RANDOM ACCESS

--In random access or contention methods, no station is superior to another station and none is assigned the control over another.

--A station that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send. This decision depends on the state of the medium (idle or busy).

--Two features give this method its name. First, there is **no scheduled time** for a station to transmit. Transmission is **random** among the stations.

--Second, **no rules specify which station should send next**. Stations compete with one another to access the medium, so called *contention* methods.

--If more than one station sends, an access conflict-collision occurs and frames are destroyed.

--To avoid access conflict & resolve it, each station follows a procedure that answers the following questions:

-When can the station access the medium?

-What can the station do if the medium is busy?

-How can the station determine the success or failure of the transmission?

-What can the station do if there is an access conflict?

--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**.

--This method later evolved into two parallel methods: **(CSMA/CD) and (CSMA/CA)**.

--CSMA/CD tells station what to do when collision is detected. CSMA/CA tries to avoid the collision.

ALOHA

--ALOHA, earliest random access method was developed at the University of Hawaii in 1970.

--Designed for a radio (wireless) LAN, but it can be used on any shared medium.

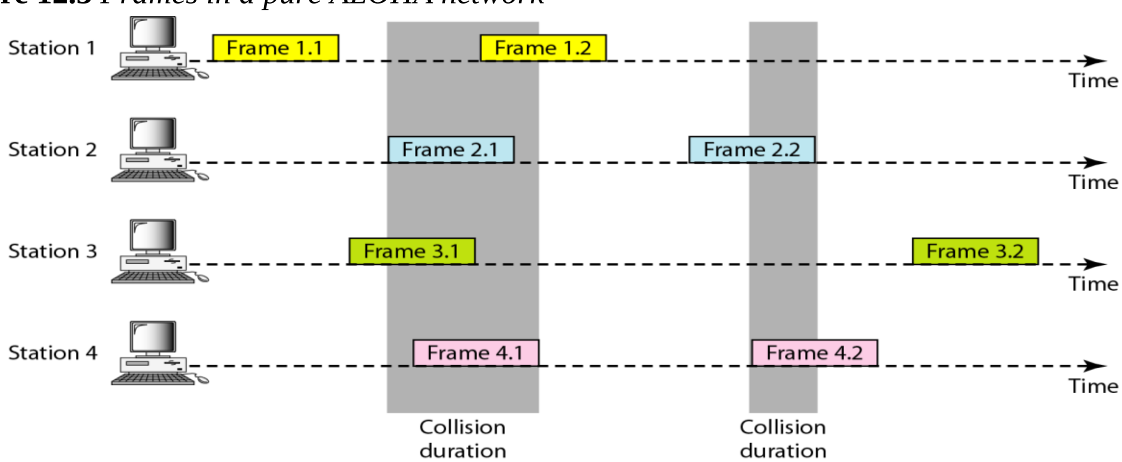
--There are potential collisions in this arrangement as the medium is shared between the stations.

Pure ALOHA

--The original ALOHA protocol is called pure ALOHA. This is a simple protocol and the idea is that each station sends a frame whenever it has a frame to send.

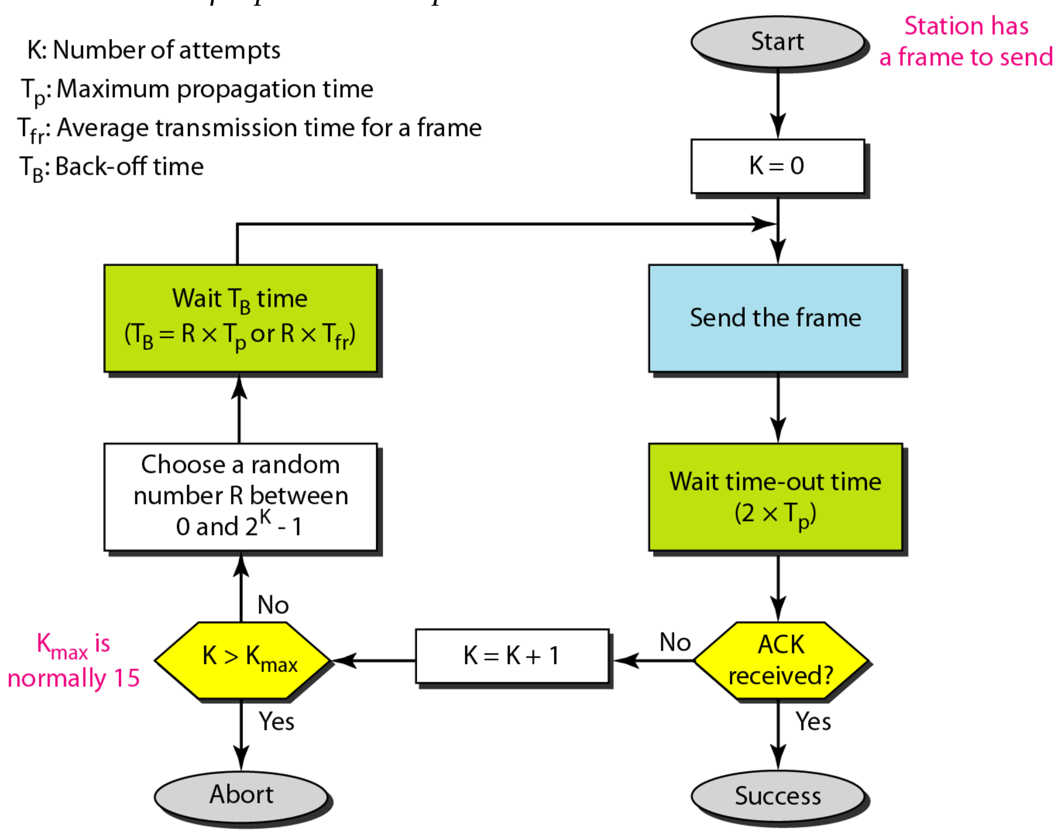
--Since there is only one channel to share, there is the possibility of collision between frames from different stations. Figure shows an example of frame collisions in pure ALOHA.

Figure 12.3 *Frames in a pure ALOHA network*



--There are four stations that contend with one another for access to the shared channel.
--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.
--In last only two frames survive: frame 1.1 from station 1 and frame 3.2 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 destroyed and frames are resent that have 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.
--In Pure ALOHA when the time-out period passes, each station waits a random amount of time before resending its frame which avoid more collisions. This time is called the **back-off time T_B**.
--Pure ALOHA has a second method to prevent congesting the channel with retransmitted frames. After maximum number of retransmission attempts **K_{max}**, a station give up and try later.

Figure 12.4 *Procedure for pure ALOHA protocol*



--The time-out period is equal to the maximum possible round-trip propagation delay, which is twice the amount of time required to send a frame between the two most widely separated stations (2 x 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. One common formula is the **binary exponential back-off.**
--For each retransmission, a multiplier in the range 0 to 2^K - 1 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.

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

Example 12.1

The stations on a wireless ALOHA networks are a maximum of 600 km apart. If we assume that signals propagate at 3 x 10^8 m/s, we find T_p = (600 x 10^5) / (3 x 10^8) = 2 ms. Now we can find the value of T_B for different values of K.

a. For K = 1, the range is {0, 1}. The station needs to generate a random number with a value of 0 or 1. This means that T_B is either 0 ms (0 x 2) or 2 ms (l x 2), based on the outcome of the random variable.

b. For K =2, the range is {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.

c. For K =3, the range is {0, 1, 2, 3,4,5,6, 7}. This means that *TB* can be 0, 2, 4... 14 ms, based on the outcome of the random variable.

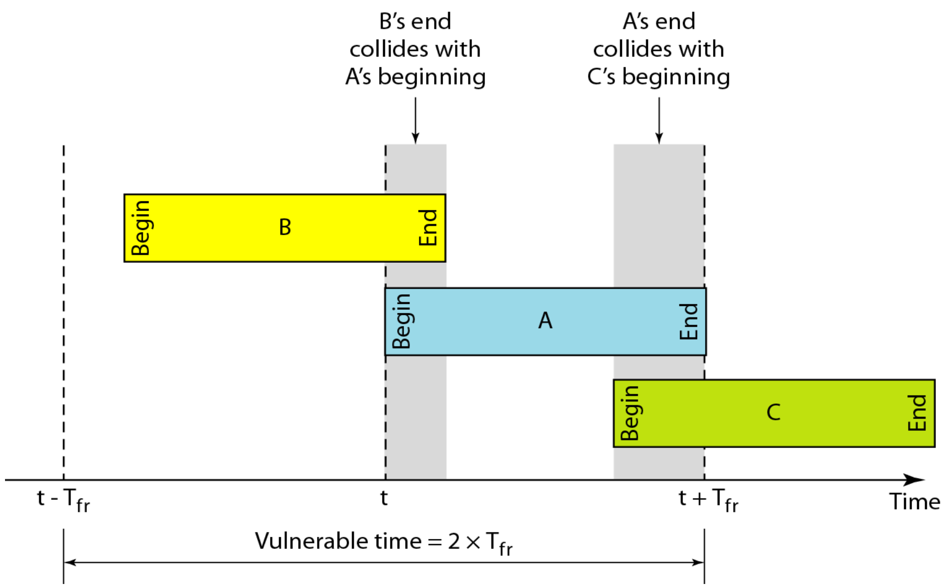
d. We need to mention that if K > 10, it is normally set to 10.

Vulnerable time

--Let us find the length of time, the **vulnerable time,** in which there is a possibility of collision.

--We assume that the stations send fixed-length frames with each frame taking T_{fr} s to send.

Figure 12.5 *Vulnerable time for pure ALOHA protocol*



--Station A sends a frame at time t. Now imagine station B has already sent a frame between t - T_{fr} and t. This leads to a collision between the frames from station A and station B. The end of B's frame collides with the beginning of A's frame.

--On the other hand, suppose that station C sends a frame between t and t + T_{fr} . Here, there is a collision between frames from station A and station C. The beginning of C's frame collides with the end of A's frame.

--The vulnerable time, during which a collision may occur in pure ALOHA, is 2 times the frame transmission time.

Example 12.2

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 *Tfr* is 200 bits/200 kbps or 1 ms. The vulnerable time is 2 x 1 ms =2 ms. This means no station should send later than 1 ms before this station starts transmission and no station should start sending during the one 1-ms period that this station is sending.

Throughput

--If G be the average number of frames generated by the system during one frame transmission time. Then it can be proved that the average number of successful transmissions for pure ALOHA is S = G x e^{-2G}.

--The maximum throughput S_{max} is 0.184, for G = 1/2. In other words, if one-half a frame is generated during one frame transmission time (in other words, one frame during two frame transmission times), then 18.4 percent of these frames reach their destination successfully. This is an expected result because the vulnerable time is 2 times the frame transmission time.

--Therefore, if a station generates only one frame in this vulnerable time, the frame will reach its destination successfully.

Example 12.3

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?

a. 1000 frames per second

b. 500 frames per second

c. 250 frames per second

Solution

The frame transmission time is *200/200* kbps or 1 ms.

a. If the system creates 1000 frames per second, this is 1 frame per millisecond. The load is 1. In this case S =G x e^{-2G} or S =0.135 (13.5 percent). This means that the throughput is 1000 X 0.135 =135 frames. Only 135 frames out of 1000 will probably survive.

b. If the system creates 500 frames per second, this is (1/2) frame per millisecond. The load is *(1/2).* In this case S = G x e^{-2G} or S = 0.184 (18.4 percent). This means that the throughput is 500 x 0.184 =92 and that only 92 frames out of 500 will probably survive. Note that this is the maximum throughput case, percentage wise.

c. If the system creates 250 frames per second, this is (1/4) frame per millisecond. The load is (1/4). In this case S = G x e^{-2G} or S =0.152 (15.2 percent). This means that the throughput is 250 x 0.152 = 38. Only 38 frames out of 250 will probably survive.

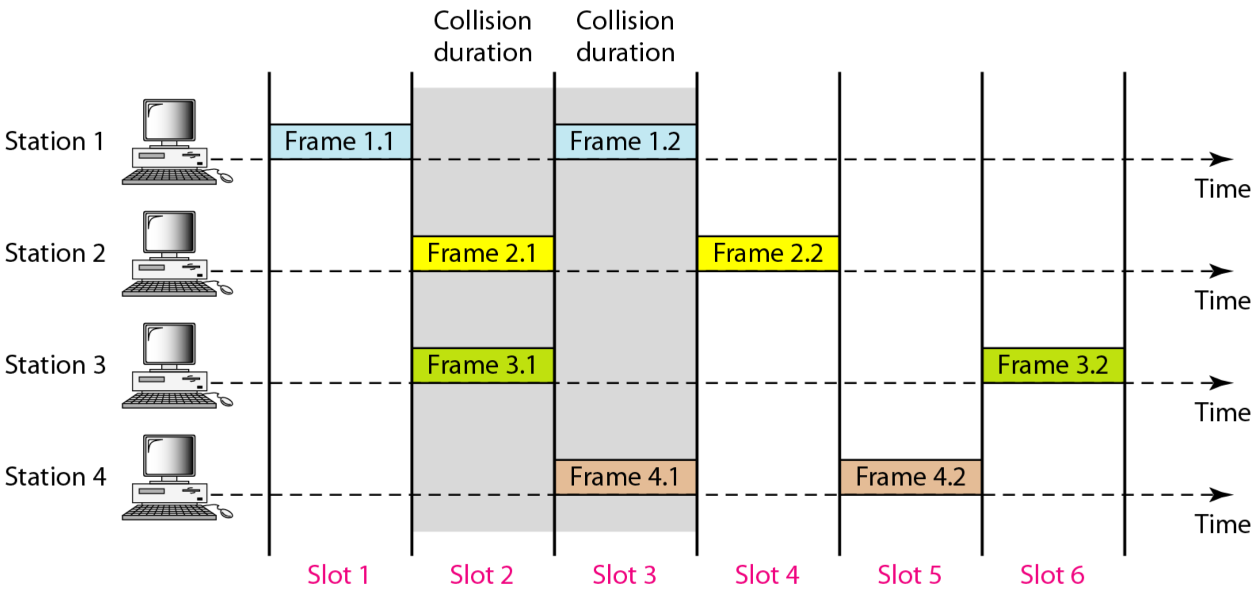
Slotted ALOHA

--Pure ALOHA has a vulnerable time of 2 x T_{fr}. This is so because there is no rule that defines when the station can send. A station may send soon after another station has started or soon before another station has finished.

--Slotted ALOHA was invented to improve the efficiency of pure ALOHA.

--In slotted ALOHA we divide the time into slots of T_{fr} s and force the station to send only at the beginning of the time slot.

Figure 12.6 *Frames in a slotted ALOHA network*



--Because a station is allowed to send only at the beginning of the synchronized time slot, if a station misses this moment, it must wait until the beginning of the next time slot. This means that the station which started at the beginning of this slot has already finished sending its frame.

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

--However, the vulnerable time is now reduced to one-half, equal to T_{fr}.

--Figure shows that the vulnerable time for slotted ALOHA is one-half that of pure ALOHA.

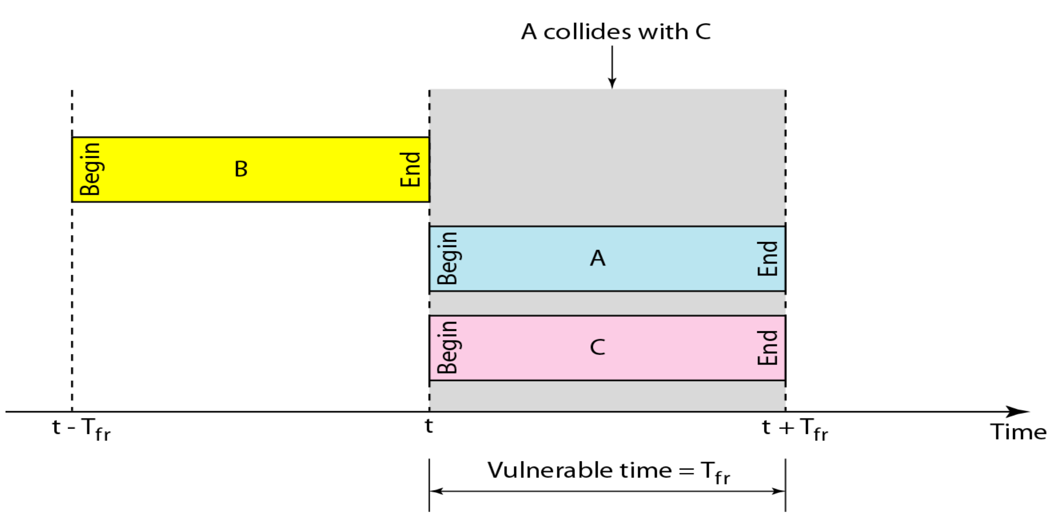
Throughput

--It can be proved that the average number of successful transmissions for slotted ALOHA is S = G x e^{-G}. The maximum throughput S_{max} is 0.368, when G = 1.

--In other words, if a frame is generated during one frame transmission time, then 36.8 percent of these frames reach their destination successfully. This result can be expected 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 station generates a frame during this time), the frame will reach its destination successfully.

Figure 12.7 *Vulnerable time for slotted ALOHA protocol*



Example 12.4

A slotted ALOHA network transmits 200-bit frames using a shared channel with a 200-kbps bandwidth. Find the throughput if the system (all stations together) produces

a. 1000 frames per second

b. 500 frames per second

c. 250 frames per second

Solution

The frame transmission time is *200/200* kbps or 1 ms.

a. In this case G is 1. So S =G x e^{-G} or S =0.368 (36.8 percent). This means that the throughput is 1000 x 0.0368 =368 frames. Only 368 out of 1000 frames will probably survive. Note that this is the maximum throughput case, percentage wise.

b. Here G is 1/2. In this case S =G x e^{-G} or S =0.303 (30.3 percent). This means that the throughput is 500 x 0.0303 =151. Only 151 frames out of 500 will probably survive.

c. Now G is 1/4. In this case S =G x e^{-G} or S =0.195 (19.5 percent). This means that the throughput is 250 x 0.195 = 49. Only 49 frames out of 250 will probably survive.

Carrier Sense Multiple Access (CSMA)

--To minimize collision & 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.

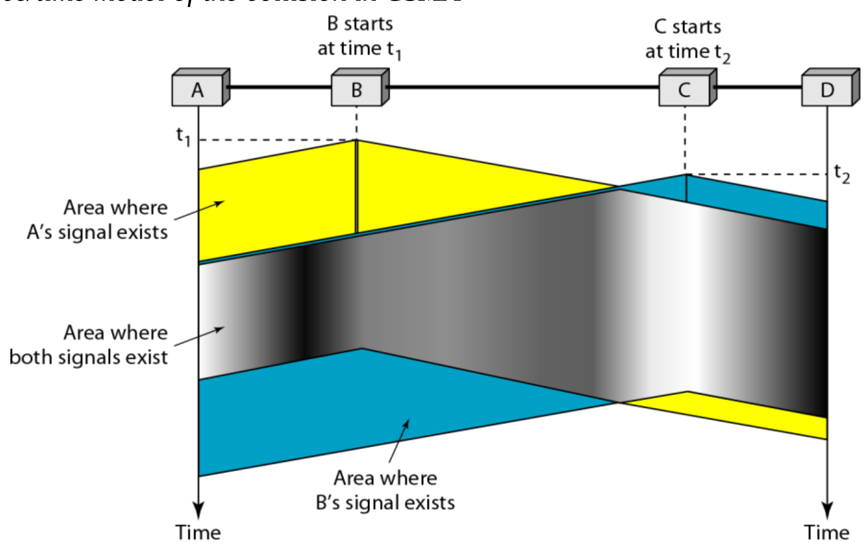
--In CSMA each station listens to the medium (check the state of the medium) before sending.

--In other words, CSMA is based on the principle "sense before transmit" or "listen before talk."

--CSMA can reduce the possibility of collision, but it cannot eliminate it.

--The reason for this is shown in Fig, a space and time model of a CSMA network. Stations are connected to a shared channel.

Figure 12.8 *Space/time model of the collision in CSMA*



--The possibility of collision still exists because of propagation delay; when a station sends a frame, it still takes time for the first bit to reach every station and for every station to sense it.

--A station may sense the medium and find it idle, only because the first bit sent by another station has not yet been received.

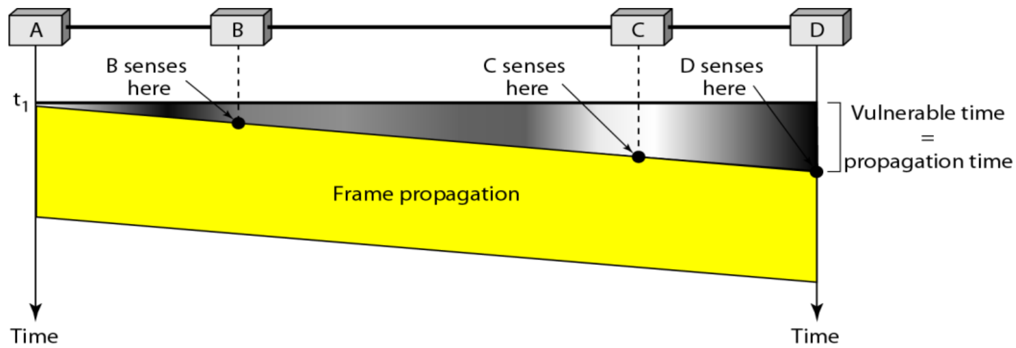
--At time t_1 station B senses the medium and finds it idle, so it sends a frame. At time t_2 $(t_2 > t_1)$, station C senses the medium and finds it idle because, at this time, the first bits from station B have not reached station C. Station C also sends a frame. The two signals collide and both frames 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 12.9 *Vulnerable time in CSMA*



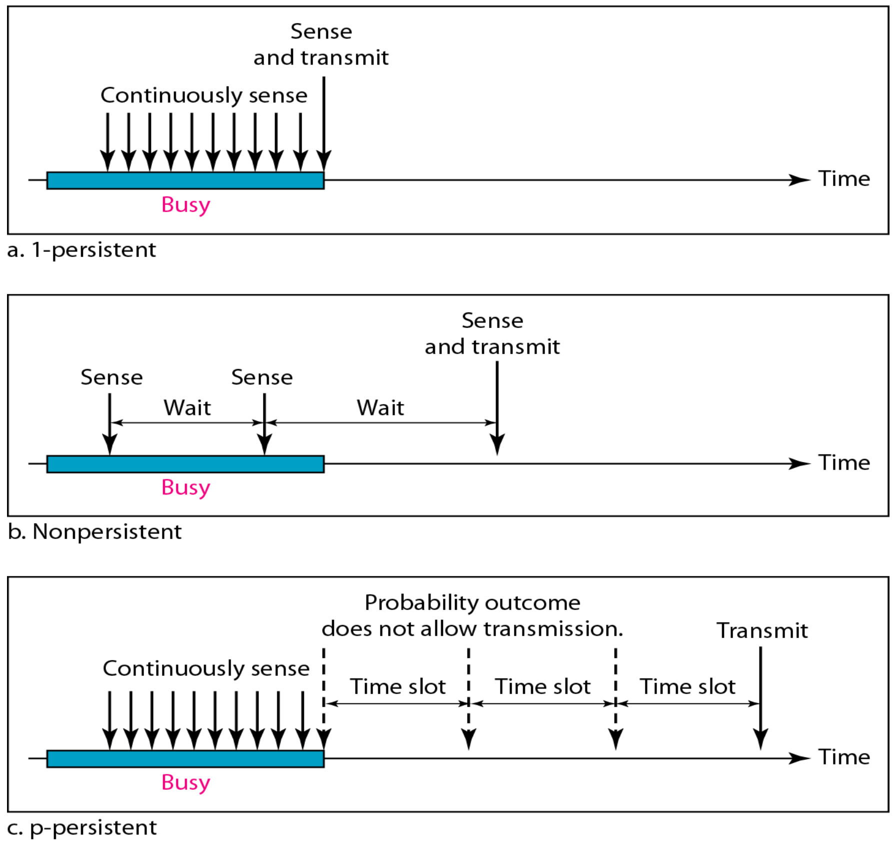
--Fig. shows the worst case. The leftmost station A sends a frame at time t_1, which reaches the rightmost station D at time $t_1 + T_p$. The gray area shows the vulnerable area in time and space.

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.

Figure 12.10 *Behavior 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.

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.

--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 slot duration equal to or greater than the maximum propagation time.

--The p-persistent approach combines the advantages of the other two strategies. It reduces the chance of collision and improves efficiency.

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

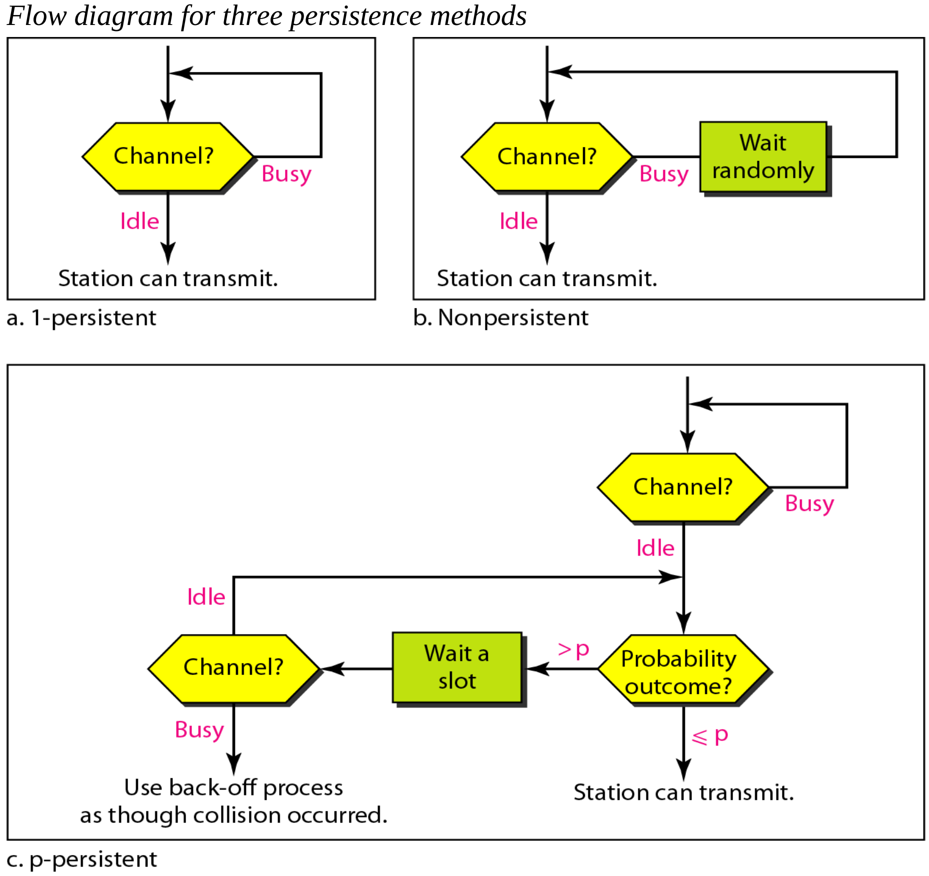
1. With probability p, the station sends its frame.

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

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

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

Figure 12.11 *Flow diagram for three persistence methods*



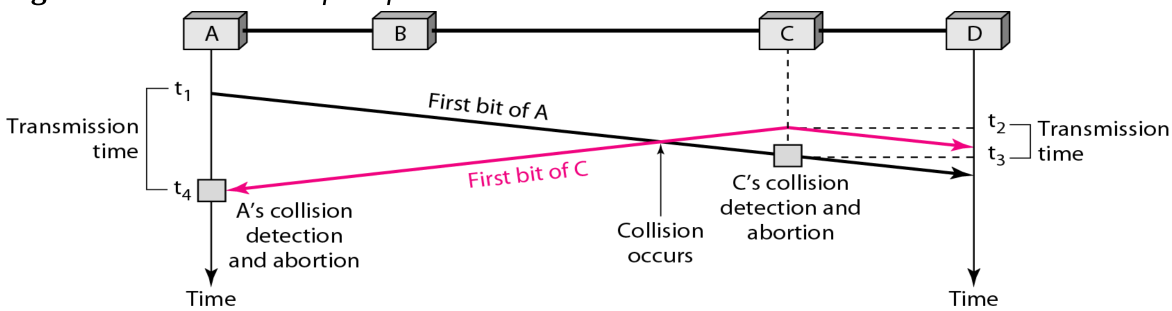
Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

--The CSMA method does not specify the procedure following a collision. CSMA/CD augments the algorithm to handle the collision.

--A station monitors the medium after it sends a frame to see if the transmission was successful. If so, the station is finished. If there is a collision, the frame is sent again.

--To better understand CSMA/CD, 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 Fig, stations A & C are collided.

Figure 12.12 *Collision of the first bit 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.

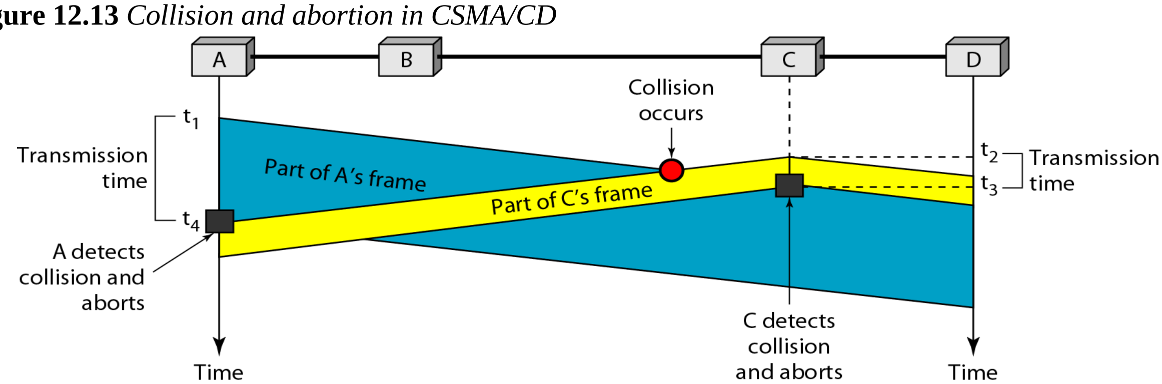
--Station C immediately (or after a short time, but we assume immediately) aborts transmission.

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

--A transmits for the duration $t_4 - t_1$,C transmits for the duration $t_3 - t_2$, for the protocol to work, the length of any frame divided by the bit rate 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 B's frame, though incomplete, is aborted.

Figure 12.13 *Collision and abortion in CSMA/CD*



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.

--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.

--In 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 $2T_p$.

Example 12.5

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 25.611S, what is the minimum size of the frame?

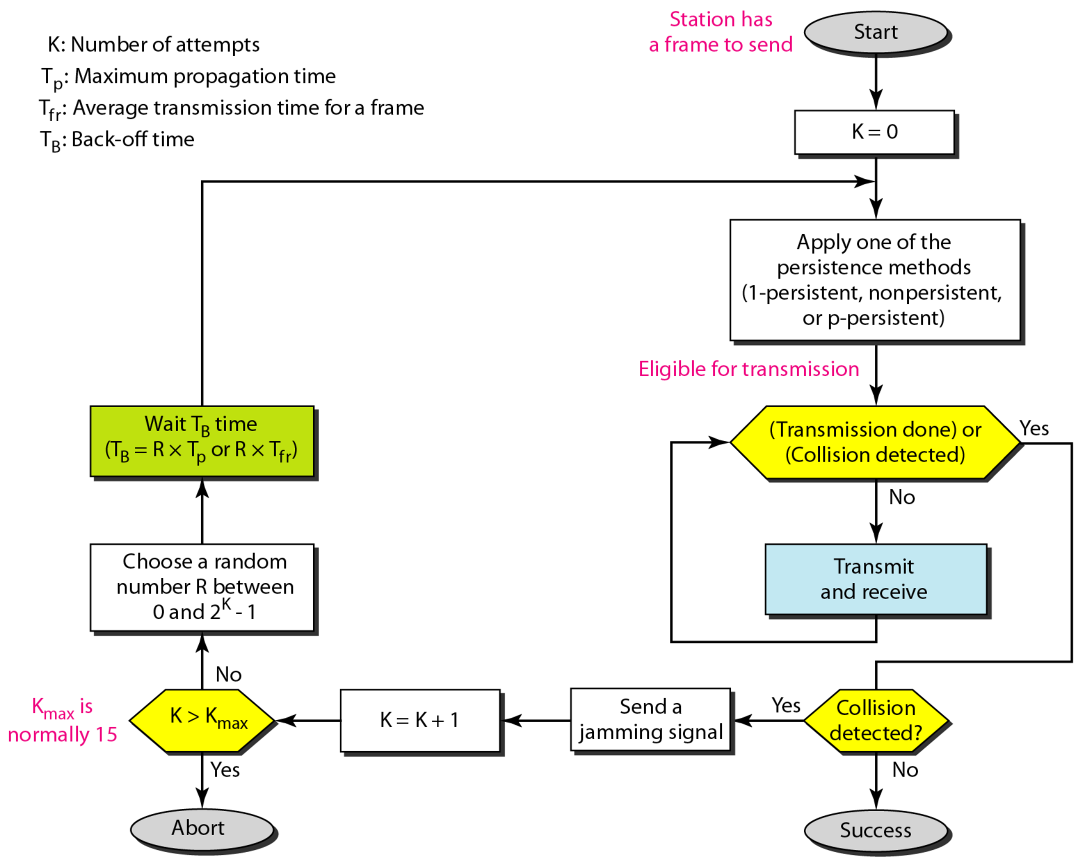
Solution

The frame transmission time is T_{fr} = 2 x T_p =51.2 μs. This means, in the worst case, a station needs to transmit for a period of 51.2 μs to detect the collision. The minimum size of the frame is 10 Mbps x 51.2 μs =512 bits or 64 bytes. This is actually the minimum size of the frame for Standard Ethernet.

Procedure

--The flow diagram for *CSMA/CD* is similar to the one for the ALOHA protocol, but there are differences.

Figure 12.14 *Flow diagram for the CSMA/CD*



--The first difference is the addition of the persistence process. We sense the channel before start sending the frame by using one of the persistence processes (nonpersistent, 1-persistent, or p-persistent). The corresponding box can be replaced by one of the persistence processes.

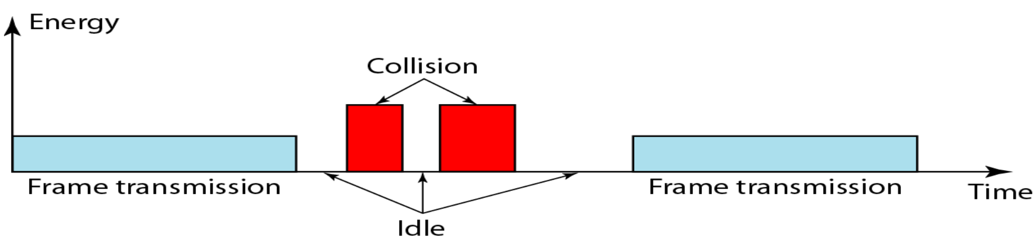
--The second difference is the frame transmission. In ALOHA, we first transmit the entire frame and then wait for an acknowledgment. In *CSMA/CD*, transmission and collision detection is a continuous process. We do not send the entire frame and then look for a collision.

--The station transmits and receives continuously and simultaneously (using two different ports).
--A loop is used to show that transmission 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.
--When we come 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.
--The third difference is the sending of a short jamming signal that enforces the collision in case other stations have not yet sensed the collision.

Energy Level

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

Figure 12.15 Energy *level during transmission, idleness, or collision*



Throughput

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

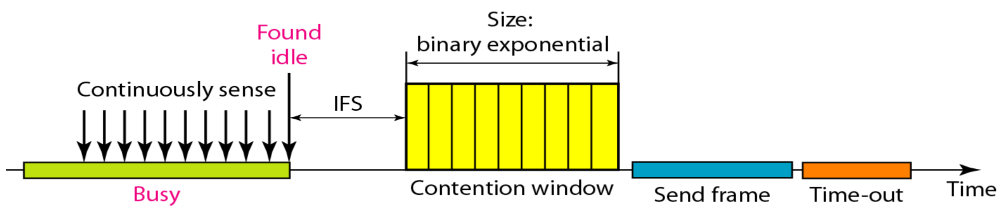
Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)

--The basic idea behind *CSMA/CD* is that a station needs to be able to receive while transmitting to detect a collision.
--When there is no collision, the station receives one signal: its own signal. When there is a collision, the station receives two signals: its own signal and the signal transmitted by a second station.
--To distinguish between these two cases, the received signals in these two cases must be significantly different. In other words, the signal from the second station needs to add a significant amount of energy to the one created by the first station.
--In a wired network, the received signal has almost the same energy as the sent signal because either the length of the cable is short or there are repeaters that amplify the energy between the sender and the receiver. This means that in a collision, the detected energy almost doubles.
--However, in a wireless network, much of the sent energy is lost in transmission. The received signal has very little energy. Therefore, a collision may add only 5 to 10 percent additional energy. This is not useful for effective collision detection.

--We need to avoid collisions on wireless networks because they cannot be detected.

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

Figure 12.16 *Timing in CSMA/CA*



Interframe Space (IFS)

--First, collisions are avoided by deferring transmission even if the channel is found idle.

--When an idle channel is found, the station does not send immediately. It waits for a period of time called the interframe space or IFS. Even though the channel may appear idle when it is sensed, a distant station may have already started transmitting. The distant station's signal has not yet reached this station.

--The IFS time allows the front of the transmitted signal by the distant station to reach this station. If after the IFS time the channel is still idle, the station can send, but it still needs to wait a time equal to the contention time. The IFS variable can also be used to prioritize stations or frame types.

--For example, a station that is assigned 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. If the station finds the channel busy, it does not restart the process; it just stops the timer and restarts it when the channel is sensed as idle. This gives priority to the station with the longest waiting time.

Acknowledgment

--With all these precautions, there still may be a collision resulting in destroyed data.

--In addition, the data may be corrupted during the transmission. The positive acknowledgment and the time-out timer can help guarantee that the receiver has received the frame.

Procedure

--The channel needs to be sensed before and after the IFS. The channel also needs to be sensed during the contention time.

--For each time slot of the contention window, the channel is sensed. If it is found idle, the timer continues; if the channel is found busy, the timer is stopped and continues after the timer becomes idle again.

CSMA/CA and Wireless Networks

--*CSMA/CA* was mostly intended for use in wireless networks.

Figure 12.17 *Flow diagram for CSMA/CA*