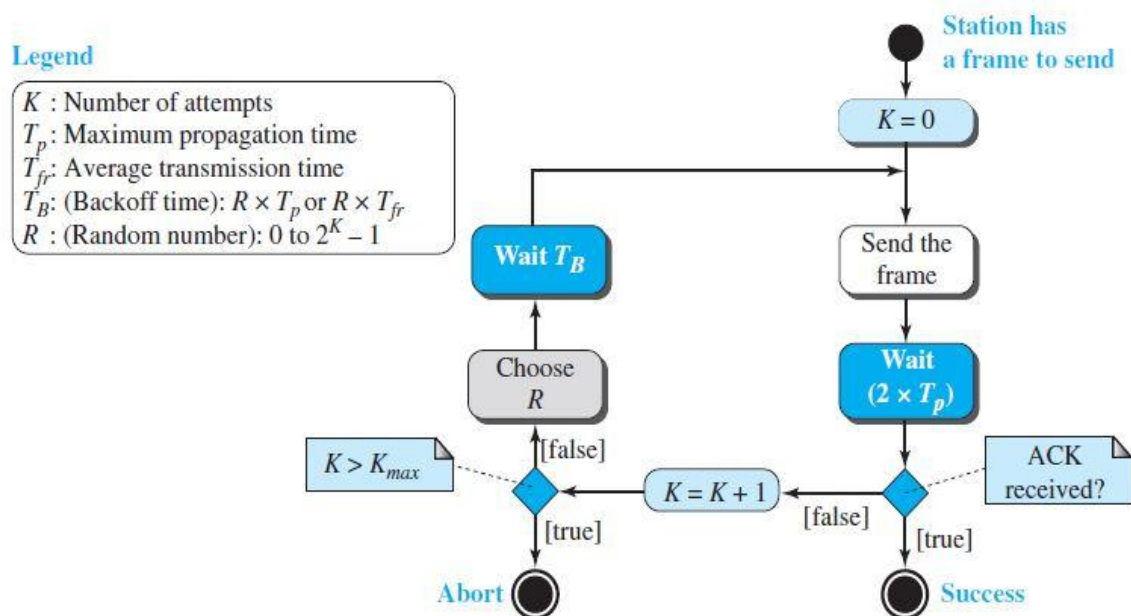


Figure 3: Procedure for pure ALOHA protocol

PROBLEM 1

The stations on a wireless ALOHA network are a maximum of 600 km apart. If we assume that signals propagate at 3×10^8 m/s, find T_p . Assume $K = 2$, Find the range of R .

Solution: $T_p = (600 \times 10^3) / (3 \times 10^8) = 2$ ms.

The range of R is, $R = (0 \text{ to } 2^k) = \{0, 1, 2, 3\}$.

This means that T_B can be 0, 2, 4, or 6 ms, based on the outcome of the random variable R .

Vulnerable time

The length of time in which there is a possibility of collision. The stations send fixed-length frames with each frame taking T_{fr} seconds to send. Figure 4 shows the vulnerable time for station B.

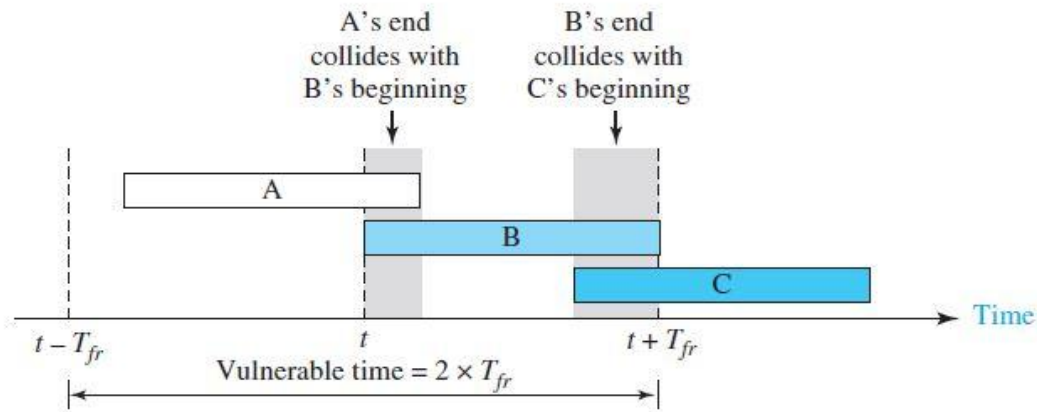


Figure 4: Vulnerable time for pure ALOHA protocol

Station B starts to send a frame at time t . Imagine station A has started to send its frame after $t - T_{fr}$. This leads to a collision between the frames from station B and station A. On the other hand, suppose that station C starts to send a frame before time $t + T_{fr}$. There is also a collision between frames from station B and station C.

From the Figure 4, it can be seen that the vulnerable time during which a collision may occur in pure ALOHA is 2 times the frame transmission time.

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

PROBLEM 2

A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the requirement to make this frame collision-free?

Solution: Average frame transmission time T_{fr} is 200 bits/200 kbps or 1 ms.

The vulnerable time is $2 \times 1 \text{ ms} = 2 \text{ ms}$.

This means no station should send later than 1 ms before this station starts transmission and no station should start sending during the period (1 ms) that this station is sending.

Throughput

G = the average number of frames generated by the system during one frame transmission time (T_{fr})

S = the average number of successfully transmitted frames for pure ALOHA.

And is given by, $S = G \times e^{-2G}$. -----(1)

Differentiate equation (1) with respect to G and equate it to 0, we get $G = 1/2$.

Substitute $G=1/2$ in equation (1) to get S_{max} .

The maximum throughput $S_{max} = 0.184$.

- If one-half a frame is generated during one frame transmission time (one frame during two frame transmission times), then 18.4 percent of these frames reach their destination successfully.
- G is set to $G = 1/2$ to produce the maximum throughput because the vulnerable time is 2 times the frame transmission time. Therefore, if a station generates only one frame in this vulnerable time (and no other stations generate a frame during this time), the frame will reach its destination successfully.

NOTE: The throughput for pure ALOHA is $S = G \times e^{-2G}$.

The maximum throughput $S_{max} = 1/(2e) = 0.184$ when $G = (1/2)$.

PROBLEM 3

A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the throughput if the system (all stations together) produces

- 1000 frames per second?
- 500 frames per second?
- 250 frames per second?

Solution: The frame transmission time T_{fr} is 200/200 kbps or 1 ms.

(a) If the system creates 1000 frames per second, or 1 frame per millisecond ($1s = 1000ms$)

then $G = 1$ (because G = number of frames generated for one T_{fr}).

$S = G \times e^{-2G} = 0.135$ (13.5 percent). This means that the throughput is

$1000 \times 0.135 = 135$ frames. Only 135 frames out of 1000 will probably survive.

(b) If the system creates 500 frames per second, or 1/2 frame per millisecond ($1s = 1000ms$)

then $G = 1/2$ (because G = number of frames generated for one T_{fr}).

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

$500 \times 0.184 = 92$ frames. Only 92 frames out of 500 will probably survive.

- (c) If the system creates 250 frames per second, or 1/4 frame per millisecond ($1\text{ s} = 1000\text{ms}$) then $G = 1/4$ (because $G =$ number of frames generated for one T_{fr}).

$S = G \times e^{-2G} = 0.152$ (15.2 percent). This means that the throughput is $250 \times 0.152 = 38$ frames. Only 38 frames out of 250 will probably survive.

Slotted ALOHA

- Pure ALOHA has a vulnerable time of $2 \times T_{fr}$. This is so because there is no rule that defines when the station can send.
- A station may send soon after another station has started or just 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} seconds and force the station to send only at the beginning of the time slot. Figure 5 shows an example of frame collisions in slotted ALOHA.

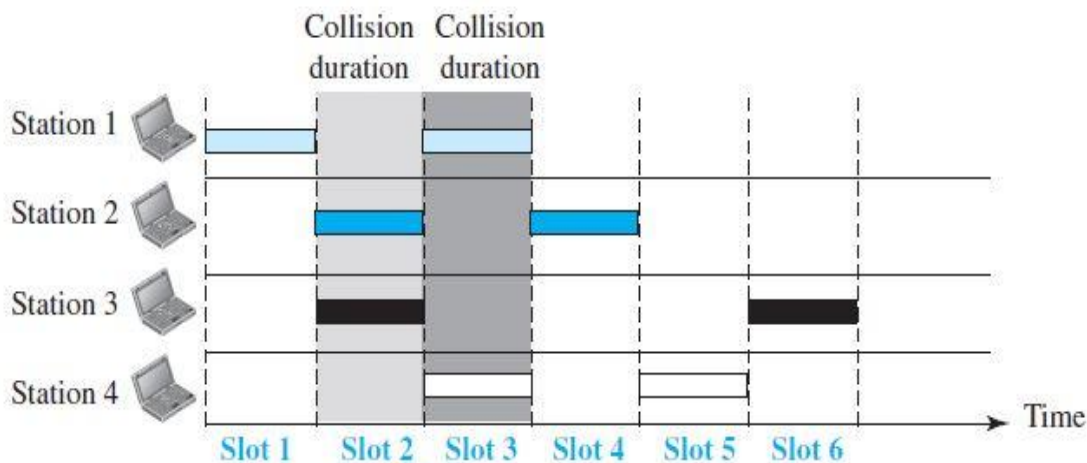


Figure 5: Frames in Slotted ALOHA network

- 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.
- 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 T_{fr} . Figure 6 shows the situation.

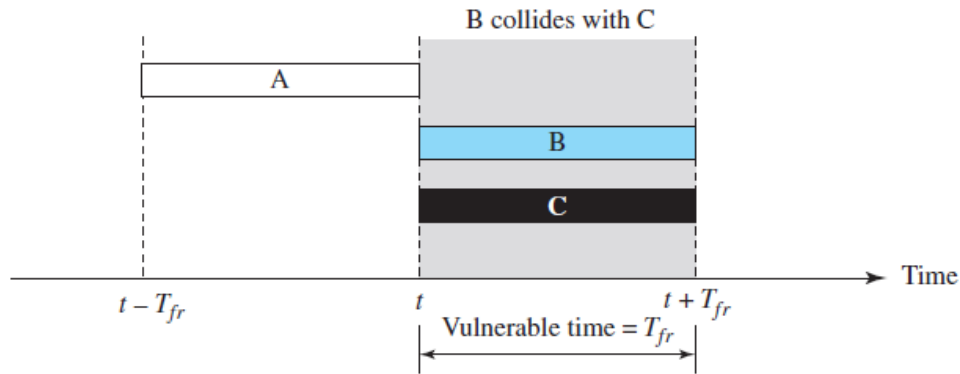


Figure 6: Vulnerable time for slotted ALOHA protocol

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

Throughput

G = the average number of frames generated by the system during one frame transmission time (T_{fr})

S = the average number of successfully transmitted frames for Slotted ALOHA.

And is given by, $S = G \times e^{-G}$. -----(1)

Differentiate equation (1) with respect to G and equate it to 0, we get $G = 1$.

Substitute $G=1$ in equation (1) to get S_{max} .

The maximum throughput $S_{max} = 0.368$.

- If one frame is generated during one frame transmission time then 36.8 percent of these frames reach their destination successfully.
- G is set to $G = 1$ to produce the maximum throughput because the vulnerable time is equal to the frame transmission time. Therefore, if a station generates only one frame in this vulnerable time (and no other stations generate a frame during this time), the frame will reach its destination successfully.

NOTE: The throughput for Slotted ALOHA is $S = G \times e^{-G}$.

The maximum throughput $S_{max} = 1/(e) = 0.368$ when $G = 1$.

PROBLEM 3

A Slotted ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the throughput if the system (all stations together) produces

- 1000 frames per second?

b. 500 frames per second?

c. 250 frames per second?

Solution: The frame transmission time T_{fr} is 200/200 kbps or 1 ms.

(a) If the system creates 1000 frames per second, or 1 frame per millisecond (1s = 1000ms)

then $G = 1$ (because $G =$ number of frames generated for one T_{fr}).

$S = G \times e^{-G} = 0.368$ (36.8 percent). This means that the throughput is

$1000 \times 0.368 = 368$ frames. Only 368 frames out of 1000 will probably survive.

(b) If the system creates 500 frames per second, or 1/2 frame per millisecond (1s = 1000ms)

then $G = 1/2$ (because $G =$ number of frames generated for one T_{fr}).

$S = G \times e^{-G} = 0.303$ (30.3 percent). This means that the throughput is

$500 \times 0.303 = 151$ frames. Only 151 frames out of 500 will probably survive.

(c) If the system creates 250 frames per second, or 1/4 frame per millisecond (1s = 1000ms)

then $G = 1/4$ (because $G =$ number of frames generated for one T_{fr}).

$S = G \times e^{-G} = 0.195$ (19.5 percent). This means that the throughput is

$250 \times 0.195 = 49$ frames. Only 49 frames out of 250 will probably survive.

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.
- 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 Figure 7, a space and time model of a CSMA network. Stations are connected to a shared channel.
- The possibility of collision still exists because of propagation delay; when a station sends a frame, it still takes time (although very short) for the first bit to reach every station and for every station to sense it.
- A station may sense the medium and find it idle, only because the first bit sent by another station has not yet been received.

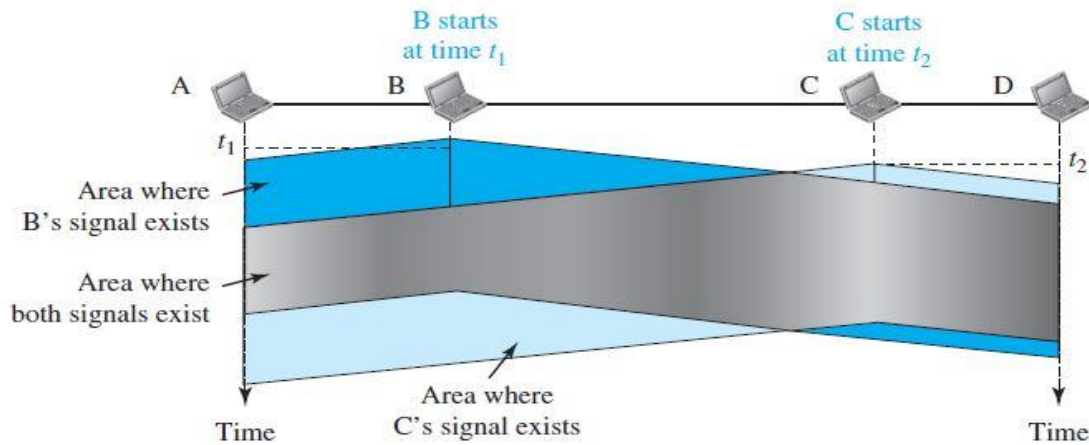


Figure 7: Space/time model of a collision in CSMA

- 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.
- Figure 8 below 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.

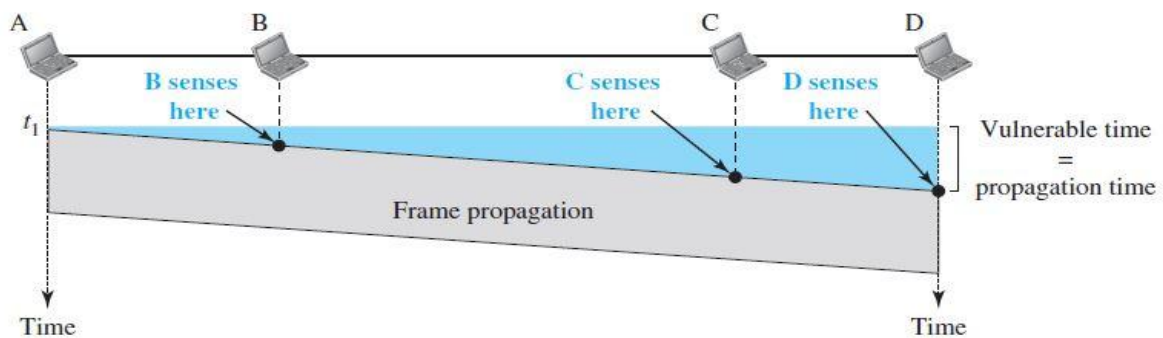


Figure 8: Vulnerable time in CSMA

Persistence Methods

Persistence method is developed to determine what the station has to do whenever it encounters the channel is idle or busy. There are 3 persistent methods

1. 1-persistent method
2. Non persistent method, and
3. p-Persistent method.

Figure 9 shows the behaviour of three persistence methods when a station finds a channel busy.

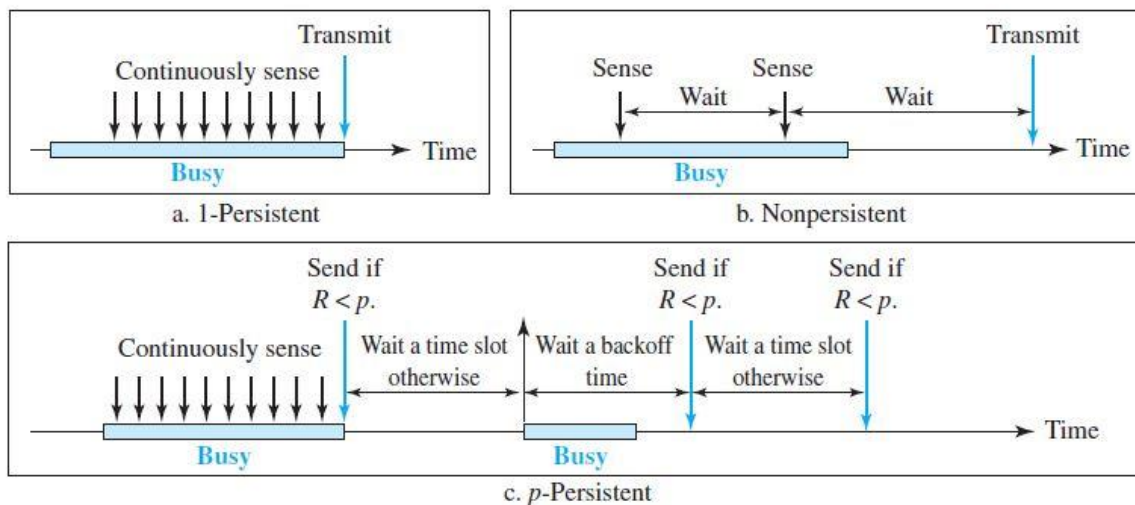


Figure 9: Behaviour of three persistence methods

1-Persistent

- The 1-persistent method is simple and straightforward.
- 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

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

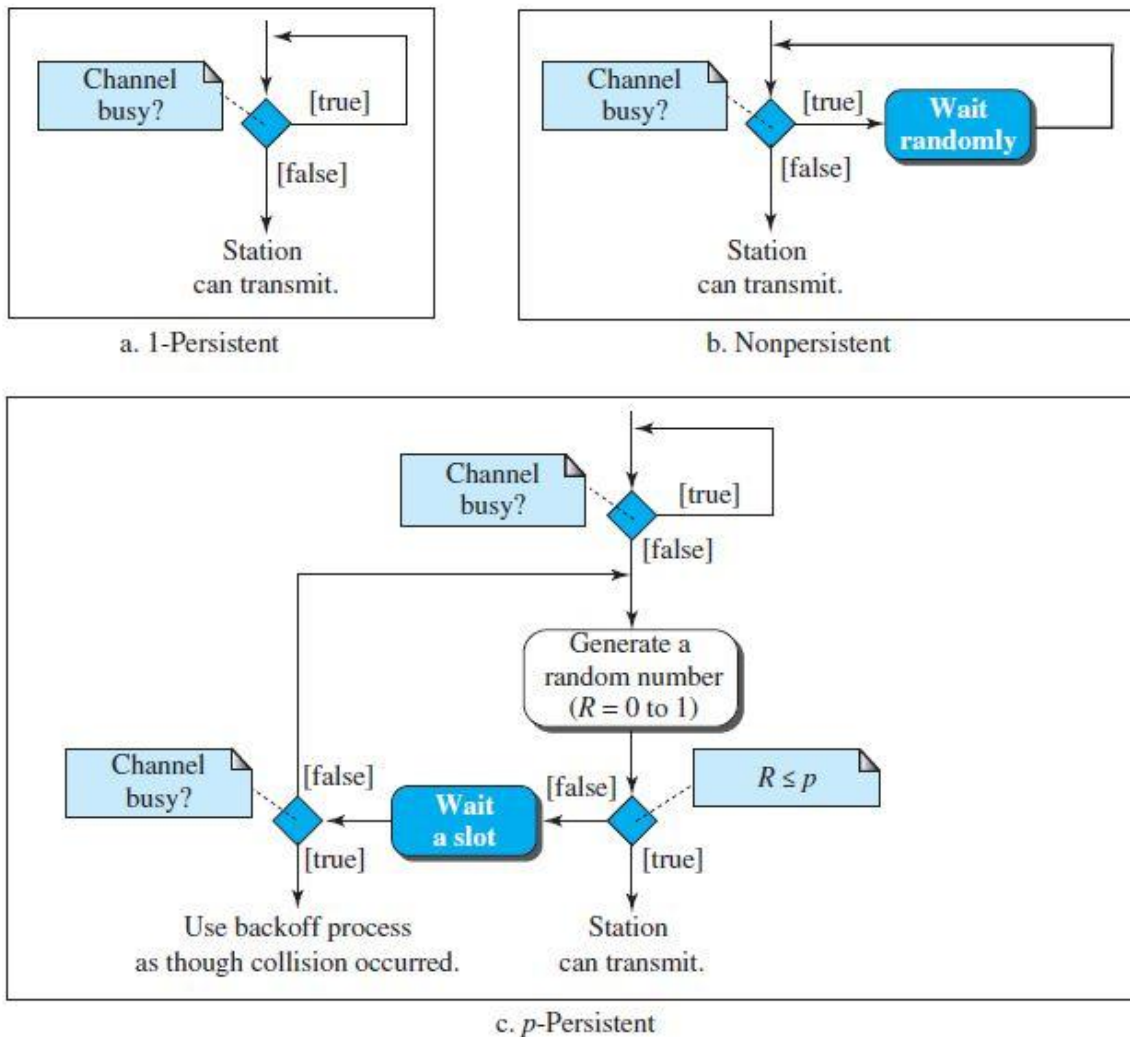


Figure 10: Flow diagram for three persistence methods

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

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.
- Station monitors the medium after it sends a frame to see if the transmission was successful.
- 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

In Figure 11, stations A and C are involved in the collision.

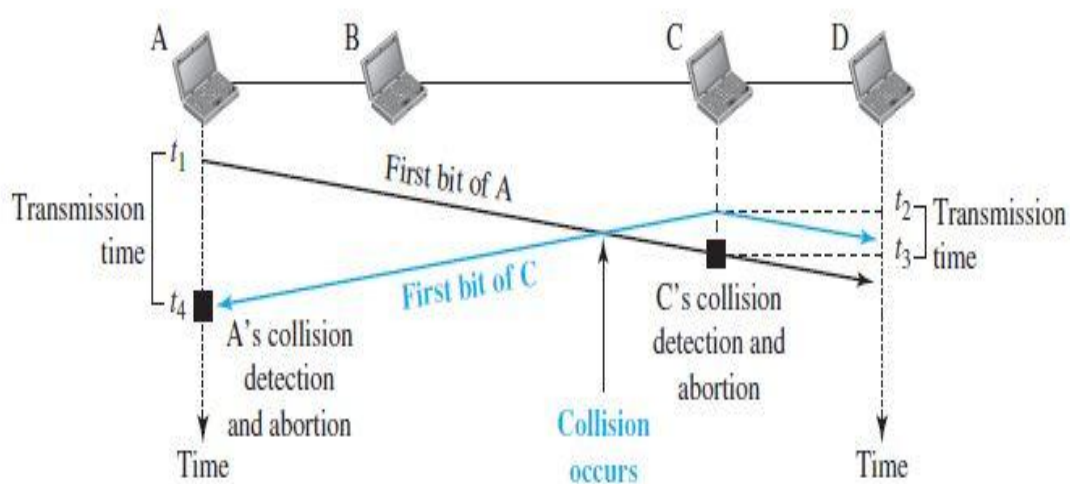


Figure 11: Collision of the first bits in CSMA/CD

- 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 C immediately aborts transmission.
- Station A detects collision at time t_4 when it receives the first bit of C's frame, it also immediately aborts transmission.

From the Figure 11, A transmits for the duration $t_4 - t_1$, C transmits for the duration $t_3 - t_2$.

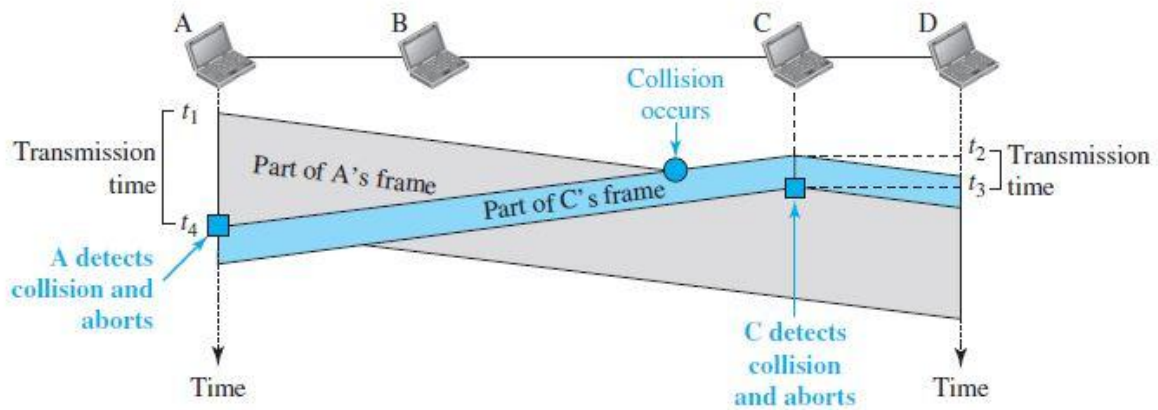


Figure 12: Collision and abortion in CSMA/CD

Minimum Frame Size

- Before sending the last bit of the frame, the sending station must detect a collision, if any, and abort the transmission.
- Once the entire frame is sent, station 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 .
- 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 $2T_p$.

PROBLEM:

A network using CSMA/CD has a bandwidth of 10 Mbps. If the maximum propagation time (including the delays in the devices and ignoring the time needed to send a jamming signal) is $25.6 \mu s$, what is the minimum size of the frame?

Solution:

The minimum frame transmission time is $T_{fr} = 2 \times T_p = 51.2 \mu s$. This means, in the worst case, a station needs to transmit for a period of $51.2 \mu s$ to detect the collision.

The minimum size of the frame is, Band width $\times T_{fr} = 10 \text{ Mbps} \times 51.2 \mu s = 512 \text{ bits}$ or 64 bytes. This is actually the minimum size of the frame for Standard Ethernet.

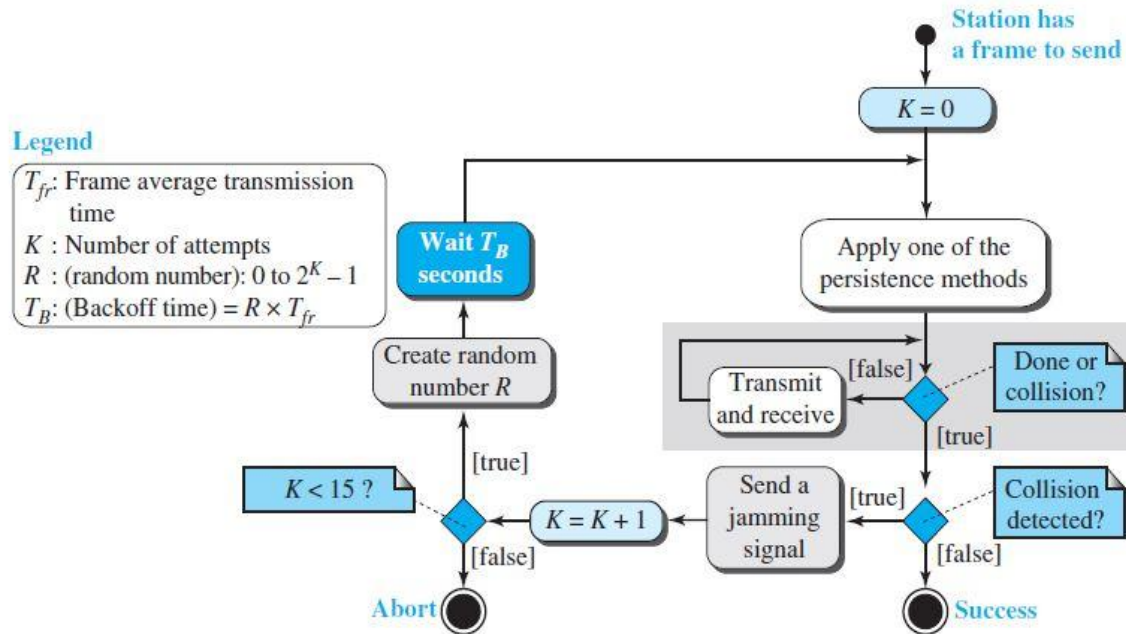


Figure 13: Flow diagram for the CSMA/CD

The flow diagram for CSMA/CD is as shown in Figure 13. It is similar to the one for the ALOHA protocol, but there are differences.

1. The first difference is the addition of the persistence process. It is required to sense the channel before sending the frame by using one of the persistence processes (non persistent, 1 persistent, or p-persistent).
2. The second difference is the frame transmission. In ALOHA, there is transmission of the entire frame and then wait for an acknowledgment. In CSMA/CD, transmission and collision detection are continuous processes.
 - It is not like the entire frame is sent and then look for a collision. The station transmits and receives continuously and simultaneously (using two different ports or a bidirectional port).
 - Loop is used to show that transmission is a continuous process. It is constantly monitored in order to detect one of two conditions: either transmission is finished or a collision is detected.
 - Either event stops transmission. When it comes out of the loop, if a collision has not been detected, it means that transmission is complete; the entire frame is transmitted. Otherwise, a collision has occurred.
3. The third difference is the sending of a short **jamming signal** to make sure that all other stations become aware of the collision.

Energy Level

The level of energy in a channel can have three values:

- 1) Zero level : The channel is idle
- 2) Normal level: A station has successfully captured the channel and is sending its frame.
- 3) Abnormal level: There is a collision and the level of the energy is twice the normal level.

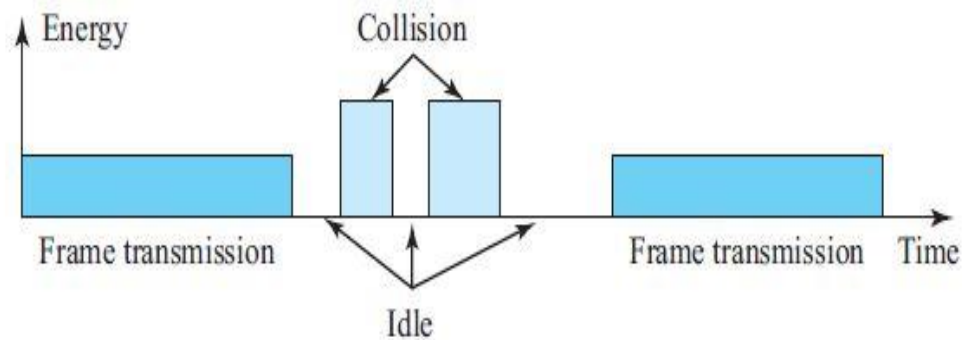


Figure 14: Energy level during transmission, idleness, or collision

NOTE: 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 the 1-persistent method, the maximum throughput is around 50 percent when $G = 1$. For the non persistent method, the maximum throughput can go up to 90 percent when G is between 3 and 8.

CSMA/CA

- **Carrier sense multiple access with collision avoidance (CSMA/CA)** was invented for wireless networks.
- Collisions are avoided through the use of CSMA/CA's three strategies: the inter frame space, the contention window, and acknowledgments.

Inter frame Space (IFS):

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

Refer Figure 16 of Contention window

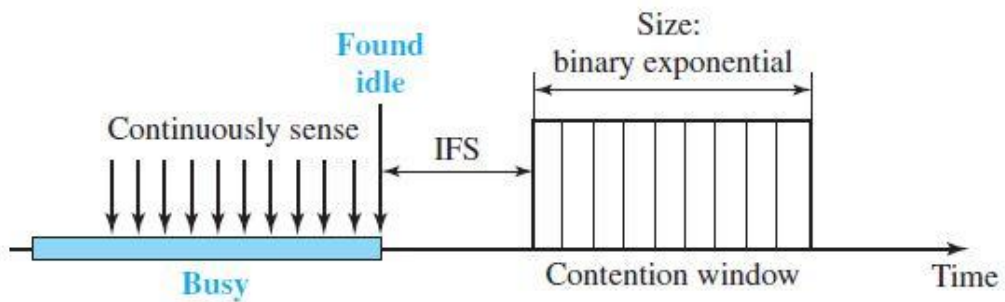


Figure 16: Contention window

Acknowledgement

Even with all the precautions considered, 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.

Frame Exchange Time Line

Figure 17 shows the exchange of data and control frames in time.

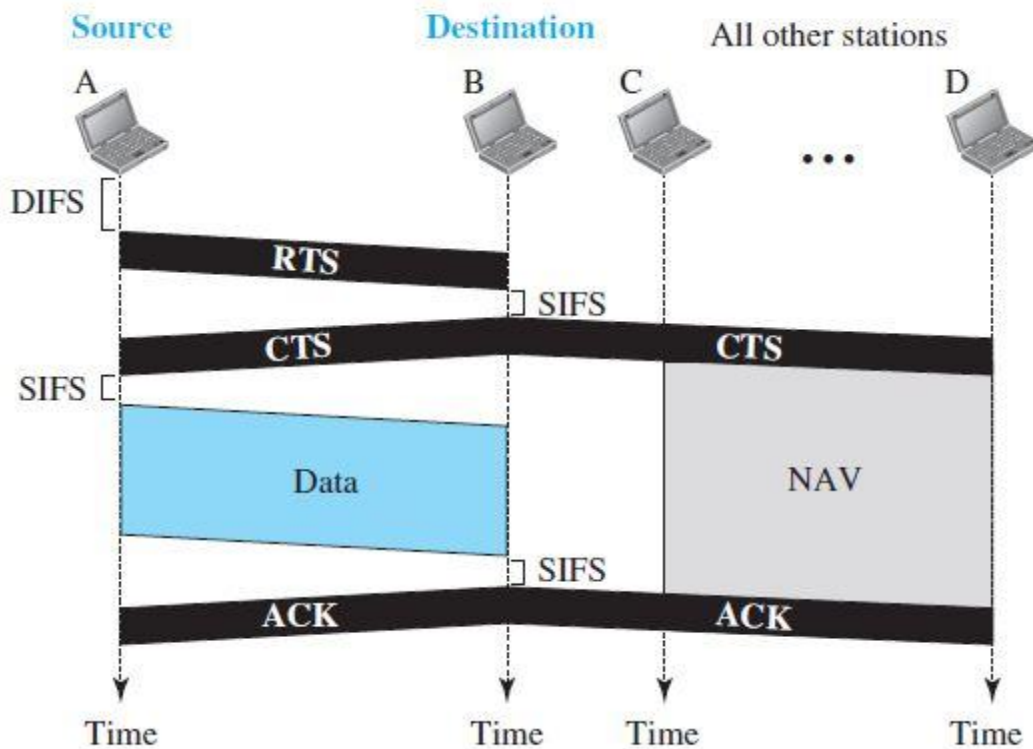


Figure 17: CSMA/CA and NAV

1. Before sending a frame, the source station senses the medium by checking the energy level at the carrier frequency.
 - a. The channel uses a persistence strategy with back off until the channel is idle.
 - b. After the station is found to be idle, the station waits for a period of time called the DCF inter frame space (DIFS), then the station sends a control frame called the request to send (RTS).
2. After receiving the RTS and waiting a period of time called the short inter frame space (SIFS), the destination station sends a control frame, called the clear to send (CTS), to the source station. This control frame indicates that the destination station is ready to receive data.
3. The source station sends data after waiting an amount of time equal to SIFS.
4. The destination station, after waiting an amount of time equal to SIFS, sends an acknowledgment to show that the frame has been received. Acknowledgment is needed in this protocol because the station does not have any means to check for the successful arrival of its data at the destination. On the other hand, the lack of collision in CSMA/CD is a kind of indication to the source that data have arrived.

NOTE: DIFS=DCF Inter frame space or Distributed Coordination Function Inter frame Space time.

Network Allocation Vector

- When a station sends an RTS frame, it includes the duration of time that it needs to occupy the channel.
- The stations that are affected by this transmission create a timer called a **Network Allocation Vector (NAV)** that shows how much time must pass before these stations are allowed to check the channel for idleness.
- Each time a station accesses the system and sends an RTS frame, other stations start their NAV. In other words, each station, before sensing the physical medium to see if it is idle, first checks its NAV to see if it has expired. Figure 17 shows the idea of NAV.

Collision during Handshaking

- If there is a collision during the time when RTS or CTS control frames are in transition, often called the handshaking period.

- Two or more stations may try to send RTS frames at the same time. These control frames may collide. However, because there is no mechanism for collision detection, the sender assumes there has been a collision if it has not received a CTS frame from the receiver. The back off strategy is employed, and the sender tries again.

Hidden-Station Problem

- The solution to the hidden station problem is the use of the handshake frames (RTS and CTS). Figure 17 shows that the RTS message from A reaches B, but not C.
- Both A and C are within the range of B, the CTS message, which contains the duration of data transmission from B to A, reaches C.
- Station C knows that some hidden station is using the channel and refrains from transmitting until that duration is over.

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.

There are three controlled access methods,

1. Reservation.
2. Polling.
3. Token passing.

1. Reservation

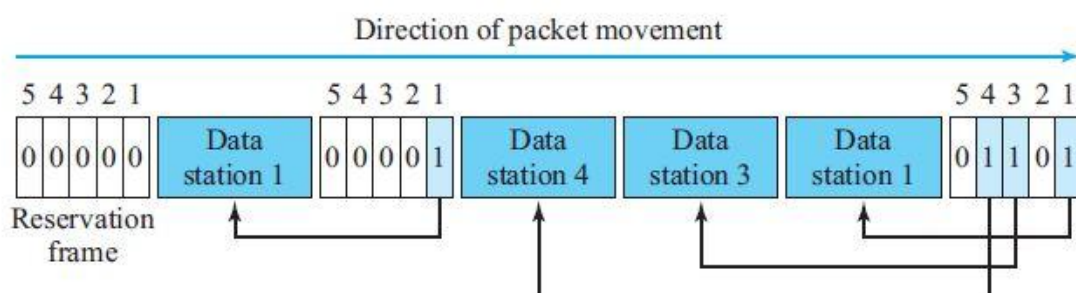


Figure 18: Reservation access method

- 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 mini slot 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.
- Figure 18 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.
- The primary device determines which device is allowed to use the channel at a given time. The primary device, therefore, is always the initiator of a session
- This method uses poll and select functions to prevent collisions. However, the drawback is if the primary station fails, the system goes down.

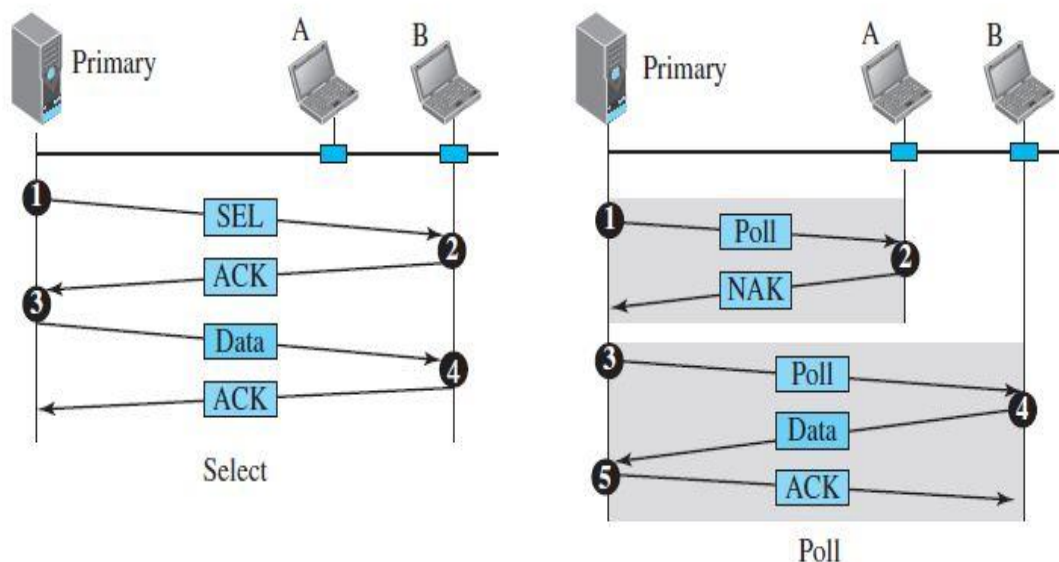


Figure 19: Select and poll functions in polling-access method

Select

- The select function is used whenever the primary device has something to send. Since the primary controls the link. If it is neither sending nor receiving data, it knows the link is available.
- If it has something to send, the primary device sends it. The primary station has to confirm whether the target device is prepared to receive.
- The primary must alert the secondary to the upcoming transmission and wait for an acknowledgment of the secondary's ready status. Before sending data, the primary creates and transmits a select (SEL) frame, one field of which includes the address of the intended secondary.

Poll

- The poll function is used by the primary device to solicit transmissions from the secondary devices.
- When the primary is ready to receive data, it must ask (poll) each device in turn if it has anything to send. When the first secondary is approached, it responds either with a NAK frame if it has nothing to send 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. 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. In this process, when a station receives the token and has no data to send, it just passes the data to the next station.
- 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. Figure 20 shows four different physical topologies that can create a logical ring.

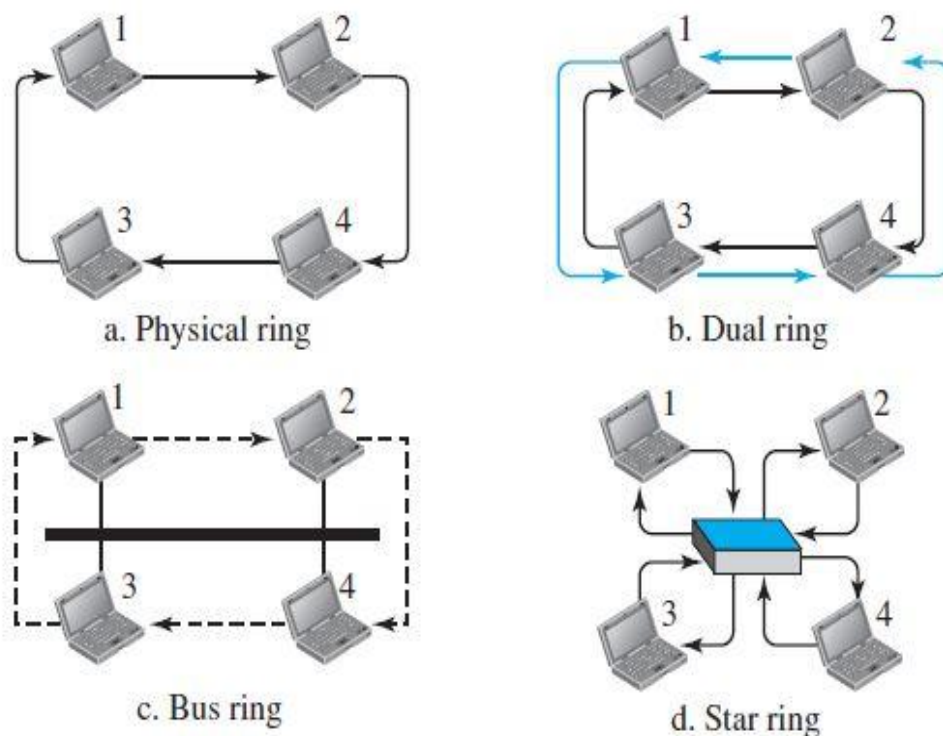


Figure 20: Logical ring and physical topology in token-passing access method

(a) Physical 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.

(b) Dual ring:

- 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 (such as a spare tire for a car).
- 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.
- Each station needs to have two transmitter ports and two receiver ports. The high-speed Token Ring networks called FDDI (Fiber Distributed Data Interface) and CDDI (Copper Distributed Data Interface) use this topology.

(c) Bus Ring:

- In the bus ring topology, also called a token bus, the stations are connected to a single cable called a bus. They, 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.

(d) Star Ring:

- 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.
- Adding and removing stations from the ring is easier. This topology is still used in the Token Ring LAN designed by IBM.

MODULE 2 (continued..)

Wired LANs: Ethernet

ETHERNET PROTOCOL

A local area network (LAN) is a computer network that is designed for a limited geographic area such as a building or a campus. Although a LAN can be used as an isolated network to connect computers in an organization for the sole purpose of sharing resources, most LANs today are also linked to a wide area network (WAN) or the Internet. Almost every LAN except Ethernet has disappeared from the marketplace because Ethernet was able to update itself to meet the needs of the time

IEEE Project 802

- In 1985, the Computer Society of the IEEE started a project, called **Project 802**, to set standards to enable intercommunication among equipment from a variety of manufacturers.
- Project 802 does not seek to replace any part of the OSI model or TCP/IP protocol suite. Instead, it is a way of specifying functions of the physical layer and the data-link layer of major LAN protocols.
- The relationship of the 802 Standard to the TCP/IP protocol suite is shown in Figure 13.1. The IEEE has subdivided the data-link layer into two sub layers:
 - Logical link control (LLC)
 - Media access control (MAC)

IEEE has also created several physical-layer standards for different LAN protocols.

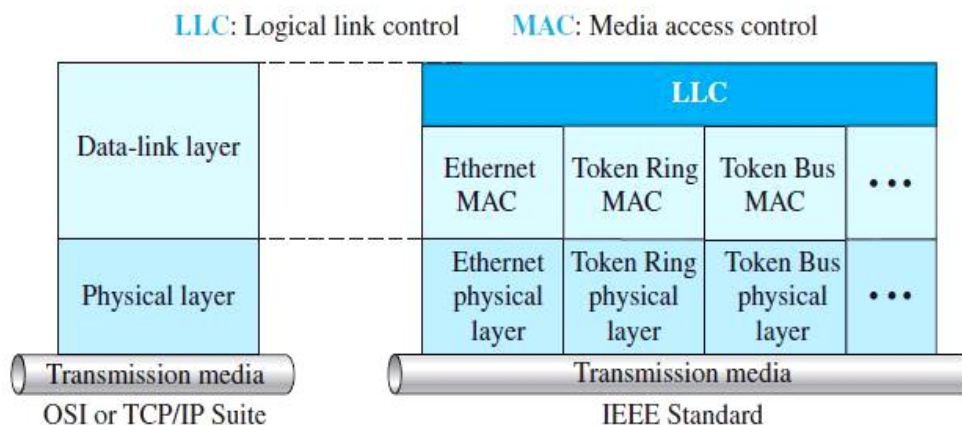


Figure 1: IEEE standard for LANs

Logical Link Control (LLC)

- In IEEE Project 802, flow control, error control, and part of the framing duties are collected into one sub layer called the logical link control (LLC). Framing is handled in both the LLC sublayer and the MAC sublayer.
- The LLC provides a single link-layer control protocol for all IEEE LANs. This means LLC protocol can provide interconnectivity between different LANs because it makes the MAC sub layer transparent.

Media Access Control (MAC)

- IEEE Project 802 has created a sublayer called media access control that defines the specific access method for each LAN. For example, it defines CSMA/CD as the media access method for Ethernet LANs and defines the token-passing method for Token Ring and Token Bus LANs.
- Part of the framing function is also handled by the MAC layer.

Ethernet Evolution

The Ethernet LAN was developed in the 1970s by Robert Metcalfe and David Boggs.

The four generations of Ethernet are :

1. Standard Ethernet (10 Mbps)
2. Fast Ethernet (100 Mbps)
3. Gigabit Ethernet (1 Gbps) and
4. 10 Gigabit Ethernet (10 Gbps)

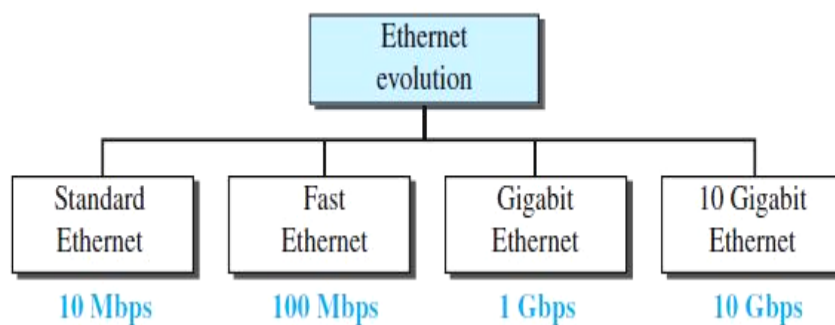


Figure 2: Ethernet evolution through four generations

STANDARD ETHERNET

Characteristics

1. Connectionless and Unreliable Service

- Ethernet provides a connectionless service, which means each frame sent is independent of the previous or next frame. Ethernet has no connection establishment or connection termination phases.
- The sender sends a frame whenever it has, the receiver may or may not be ready for it. The sender may overwhelm the receiver with frames, which may result in dropping frames. If a frame drops, the sender will not know about it. Since IP, which is using the service of Ethernet, is also connectionless, it will not know about it either. If the transport layer is also a connectionless protocol, such as UDP, the frame is lost and salvation may only come from the application layer. However, if the transport layer is TCP, the sender TCP does not receive acknowledgment for its segment and sends it again.
- Ethernet is also unreliable like IP and UDP. If a frame is corrupted during transmission and the receiver finds out about the corruption, which has a high level of probability of happening because of the CRC-32, the receiver drops the frame silently. It is the duty of high-level protocols to find out about it.

2. Frame Format

The Ethernet frame contains seven fields, as shown in Figure 3

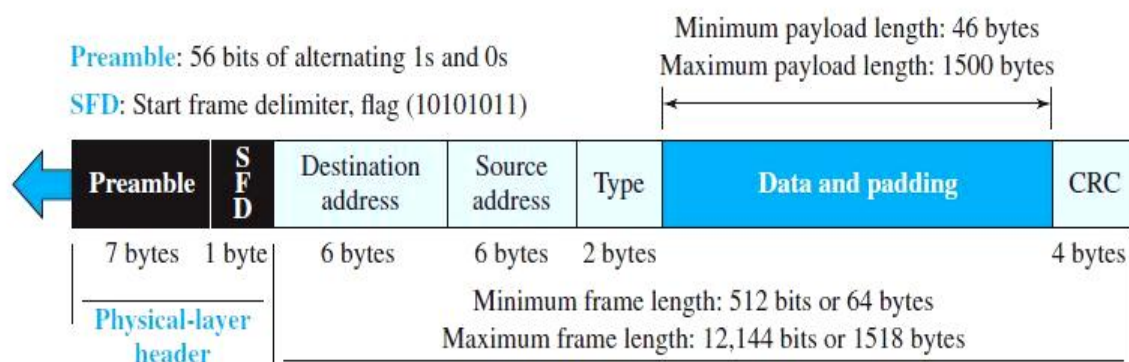


Figure 3: Ethernet frame

- **Preamble.** This field contains 7 bytes (56 bits) of alternating 0s and 1s that alert the receiving system to the coming frame and enable it to synchronize its clock if it's out of synchronization. The pattern provides only an alert and a timing pulse. The 56-bit pattern allows the stations to miss some bits at the beginning of the frame. The preamble is actually added at the physical layer and is not part of the frame.

- **Start frame delimiter (SFD).** This field (1 byte: 10101011) signals the beginning of the frame. The SFD warns the station or stations that this is the last chance for synchronization. The last 2 bits are $(11)_2$ and alert the receiver that the next field is the destination address. This field is actually a flag that defines the beginning of the frame, an Ethernet frame is a variable-length frame. It needs a flag to define the beginning of the frame. The SFD field is also added at the physical layer.

- **Destination address (DA).** This field is six bytes (48 bits) and contains the link layer address of the destination station or stations to receive the packet. When the receiver sees its own link-layer address, or a multicast address for a group that the receiver is a member of, or a broadcast address, it decapsulates the data from the frame and passes the data to the upper layer protocol defined by the value of the type field.

- **Source address (SA).** This field is also six bytes and contains the link-layer address of the sender of the packet.

- **Type.** This field defines the upper-layer protocol whose packet is encapsulated in the frame. This protocol can be IP, ARP, OSPF, and so on. In other words, it serves the same purpose as the protocol field in a datagram and the port number in a segment or user datagram. It is used for multiplexing and demultiplexing.

- **Data.** This field carries data encapsulated from the upper-layer protocols. It is a minimum of 46 and a maximum of 1500 bytes. If the data coming from the upper layer is more than 1500 bytes, it should be fragmented and encapsulated in more than one frame. If it is less than 46 bytes, it needs to be padded with extra 0s. A padded data frame is delivered to the upper-layer protocol as it is (without removing the padding), which means that it is the responsibility of the upper layer to remove or, in the case of the sender, to add the

padding. The upper-layer protocol needs to know the length of its data. For example, a datagram has a field that defines the length of the data.

- **CRC.** The last field contains error detection information, in this case a CRC-32. The CRC is calculated over the addresses, types, and data field. If the receiver calculates the CRC and finds that it is not zero (corruption in transmission), it discards the frame.

3. Frame Length

- Ethernet has imposed restrictions on both the minimum and maximum lengths of a frame. The minimum length restriction is required for the correct operation of CSMA/CD.
- An Ethernet frame needs to have a minimum length of 512 bits or 64 bytes. Part of this length is the header and the trailer. If we count 18 bytes of header and trailer (6 bytes of source address, 6 bytes of destination address, 2 bytes of length or type, and 4 bytes of CRC), then the minimum length of data from the upper layer is $64 - 18 = 46$ bytes. If the upper-layer packet is less than 46 bytes, padding is added to make up the difference.
- The standard defines the maximum length of a frame (without preamble and SFD field) as 1518 bytes. If we subtract the 18 bytes of header and trailer, the maximum length of the payload is 1500 bytes.
- The maximum length restriction has two historical reasons. First, memory was very expensive when Ethernet was designed; a maximum length restriction helped to reduce the size of the buffer. Second, the maximum length restriction prevents one station from monopolizing the shared medium, blocking other stations that have data to send.

NOTE:

Minimum frame length: 64 bytes

Minimum data length: 46 bytes

Maximum frame length: 1518 bytes

Maximum data length: 1500 bytes

Addressing

Each station on an Ethernet network (such as a PC, workstation, or printer) has its own network interface card (NIC). The NIC fits inside the station and provides the station with a link-layer address. The Ethernet address is 6 bytes (48 bits), normally written in hexadecimal notation, with a colon between the bytes. For example, the following shows an Ethernet MAC address:

4A:30:10:21:10:1A

Transmission of Address Bits

The way the addresses are sent out online is different from the way they are written in hexadecimal notation. The transmission is left to right, byte by byte; however, for each byte, the least significant bit is sent first and the most significant bit is sent last. This means that the bit that defines an address as unicast or multicast arrives first at the receiver. This helps the receiver to immediately know if the packet is unicast or multicast.

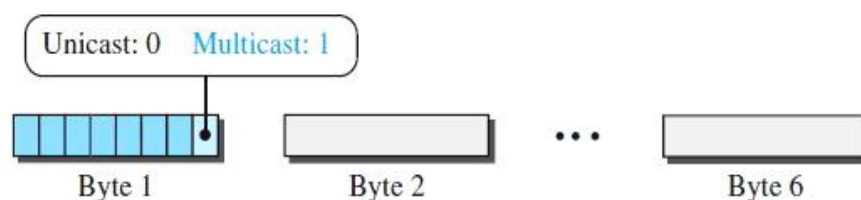
Example

Show how the address 47:20:1B:2E:08:EE is sent out online.

Solution: The address is sent left to right, byte by byte; for each byte, it is sent right to left, bit by bit, as shown below

Hexadecimal	47	20	1B	2E	08	EE
Binary	01000111	00100000	00011011	00101110	00001000	11101110
Transmitted ←	11100010	00000100	11011000	01110100	00010000	01110111

Unicast, Multicast, and Broadcast Addresses



A source address is always a unicast address, the frame comes from only one station. The destination address, however, can be unicast, multicast, or broadcast. Figure 4 shows how to

distinguish a unicast address from a multicast address. If the least significant bit of the first byte in a destination address is 0, the address is unicast; otherwise, it is multicast. With the way the bits are transmitted, the unicast/multicast bit is the first bit which is transmitted or received. The broadcast address is a special case of the multicast address: the recipients are all the stations on the LAN. A broadcast destination address is forty-eight 1s.

Example

Define the type of the following destination addresses

- a. 4A:30:10:21:10:1A
- b. 47:20:1B:2E:08:EE
- c. FF:FF:FF:FF:FF:FF

Solution: To find the type of the address, we need to look at the second hexadecimal digit from the left. If it is even, the address is unicast. If it is odd, the address is multicast. If all digits are Fs, the address is broadcast. Therefore, we have the following:

- a. This is a unicast address because A in binary is 1010 (even).
- b. This is a multicast address because 7 in binary is 0111 (odd).
- c. This is a broadcast address because all digits are Fs in hexadecimal.

Access Method

Since the network that uses the standard Ethernet protocol is a broadcast network, The standard Ethernet chose CSMA/CD with 1-persistent method, Let us use a scenario to see how this method works for the Ethernet protocol.

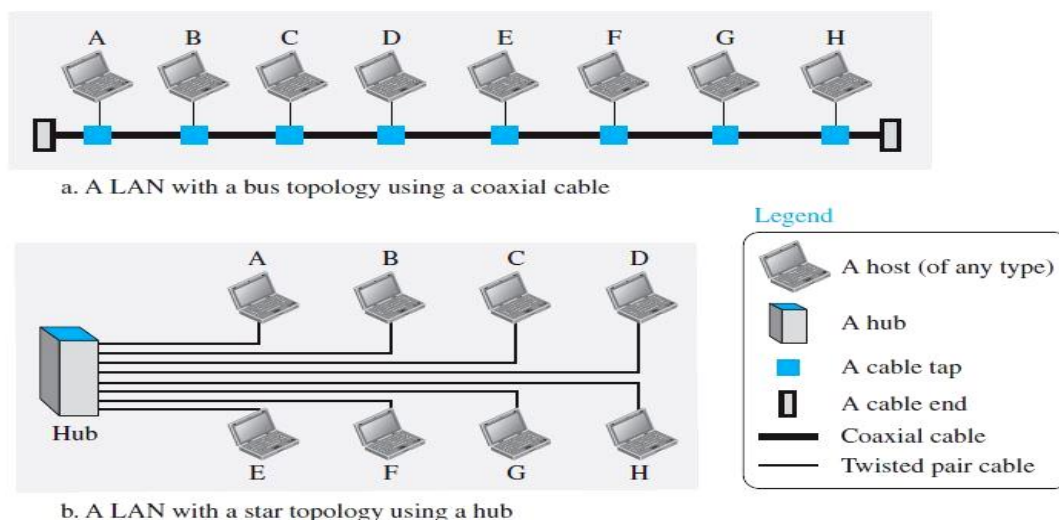


Figure 5: Implementation of standard Ethernet

- Assume station A in Figure.5 has a frame to send to station D. Station A first should check whether any other station is sending (carrier sense). Station A measures the level of energy on the medium (for a short period of time, normally less than $100\mu\text{s}$). If there is no signal energy on the medium, it means that no station is sending (or the signal has not reached station A). Station A interprets this situation as idle medium. It starts sending its frame. On the other hand, if the signal energy level is not zero, it means that the medium is being used by another station. Station A continuously monitors the medium until it becomes idle for $100\mu\text{s}$. It then starts sending the frame. However, station A needs to keep a copy of the frame in its buffer until it is sure that there is no collision.
- The medium sensing does not stop after station A has started sending the frame. Station A needs to send and receive continuously. Two cases may occur:
 - (a) Station A has sent 512 bits and no collision is sensed (the energy level did not go above the regular energy level), the station then is sure that the frame will go through and stops sensing the medium. Where does the number 512 bits come from? If we consider the transmission rate of the Ethernet as 10 Mbps, this means that it takes the station $512/(10 \text{ Mbps}) = 51.2 \mu\text{s}$ to send out 512 bits. With the speed of propagation in a cable (2×10^8 meters), the first bit could have gone 10,240 meters (one way) or only 5120 meters (round trip), have collided with a bit from the last station on the cable, and have gone back. In other words, if a collision were to occur, it should occur by the time the sender has sent out 512 bits (worst case) and the first bit has made a round trip of 5120 meters, if the collision happens in the middle of the cable, not at the end, station A hears the collision earlier and aborts the transmission. The above assumption is that the length of the cable is 5120 meters. The designer of the standard Ethernet actually put a restriction of 2500 meters because we need to consider the delays encountered throughout the journey. It means that they considered the worst case. The whole idea is that if station A does not sense the collision before sending 512 bits, there must have been no collision, because during this time, the first bit has reached the end of the line and all other stations know that a station is sending and refrain from sending. In other words, the problem occurs when another station (for example, the last station) starts sending before the first bit of station A has reached it. The other station mistakenly thinks that the line is free because the first bit has not yet reached it. The restriction of 512 bits actually helps the sending station: The sending

station is certain that no collision will occur if it is not heard during the first 512 bits, so it can discard the copy of the frame in its buffer.

- (b) Station A has sensed a collision before sending 512 bits. This means that one of the previous bits has collided with a bit sent by another station. In this case both stations should refrain from sending and keep the frame in their buffer for resending when the line becomes available. However, to inform other stations that there is a collision in the network, the station sends a 48-bit jam signal. The jam signal is to create enough signal (even if the collision happens after a few bits) to alert other stations about the collision. After sending the jam signal, the stations need to increment the value of K (number of attempts). If after increment $K = 15$, the experience has shown that the network is too busy, the station needs to abort its effort and try again. If $K < 15$, the station can wait a backoff time (T_B) and restart the process. The station creates a random number between 0 and $2^K - 1$, which means each time the collision occurs, the range of the random number increases exponentially. After the first collision ($K = 1$) the random number is in the range (0, 1). After the second collision ($K = 2$) it is in the range (0, 1, 2, 3). After the third collision ($K = 3$) it is in the range (0, 1, 2, 3, 4, 5, 6, 7). So after each collision, the probability increases that the backoff time becomes longer. This is due to the fact that if the collision happens even after the third or fourth attempt, it means that the network is really busy; a longer backoff time is needed.

Efficiency of Standard Ethernet

The efficiency of the Ethernet is defined as the ratio of the time used by a station to send data to the time the medium is occupied by this station. The practical efficiency of standard Ethernet has been measured to be,

$$\text{Efficiency} = \eta = \frac{1}{1 + 6.4 \times \alpha}$$

Where, α = the number of frames that can fit on the medium.

$$= \frac{\text{Propogation delay (Tp)}}{\text{Transmission delay (Tf)}}$$

The transmission delay is the time it takes a frame of average size to be sent out and the propagation delay is the time it takes to reach the end of the medium. As the value of parameter α decreases, the efficiency increases. This means that if the length of the media is

shorter or the frame size longer, the efficiency increases. In the ideal case, $\alpha = 0$ and the efficiency is 1.

Example 13.3

In the Standard Ethernet with the transmission rate of 10 Mbps, we assume that the length of the medium is 2500 m and the size of the frame is 512 bits. The propagation speed of a signal in a cable is normally 2×10^8 m/s.

$$\text{Propagation delay} = 2500 / (2 \times 10^8) = 12.5 \mu\text{s} \quad \text{Transmission delay} = 512 / (10^7) = 51.2 \mu\text{s}$$

$$a = 12.5 / 51.2 = 0.24 \quad \text{Efficiency} = 39\%$$

The example shows that $a = 0.24$, which means only 0.24 of a frame occupies the whole medium in this case. The efficiency is 39 percent, which is considered moderate; it means that only 61 percent of the time the medium is occupied but not used by a station.

Implementation

The Standard Ethernet defined several implementations, but only four of them became popular during the 1980s. Table below shows a summary of Standard Ethernet implementations.

<i>Implementation</i>	<i>Medium</i>	<i>Medium Length</i>	<i>Encoding</i>
10Base5	Thick coax	500 m	Manchester
10Base2	Thin coax	185 m	Manchester
10Base-T	2 UTP	100 m	Manchester
10Base-F	2 Fiber	2000 m	Manchester

In the nomenclature 10BaseX, the number defines the data rate (10 Mbps), the term *Base* means baseband (digital) signal, and X approximately defines either the maximum size of the cable in 100 meters (for example 5 for 500 or 2 for 185 meters) or the type of cable, T for unshielded twisted pair cable (UTP) and F for fiber-optic. The standard Ethernet uses a baseband signal, which means that the bits are changed to a digital signal and directly sent on the line.

Encoding and Decoding

All standard implementations use digital signalling (baseband) at 10 Mbps. At the sender, data are converted to a digital signal using the Manchester scheme; at the receiver, the received signal is interpreted as Manchester and decoded into data. Manchester encoding is

self-synchronous, providing a transition at each bit interval. Figure 6 shows the encoding scheme for Standard Ethernet.

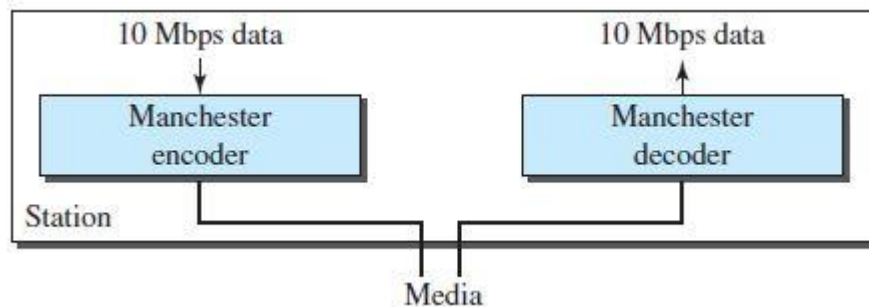


Figure 6: Encoding in a Standard Ethernet implementation

10Base5: Thick Ethernet

The first implementation is called 10Base5, thick Ethernet, or Thicknet. The nickname derives from the size of the cable, which is roughly the size of a garden hose and too stiff to bend with your hands. 10Base5 was the first Ethernet specification to use a bus topology with an external transceiver (transmitter/receiver) connected via a tap to a thick coaxial cable. Figure 7 shows a schematic diagram of a 10Base5 implementation.

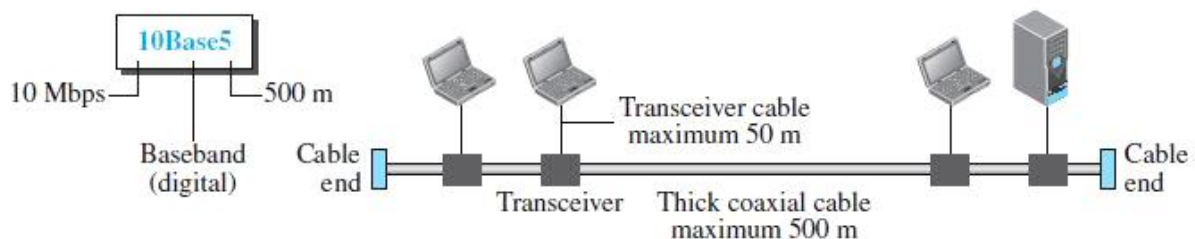


Figure 7 : 10Base5 implementation

The transceiver is responsible for transmitting, receiving, and detecting collisions. The transceiver is connected to the station via a transceiver cable that provides separate paths for sending and receiving. This means that collision can only happen in the coaxial cable.

The maximum length of the coaxial cable must not exceed 500 m, otherwise, there is excessive degradation of the signal. If a length of more than 500 m is needed, up to five segments, each a maximum of 500 meters, can be connected using repeaters.

10Base2: Thin Ethernet

The second implementation is called 10Base2, thin Ethernet, or Cheapernet. 10Base2 also uses a bus topology, but the cable is much thinner and more flexible. The cable can be bent to pass very close to the stations. In this case, the transceiver is normally part of the network

interface card (NIC), which is installed inside the station. Figure 8 shows the schematic diagram of a 10Base2 implementation.

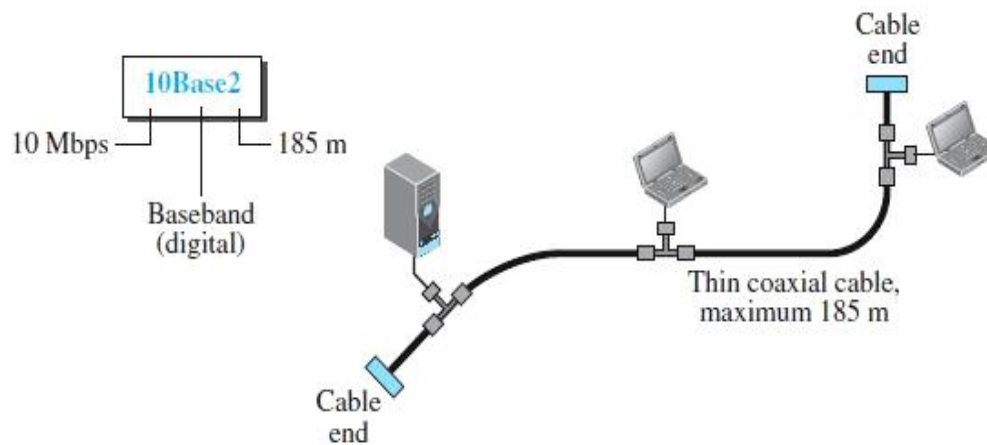


Figure 8: 10Base2 implementation

The collision here occurs in the thin coaxial cable. This implementation is more cost effective than 10Base5 because thin coaxial cable is less expensive than thick coaxial and the tee connections are much cheaper than taps. Installation is simpler because the thin coaxial cable is very flexible. However, the length of each segment cannot exceed 185 m (close to 200 m) due to the high level of attenuation in thin coaxial cable.

10Base-T: Twisted-Pair Ethernet

The third implementation is called **10Base-T** or **twisted-pair Ethernet**. 10Base-T uses a physical star topology. The stations are connected to a hub via two pairs of twisted cable, as shown in Figure 9.

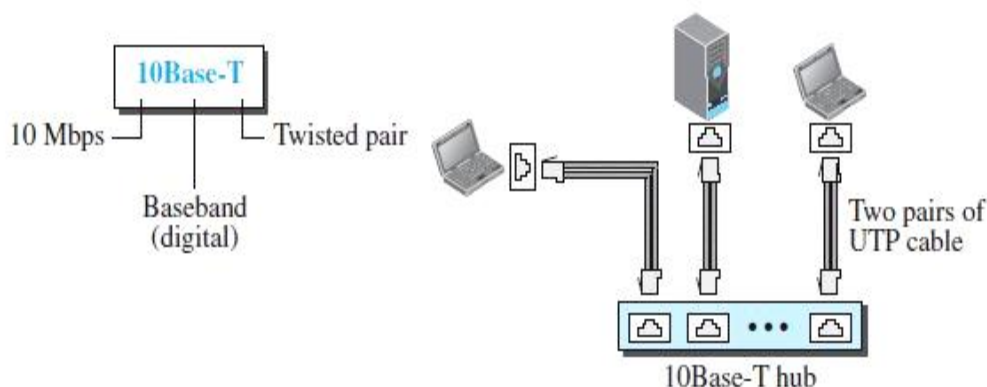


Figure 9: 10Base-T implementation

Two pairs of twisted cable create two paths (one for sending and one for receiving) between the station and the hub. Any collision here happens in the hub. Compared to 10Base5 or

10Base2, we can see that the hub actually replaces the coaxial cable as far as a collision is concerned. The maximum length of the twisted cable here is defined as 100 m, to minimize the effect of attenuation in the twisted cable.

10Base-F: Fiber Ethernet

Although there are several types of optical fiber 10-Mbps Ethernet, the most common is called **10Base-F**. 10Base-F uses a star topology to connect stations to a hub. The stations are connected to the hub using two fiber-optic cables, as shown in Figure 10.

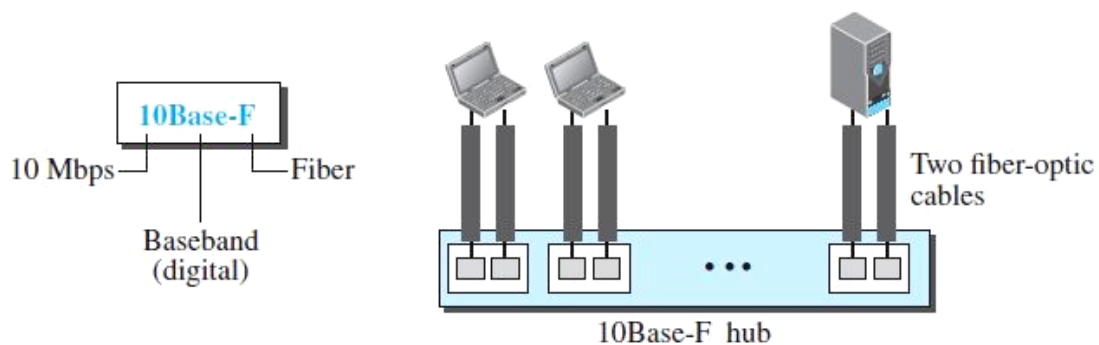


Figure 10: 10Base-F implementation

FAST ETHERNET (100 MBPS)

In the 1990s, some LAN technologies with transmission rates higher than 10 Mbps, such as FDDI and Fiber Channel, appeared on the market. If the Standard Ethernet wanted to survive, it had to compete with these technologies. Ethernet made a big jump by increasing the transmission rate to 100 Mbps, and the new generation was called the Fast Ethernet. The designers of the Fast Ethernet needed to make it compatible with the Standard Ethernet. The MAC sublayer was left unchanged, which meant the frame format and the maximum and minimum size could also remain unchanged. By increasing the transmission rate, features of the Standard Ethernet that depend on the transmission rate, access method, and implementation had to be reconsidered.

The goals of Fast Ethernet can be summarized as follows:

1. Upgrade the data rate to 100 Mbps.
2. Make it compatible with Standard Ethernet.
3. Keep the same 48-bit address.
4. Keep the same frame format.

Access Method

The proper operation of the CSMA/CD depends on the transmission rate, the minimum size of the frame, and the maximum network length. If we want to keep the minimum size of the frame, the maximum length of the network should be changed. In other words, if the minimum frame size is still 512 bits, and it is transmitted 10 times faster, the collision needs to be detected 10 times sooner, which means the maximum length of the network should be 10 times shorter (the propagation speed does not change). So the Fast Ethernet came with two solutions (it can work with either choice):

1. The first solution was to totally drop the bus topology and use a passive hub and star topology but make the maximum size of the network 250 meters instead of 2500 meters as in the Standard Ethernet. This approach is kept for compatibility with the Standard Ethernet.
2. The second solution is to use a link-layer switch with a buffer to store frames and a full-duplex connection to each host to make the transmission medium private for each host. In this case, there is no need for CSMA/CD because the hosts are not competing with each other. The link-layer switch receives a frame from a source host and stores it in the buffer (queue) waiting for processing. It then checks the destination address and sends the frame out of the corresponding interface. Since the connection to the switch is full-duplex, the destination address can even send a frame to another station at the same time that it is receiving a frame. In other words, the shared medium is changed to many point-to-point media, and there is no need for contention.

Autonegotiation

A new feature added to Fast Ethernet is called autonegotiation. It allows a station or a hub a range of capabilities. Autonegotiation allows two devices to negotiate the mode or data rate of operation. It was designed particularly to allow incompatible devices to connect to one another.

It was designed particularly for these purposes:

- To allow incompatible devices to connect to one another. For example, a device with a maximum capacity of 10 Mbps can communicate with a device with a 100 Mbps capacity (but which can work at a lower rate).
- To allow one device to have multiple capabilities.
- To allow a station to check a hub's capabilities.

Physical Layer

Topology

Fast Ethernet is designed to connect two or more stations. If there are only two stations, they can be connected point-to-point. Three or more stations need to be connected in a star topology with a hub or a switch at the center.

Encoding

Manchester encoding needs a 200-Mbaud bandwidth for a data rate of 100 Mbps, which makes it unsuitable for a medium such as twisted-pair cable. For this reason, the Fast Ethernet designers sought some alternative encoding/decoding scheme. However, it was found that one scheme would not perform equally well for all three implementations. Therefore, three different encoding schemes were chosen.

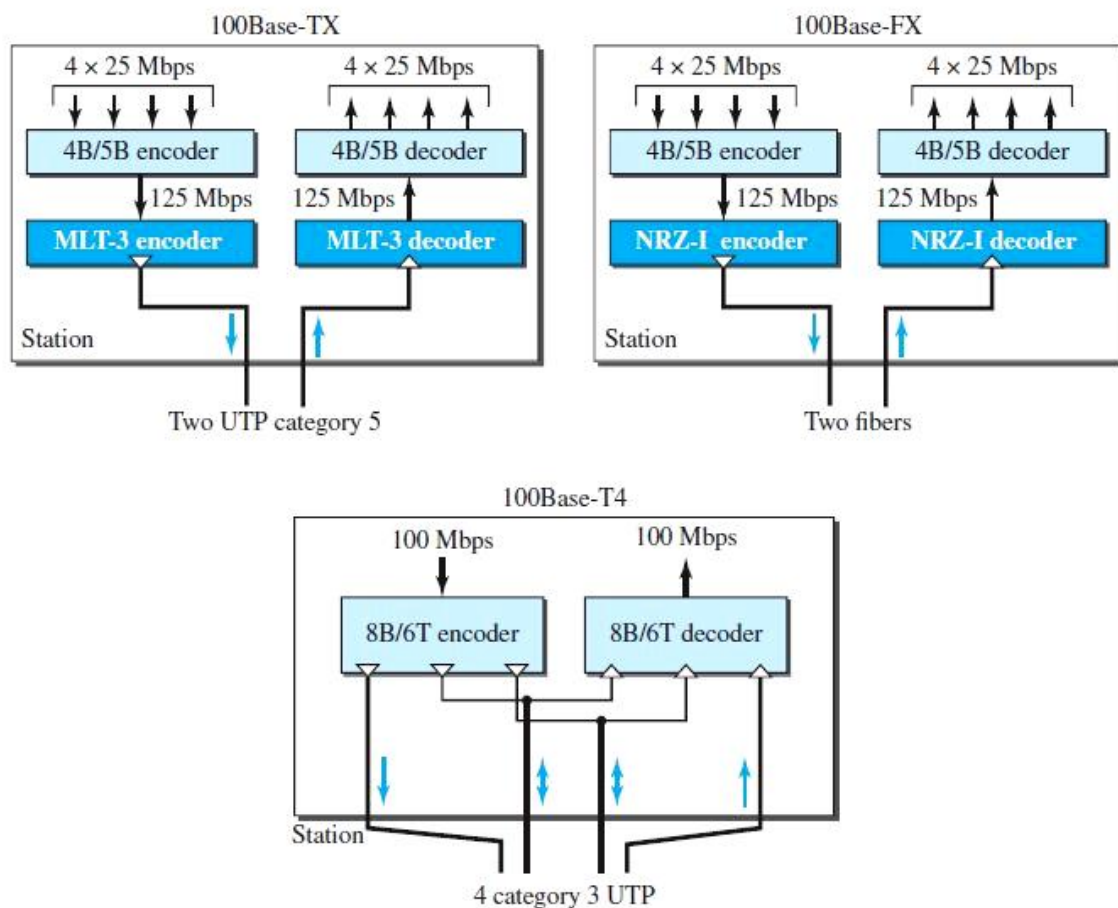


Figure 11: Encoding for fast Ethernet implementations

1. **100Base-TX** uses two pairs of twisted-pair cable (either category 5 UTP or STP). For this implementation, the MLT-3 scheme was selected since it has good bandwidth performance. However, since MLT-3 is not a self-synchronous line coding scheme, 4B/5B block coding is used to provide bit synchronization by preventing the

occurrence of a long sequence of 0s and 1s. This creates a data rate of 125 Mbps, which is fed into MLT-3 for encoding.

2. **100Base-FX** uses two pairs of fiber-optic cables. Optical fiber can easily handle high bandwidth requirements by using simple encoding schemes. The designers of 100Base-FX selected the NRZ-I encoding scheme for this implementation. However, NRZ-I has a bit synchronization problem for long sequences of 0s (or 1s, based on the encoding). To overcome this problem, the designers used 4B/5B block encoding, as we described for 100Base-TX. The block encoding increases the bit rate from 100 to 125 Mbps, which can easily be handled by fiber-optic cable.

A 100Base-TX network can provide a data rate of 100 Mbps, but it requires the use of category 5 UTP or STP cable. This is not cost-efficient for buildings that have already been wired for voice-grade twisted-pair (category 3).

3. **100Base-T4**, was designed to use category 3 or higher UTP. The implementation uses four pairs of UTP for transmitting 100 Mbps. Encoding/decoding in 100Base-T4 is more complicated. As this implementation uses category 3 UTP, each twisted-pair cannot easily handle more than 25 Mbaud. In this design, one pair switches between sending and receiving. Three pairs of UTP category 3, however, can handle only 75 Mbaud (25 Mbaud) each. We need to use an encoding scheme that converts 100 Mbps to a 75 Mbaud signal. 8B/6T satisfies this requirement. In 8B/6T, eight data elements are encoded as six signal elements. This means that 100 Mbps uses only $(6/8) \times 100$ Mbps, or 75 Mbaud.

<i>Implementation</i>	<i>Medium</i>	<i>Medium Length</i>	<i>Wires</i>	<i>Encoding</i>
100Base-TX	UTP or STP	100 m	2	4B5B + MLT-3
100Base-FX	Fiber	185 m	2	4B5B + NRZ-I
100Base-T4	UTP	100 m	4	Two 8B/6T

GIGABIT ETHERNET

The need for an even higher data rate resulted in the design of the Gigabit Ethernet Protocol (1000 Mbps). The IEEE committee calls it the Standard 802.3z. The goals of the Gigabit Ethernet were to upgrade the data rate to 1 Gbps, but keep the address length, the frame format, and the maximum and minimum frame length the same. The goals of the Gigabit Ethernet design can be summarized as follows:

1. Upgrade the data rate to 1 Gbps.
2. Make it compatible with Standard or Fast Ethernet.
3. Use the same 48-bit address.
4. Use the same frame format.
5. Keep the same minimum and maximum frame lengths.
6. Support autonegotiation as defined in Fast Ethernet.

MAC Sublayer

A main consideration in the evolution of Ethernet was to keep the MAC sublayer untouched. However, to achieve a data rate of 1 Gbps, this was no longer possible. Gigabit Ethernet has two distinctive approaches for medium access: half-duplex and fullduplex. Almost all implementations of Gigabit Ethernet follow the full-duplex approach, so we mostly ignore the half-duplex mode.

Full-Duplex Mode

In full-duplex mode, there is a central switch connected to all computers or other switches. In this mode, for each input port, each switch has buffers in which data are stored until they are transmitted. Since the switch uses the destination address of the frame and sends a frame out of the port connected to that particular destination, there is no collision. This means that CSMA/CD is not used. Lack of collision implies that the maximum length of the cable is determined by the signal attenuation in the cable, not by the collision detection process.

NOTE: In the full-duplex mode of Gigabit Ethernet, there is no collision; the maximum length of the cable is determined by the signal attenuation in the cable.

Half-Duplex Mode

The half-duplex approach uses CSMA/CD. the maximum length of the network in this approach is totally dependent on the minimum frame size.

Three methods have been defined:

- Traditional
- Carrier extension, and
- Frame bursting.

Traditional

In the traditional approach, we keep the minimum length of the frame as in traditional Ethernet (512 bits). However, because the length of a bit is 1/100 shorter in Gigabit Ethernet than in 10-Mbps Ethernet, the slot time for Gigabit Ethernet is $512 \text{ bits} \times 1/1000 \mu\text{s}$, which is equal to $0.512 \mu\text{s}$. The reduced slot time means that collision is detected 100 times earlier. This means that the maximum length of the network is 25 m. This length may be suitable if all the stations are in one room, but it may not even be long enough to connect the computers in one single office.

Carrier Extension

To allow for a longer network, we increase the minimum frame length. The carrier extension approach defines the minimum length of a frame as 512 bytes (4096 bits). This means that the minimum length is 8 times longer. This method forces a station to add extension bits (padding) to any frame that is less than 4096 bits. In this way, the maximum length of the network can be increased 8 times to a length of 200 m. This allows a length of 100m from the hub to the station.

Frame Bursting

Carrier extension is very inefficient if we have a series of short frames to send; each frame carries redundant data. To improve efficiency, **frame bursting** was proposed. Instead of adding an extension to each frame, multiple frames are sent. However, to make these multiple frames look like one frame, padding is added between the frames (the same as that used for the carrier extension method) so that the channel is not idle. In other words, the method deceives other stations into thinking that a very large frame has been transmitted.

Physical Layer

The physical layer in Gigabit Ethernet is more complicated than that in Standard or Fast Ethernet.

Topology

Gigabit Ethernet is designed to connect two or more stations. If there are only two stations, they can be connected point-to-point. Three or more stations need to be connected in a star topology with a hub or a switch at the center.

Implementation

Gigabit Ethernet can be categorized as either a two-wire or a four-wire implementation. The two-wire implementations use fiber-optic cable (**1000Base-SX**, short-wave, or **1000Base-**

LX, long-wave), or STP (**1000Base-CX**). The four-wire version uses category 5 twisted-pair cable (**1000Base-T**).

Implementation	Medium	Medium Length	Wires	Encoding
1000Base-SX	Fiber S-W	550 m	2	8B/10B + NRZ
1000Base-LX	Fiber L-W	5000 m	2	8B/10B + NRZ
1000Base-CX	STP	25 m	2	8B/10B + NRZ
1000Base-T4	UTP	100 m	4	4D-PAM5

Encoding

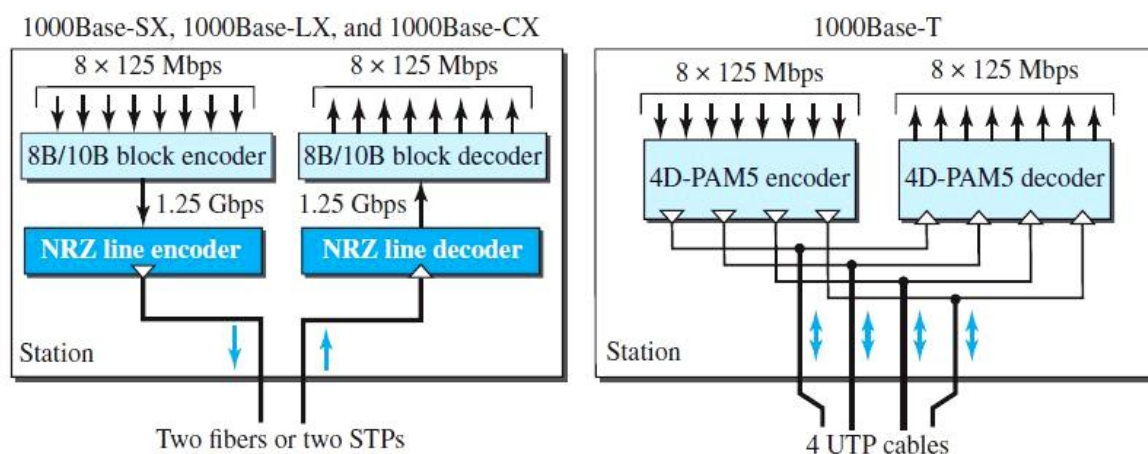


Figure 12: Encoding in Gigabit Ethernet implementations

Figure 12 shows the encoding/decoding schemes for the four implementations.

- Gigabit Ethernet cannot use the Manchester encoding scheme because it involves a very high bandwidth (2 GBaud).
- The two-wire implementations use an NRZ scheme, but NRZ does not self-synchronize properly. To synchronize bits, particularly at this high data rate, 8B/10B block encoding, is used. This block encoding prevents long sequences of 0s or 1s in the stream, but the resulting stream is 1.25 Gbps. In this implementation, one wire (fiber or STP) is used for sending and one for receiving.
- In the four-wire implementation it is not possible to have 2 wires for input and 2 for output, because each wire would need to carry 500 Mbps, which exceeds the capacity for category 5 UTP. As a solution, 4D-PAM5 encoding, is used to reduce the bandwidth. Thus, all four wires are involved in both input and output; each wire carries 250 Mbps, which is in the range for category 5 UTP cable.

10 GIGABIT ETHERNET

- The IEEE committee created 10 Gigabit Ethernet and called it Standard 802.3ae.
- The goals of the 10 Gigabit Ethernet design can be summarized as upgrading the data rate to 10 Gbps, keeping the same frame size and format, and allowing the interconnection of LANs, MANs, and WAN possible.
- This data rate is possible only with fiber-optic technology at this time. The standard defines two types of physical layers: LAN PHY and WAN PHY. The first is designed to support existing LANs; the second actually defines a WAN with links connected through SONET OC-192.

Implementation

10 Gigabit Ethernet operates only in full-duplex mode, which means there is no need for contention; CSMA/CD is not used in 10 Gigabit Ethernet.

Four implementations are the most common:

1. 10GBase-SR
2. 10GBase-LR
3. 10GBase-EW and
4. 10GBase-X4.

<i>Implementation</i>	<i>Medium</i>	<i>Medium Length</i>	<i>Number of wires</i>	<i>Encoding</i>
10GBase-SR	Fiber 850 nm	300 m	2	64B66B
10GBase-LR	Fiber 1310 nm	10 Km	2	64B66B
10GBase-EW	Fiber 1350 nm	40 Km	2	SONET
10GBase-X4	Fiber 1310 nm	300 m to 10 Km	2	8B10B

Table: Summary of 10 Gigabit Ethernet implementations