DATA LINK CONTROL

FRAMING

- Framing in the data-link layer separates a message from one source to a destination by adding a sender address and a destination address.
- The destination address defines where the packet is to go, the sender address helps the recipient acknowledge the receipt.

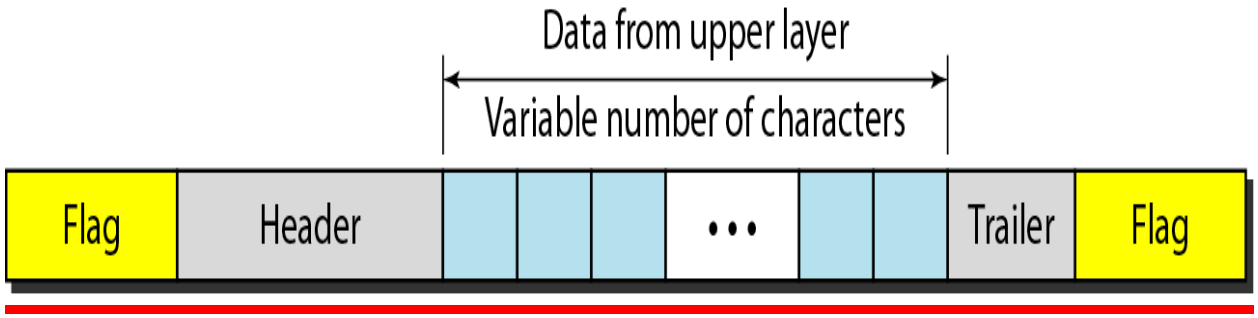
FRAME SIZE

- Frames can be of fixed or variable size.
- In fixed-size framing, there is no need for defining the boundaries of the frames, the size itself can be used as a delimiter.
- In variable-size framing, we need a way to define the end of one frame and the beginning of the next.
- The two approaches were used for this purpose:
    - Character-oriented approach and
    - Bit-oriented approach.

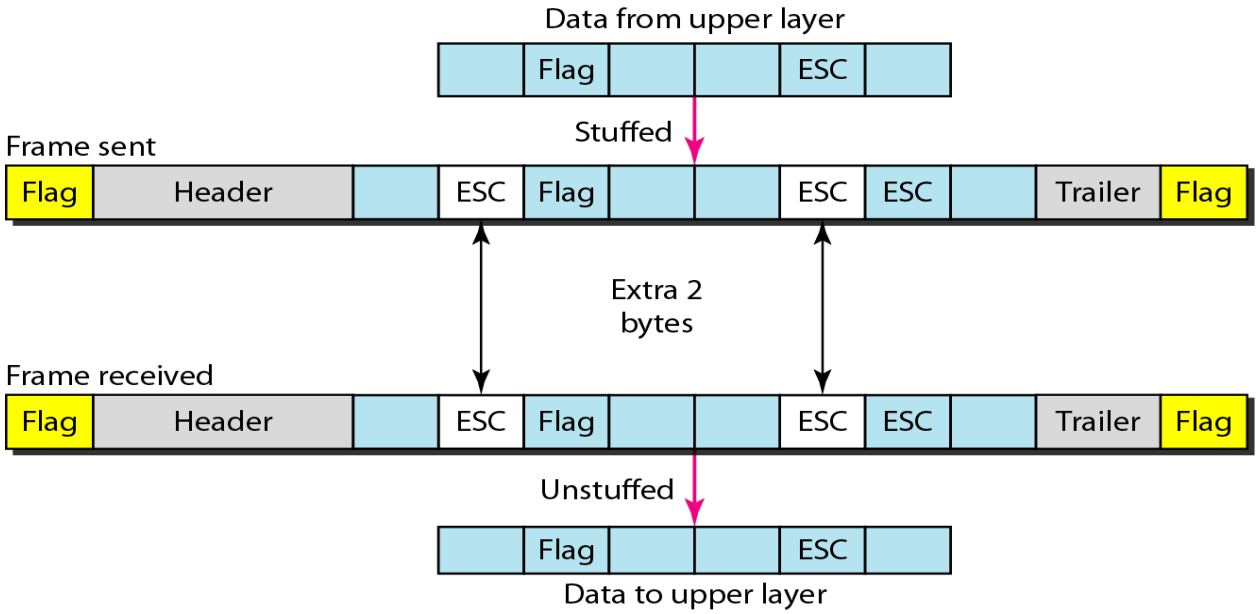
character-oriented approach

- In character-oriented (or byte-oriented) framing, data to be carried are 8-bit characters from a coding system such as ASCII.
- The header, which normally carries the source and destination addresses and other control information, and the trailer, which carries error detection redundant bits, are also multiples of 8 bits.
- To separate one frame from the next, an 8-bit (1-byte) flag is added at the beginning and the end of a frame.
- The flag, composed of protocol-dependent special characters.

Figure A frame in a character-oriented protocol

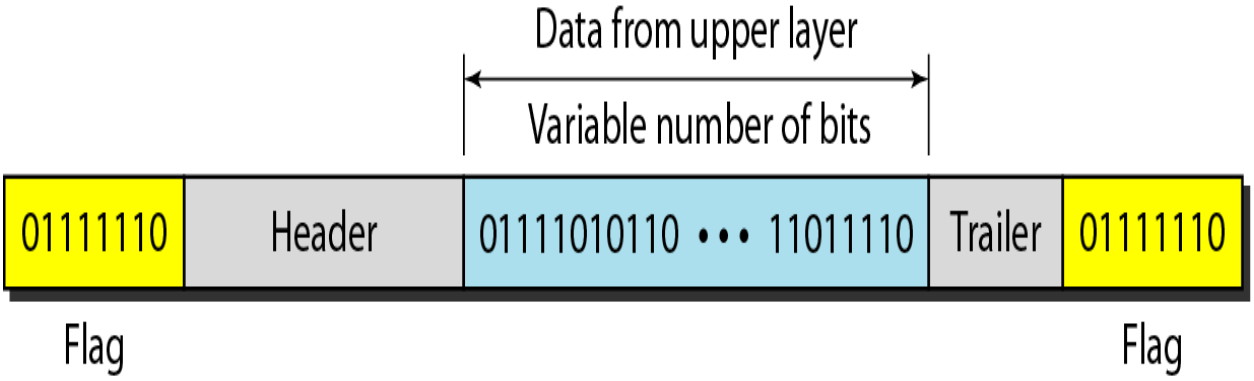


- Here character used for the flag could also be part of the information.
- If this happens, the receiver, when it encounters this pattern in the middle of the data, thinks it has reached the end of the frame.
- To fix this problem, a byte-stuffing strategy was added to character-oriented framing.
- In byte stuffing (or character stuffing), a special byte is added to the data section of the frame when there is a character with the same pattern as the flag. The data section is stuffed with an extra byte.

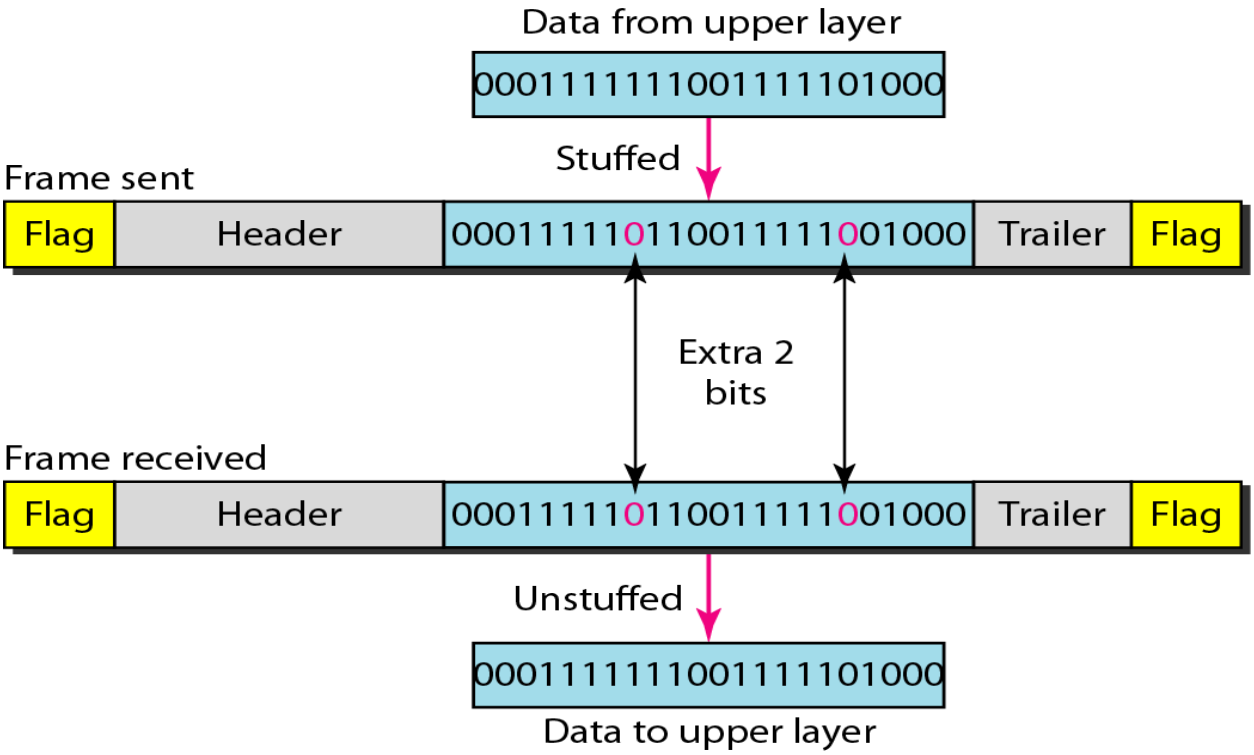


Bit-oriented approach.

- In bit-oriented framing, the data section of a frame is a sequence of bits to be interpreted by the upper layer as text, graphic, audio, video, and so on.
- However, in addition to headers (and possible trailers), we still need a delimiter to separate one frame from the other.
- Most protocols use a special 8-bit pattern flag, 01111110, as the delimiter to define the beginning and the end of the frame.

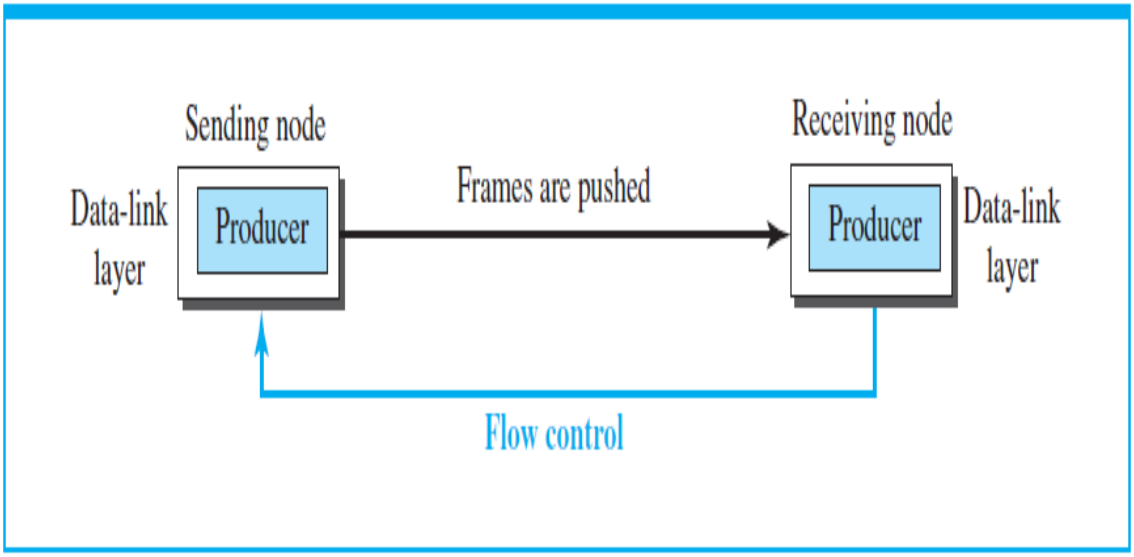


- if the flag pattern appears in the data.
- Inform the receiver that this is not the end of the frame.
- By stuffing 1 single bit (instead of 1 byte) to prevent the pattern from looking like a flag. The strategy is called bit stuffing.
- In bit stuffing, if a 0 and five consecutive 1 bits are encountered, an extra 0 is added.
- This extra stuffed bit is eventually removed from the data by the receiver.
- The extra bit is added after one 0 followed by five 1s regardless of the value of the next bit.
- This guarantees that the flag field sequence does not inadvertently appear in the frame.



FLOW AND ERROR CONTROL

- The most important responsibilities of the data link layer are flow control and error control. Collectively, these functions are known as data link control.
- Flow control refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgment.
- Error control in the data link layer is based on automatic repeat request, which is the retransmission of data.
- In communication at the data-link layer, we are dealing with four entities: network and data-link layers at the sending node and network and data-link layers at the receiving node.



- The figure shows that the data-link layer at the sending node tries to push frames toward the data-link layer at the receiving node.
- If the receiving node cannot process and deliver the packet to its network at the same rate that the frames arrive, it becomes overwhelmed with frames.

Buffers

- Although flow control can be implemented in several ways, one of the solutions is normally to use two buffers one at the sending data-link layer and the other at the receiving data-link layer.
- A buffer is a set of memory locations that can hold packets at the sender and receiver.
- The flow control communication can occur by sending signals from the consumer to the producer.
- When the buffer of the receiving data-link layer is full, it informs the sending data-link layer to stop pushing frames.

Error Control

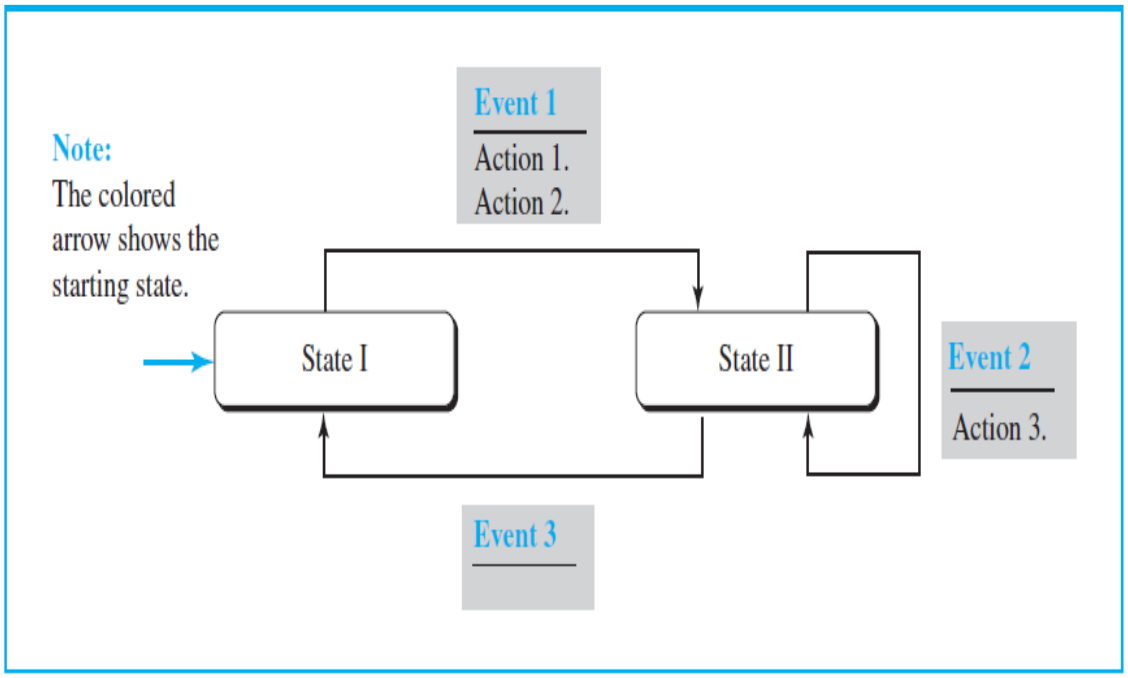
- Error control is required at the data-link layer to prevent the receiving node from delivering corrupted packets to its network layer.
- Error control at the data-link layer is normally very simple and implemented using one of the following two methods.
- In both methods, a CRC is added to the frame header by the sender and checked by the receiver.

1. In the first method, if the frame is corrupted, it is silently discarded, if it is not corrupted, the packet is delivered to the network layer.
2. In the second method, if the frame is corrupted, it is silently discarded, if it is not corrupted, an acknowledgment is sent (for the purpose of both flow and error control) to the sender

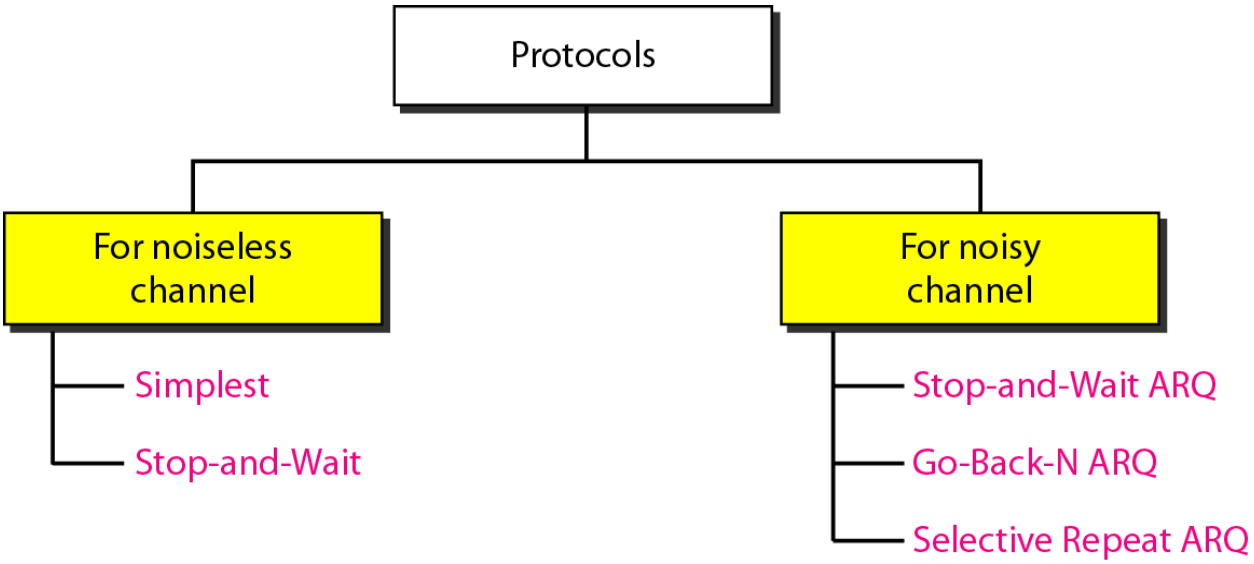
DATA-LINK LAYER PROTOCOLS

- The behavior of a data-link-layer protocol shown as a finite state machine (FSM).
- An FSM is a machine with a finite number of states.
- The machine is always in one of the states until an event occurs.
- Each event is associated with two reactions:
    - defining the list (possibly empty) of actions to be performed and determining the next state.
    - One of the states must be defined as the initial state, the state in which the machine starts when it turns on.
- The figure shows a machine with three states.
- There are only three possible events and three possible actions.
- The machine starts in state I. If event 1 occurs, the machine performs actions 1 and 2 and moves to state II.
- When the machine is in state II, two events may occur.

- If event 1 occurs, the machine performs action 3 and remains in the same state, state II.
- If event 3 occurs, the machine performs no action, but move to state I.

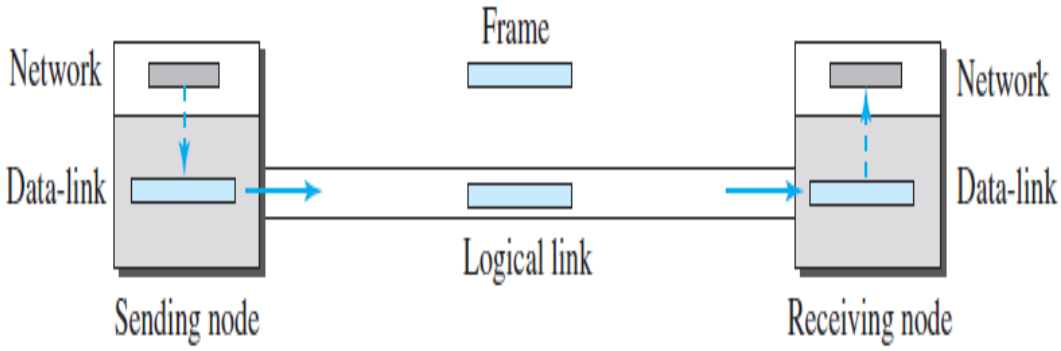


DATA-LINK LAYER PROTOCOLS



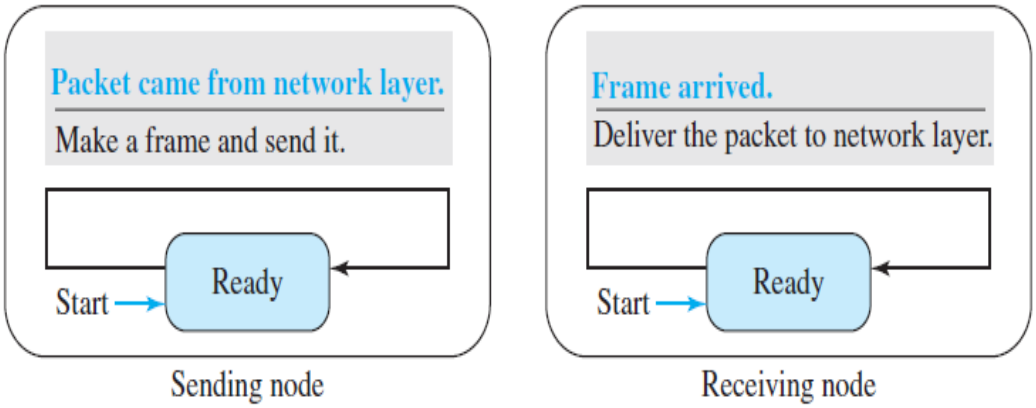
Simple Protocol

- A simple protocol with neither flow nor error control.
- The receiver can immediately handle any frame it receives.
- The receiver can never be overwhelmed with incoming frames.
- The data-link layer at the sender gets a packet from its network layer, makes a frame out of it, and sends the frame.
- The data-link layer at the receiver receives a frame from the link, extracts the packet from the frame, and delivers the packet to its network layer.
- The data-link layers of the sender and receiver provide transmission services for their network layers.

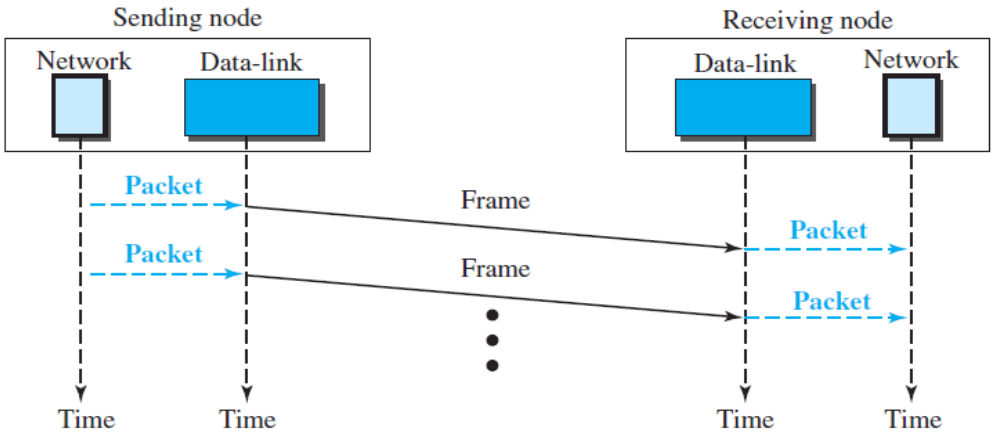


FSMs

- The sender site should not send a frame until its network layer has a message to send.
- The receiver site cannot deliver a message to its network layer until a frame arrives.
- This is using two FSMs. Each FSM has only one state, the ready state.
- The sending machine remains in the ready state until a request comes from the process in the network layer.
- When this event occurs, the sending machine encapsulates the message in a frame and sends it to the receiving machine.
- The receiving machine remains in the ready state until a frame arrives from the sending machine.
- When this event occurs, the receiving machine decapsulates the message out of the frame and delivers it to the process at the network layer.

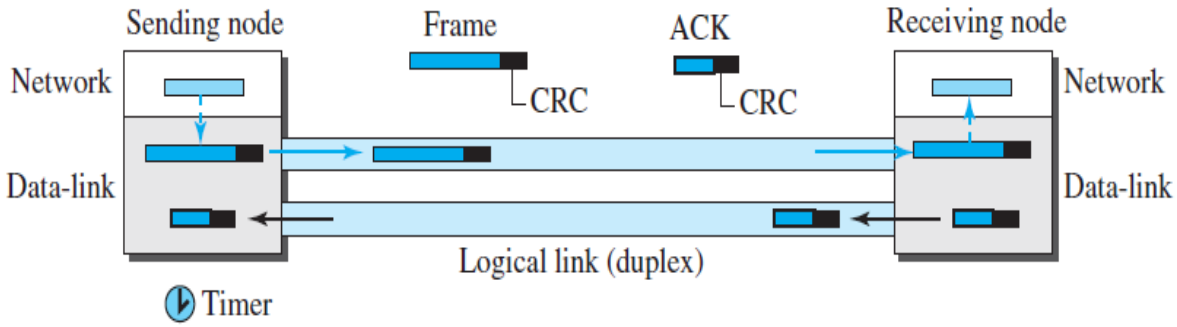


- Below Figure shows an example of communication using this protocol.
- It is very simple.
- The sender sends frames one after another without even thinking about the receiver.



Stop-and-Wait Protocol

- Stop-and-Wait protocol, which uses both flow and error control.
- In this protocol, the sender sends one frame at a time and waits for an acknowledgment before sending the next one.
- To detect corrupted frames, add a CRC to each data frame.
- If its CRC is incorrect, the frame is corrupted and silently discarded.
- The silence of the receiver is a signal for the sender that a frame was either corrupted or lost.
- Every time the sender sends a frame, it starts a timer.
- If an acknowledgment arrives before the timer expires, the timer is stopped and the sender sends the next frame (if it has one to send).
- If the timer expires, the sender resends the previous frame, assuming that the frame was either lost or corrupted.
- This means that the sender needs to keep a copy of the frame until its acknowledgment arrives.
- When the corresponding acknowledgment arrives, the sender discards the copy and sends the next frame if it is ready



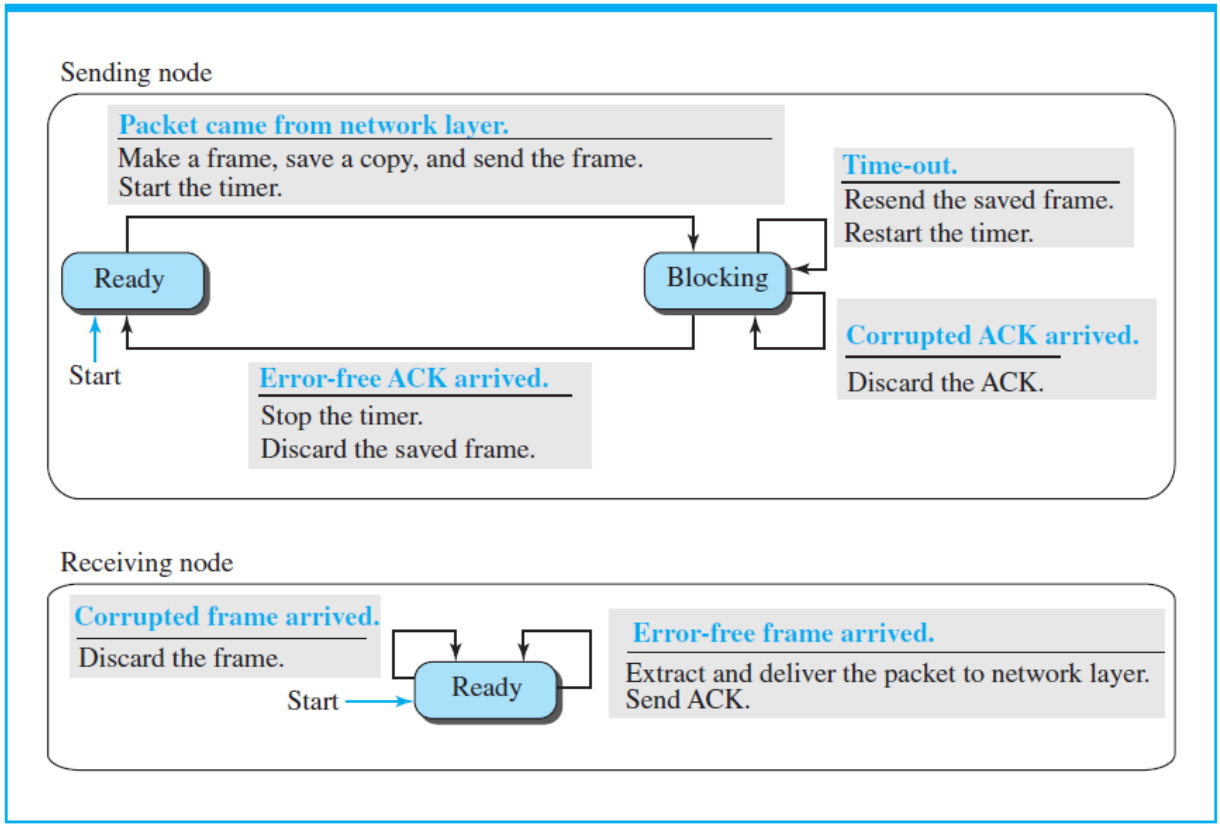
FSM

Sender States

- The sender is initially in the ready state, but it can move between the ready and blocking state.

Ready State.

- When the sender is in this state, it is only waiting for a packet from the network layer.
- If a packet comes from the network layer, the sender creates a frame, saves a copy of the frame, starts the timer and sends the frame.
- The sender then moves to the blocking state.



Blocking State.

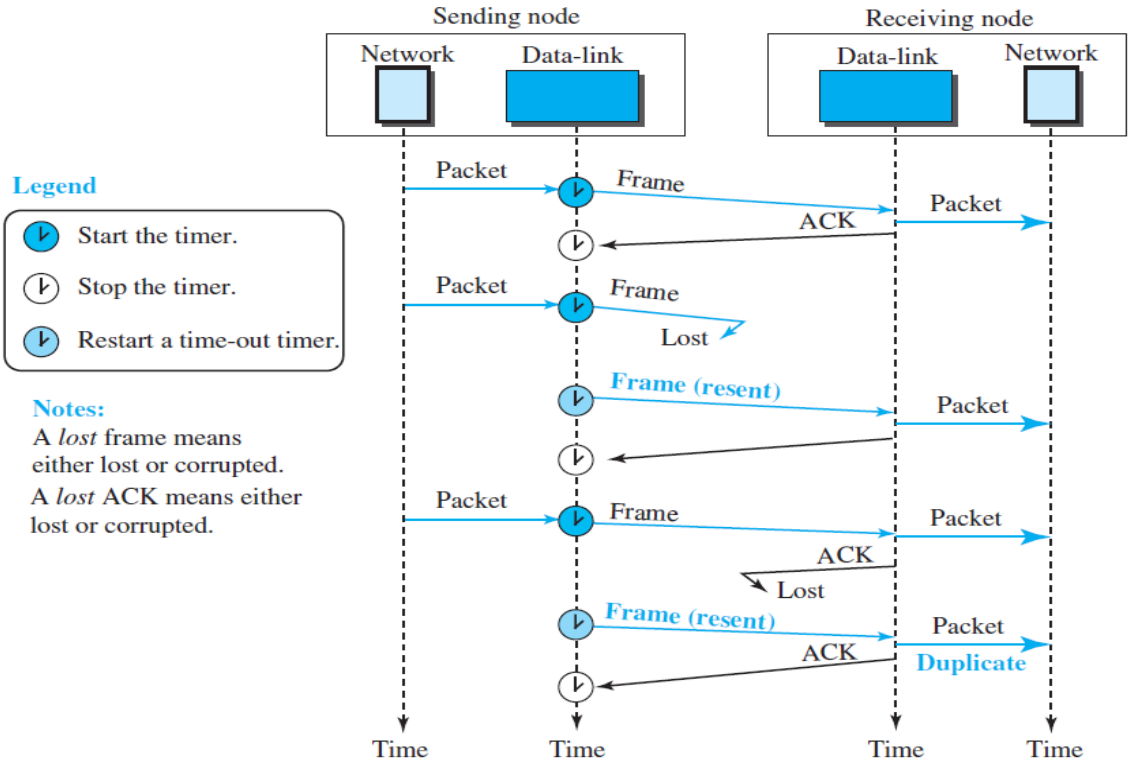
- When the sender is in this state, three events can occur:
- If a time-out occurs, the sender resends the saved copy of the frame and restarts the timer.
- If a corrupted ACK arrives, it is discarded.
- If an error-free ACK arrives, the sender stops the timer and discards the saved copy of the frame. It then moves to the ready state.

Receiver

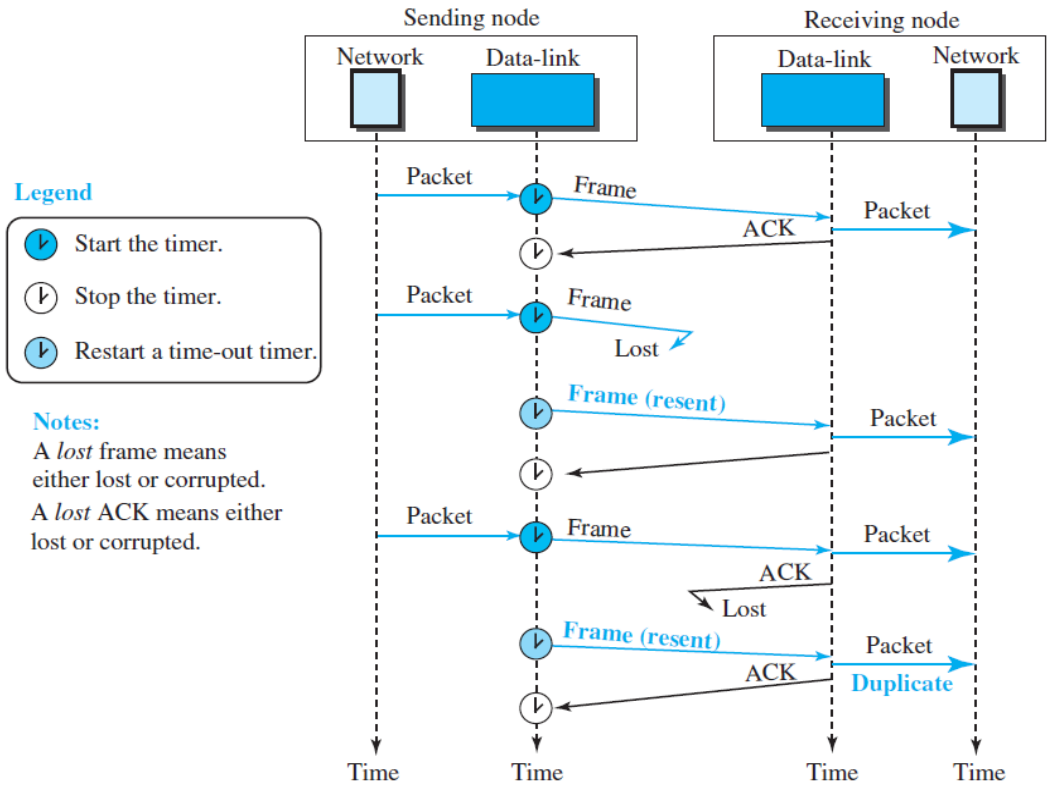
- The receiver is always in the ready state.
- Two events may occur:
  1. If an error-free frame arrives, the message in the frame is delivered to the network layer and an ACK is sent.
  2. If a corrupted frame arrives, the frame is discarded

Stop-and-Wait Protocol-Example

- The first frame is sent and acknowledged.
- The second frame is sent, but lost.
- After time-out, it is resent.
- The third frame is sent and acknowledged, but the acknowledgment is lost. The frame is resent.
- However, there is a problem with this scheme.
- The network layer at the receiver site receives two copies of the third packet, which is not right.
- To avoid unordering and duplication add sequence numbers to the data frames and acknowledgment numbers to the ACK frames.
- However, numbering in this case is very simple. Sequence numbers are 0, 1, 0, 1, 0, 1, . .



- The acknowledgment numbers can also be 1, 0, 1, 0, 1, 0, …
- In other words, the sequence numbers start with 0, the acknowledgment numbers start with 1.
- An acknowledgment number always defines the sequence number of the next frame to receive
- Adding sequence numbers and acknowledgment numbers can prevent duplicates.
- The first frame is sent and acknowledged.
- The second frame is sent, but lost.
- After time-out, it is resent.
- The third frame is sent and acknowledged, but the acknowledgment is lost.
- The frame is resent.

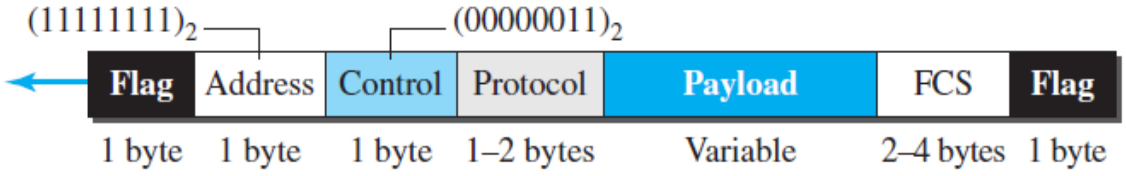


POINT-TO-POINT PROTOCOL (PPP)

- One of the most common protocols for point-to-point access is the Point-to-Point Protocol (PPP).
- Internet users who need to connect their home computers to the server of an Internet service provider use PPP.
- The majority of these users have a traditional modem, they are connected to the Internet through a telephone line, which provides the services of the physical layer.
- But to control and manage the transfer of data, there is a need for a point-to-point protocol at the data-link layer.

Framing

- PPP uses a character-oriented (or byte-oriented) frame. Below Figure shows the format of a PPP frame.
- The description of each field follows
  - **Flag:-** A PPP frame starts and ends with a 1-byte flag with the bit pattern 01111110.
  - **Address.** The address field in this protocol is a constant value and set to 11111111 .
  - **Control.** This field is set to the constant value 00000011
    - ✓ PPP does not provide any flow control.
    - ✓ Error control is also limited to error detection.
  - **Protocol.** The protocol field defines what is being carried in the data field, either user data or other information. This field is by default 2 bytes long, but the two parties can agree to use only 1 byte.
  - **Payload field.** This field carries either the user data or other information. The data field is a sequence of bytes with the default of a maximum of 1500 bytes, but this can be changed during negotiation. The data field is byte-stuffed if the flag byte pattern appears in this field.
  - **FCS.** The frame check sequence (FCS) is simply a 2-byte or 4-byte standard CRC

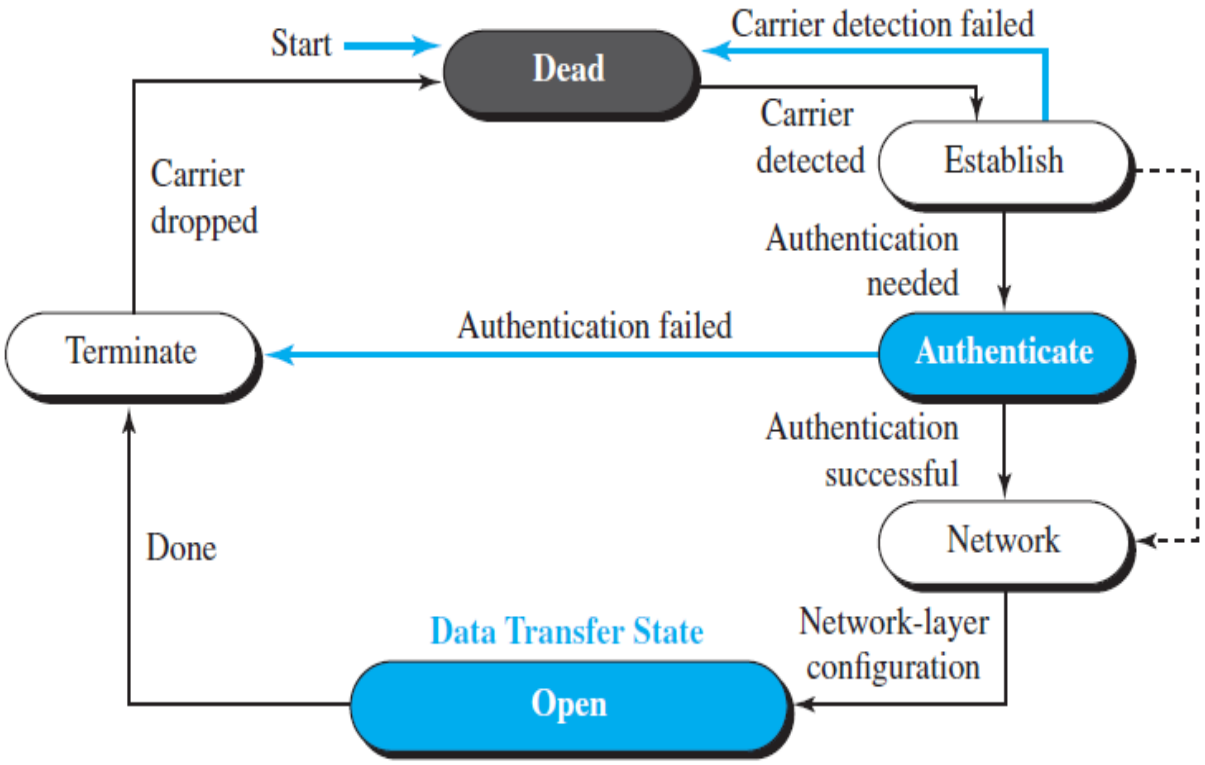


Byte Stuffing

- PPP is a byte-oriented protocol, the flag in PPP is a byte that needs to be escaped whenever it appears in the data section of the frame.
- The escape byte is 01111101, which means that every time the flaglike pattern appears in the data, this extra byte is stuffed to tell the receiver that the next byte is not a flag.

Transition Phases

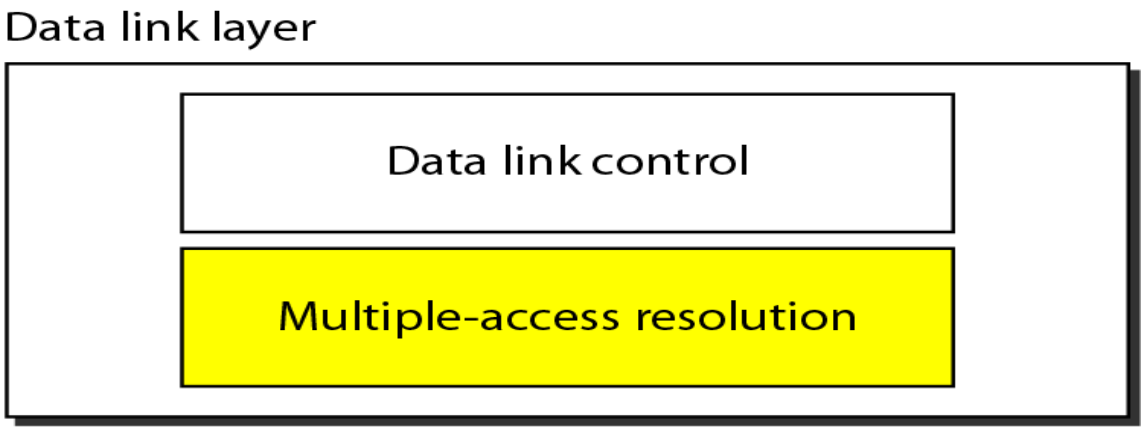
- A PPP connection goes through phases which can be shown in a transition phase diagram shown below figure.
- The transition diagram, which is an FSM, starts with the dead state. In this state, there is no active carrier.
- When one of the two nodes starts the communication, the connection goes into the establish state. In this state, options are negotiated between the two parties.
- If the two parties agree that they need authentication, then the system needs to do authentication, otherwise, the parties can simply start communication.
- Data transfer takes place in the open state. When a connection reaches this state, the exchange of data packets can be started.
- The connection remains in this state until one of the endpoints wants to terminate the connection.
- In this case, the system goes to the terminate state.
- The system remains in this state until the carrier is dropped, which moves the system to the dead state again.


Start
Dead
Carrier detection failed
Carrier
detected
Establish
Carrier
dropped
Authentication
needed
Terminate
Authentication failed
Authenticate
Authentication
successful
Done
Network
Data Transfer State
Network-layer
configuration
Open

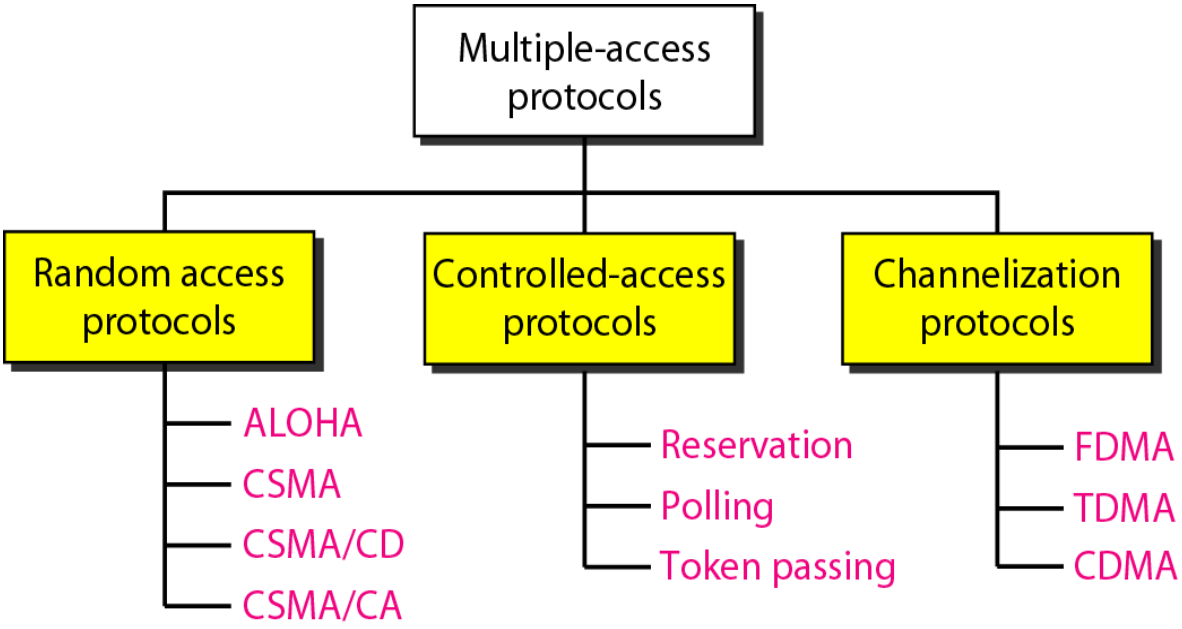
MEDIA ACCESS CONTROL

INTRODUCTION

- Data link layer divided into two functionality-oriented sublayers as shown below figure



Categories of MAC are as follows

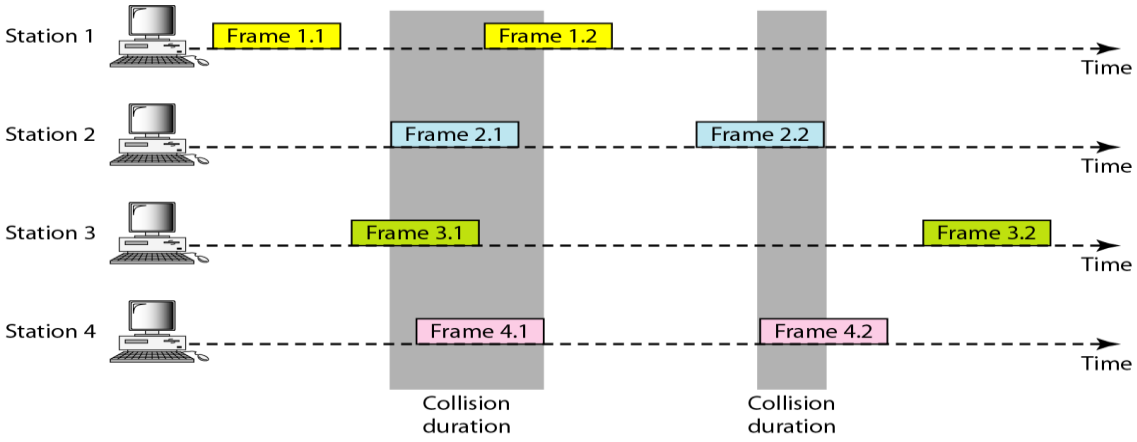


RANDOM ACCESS

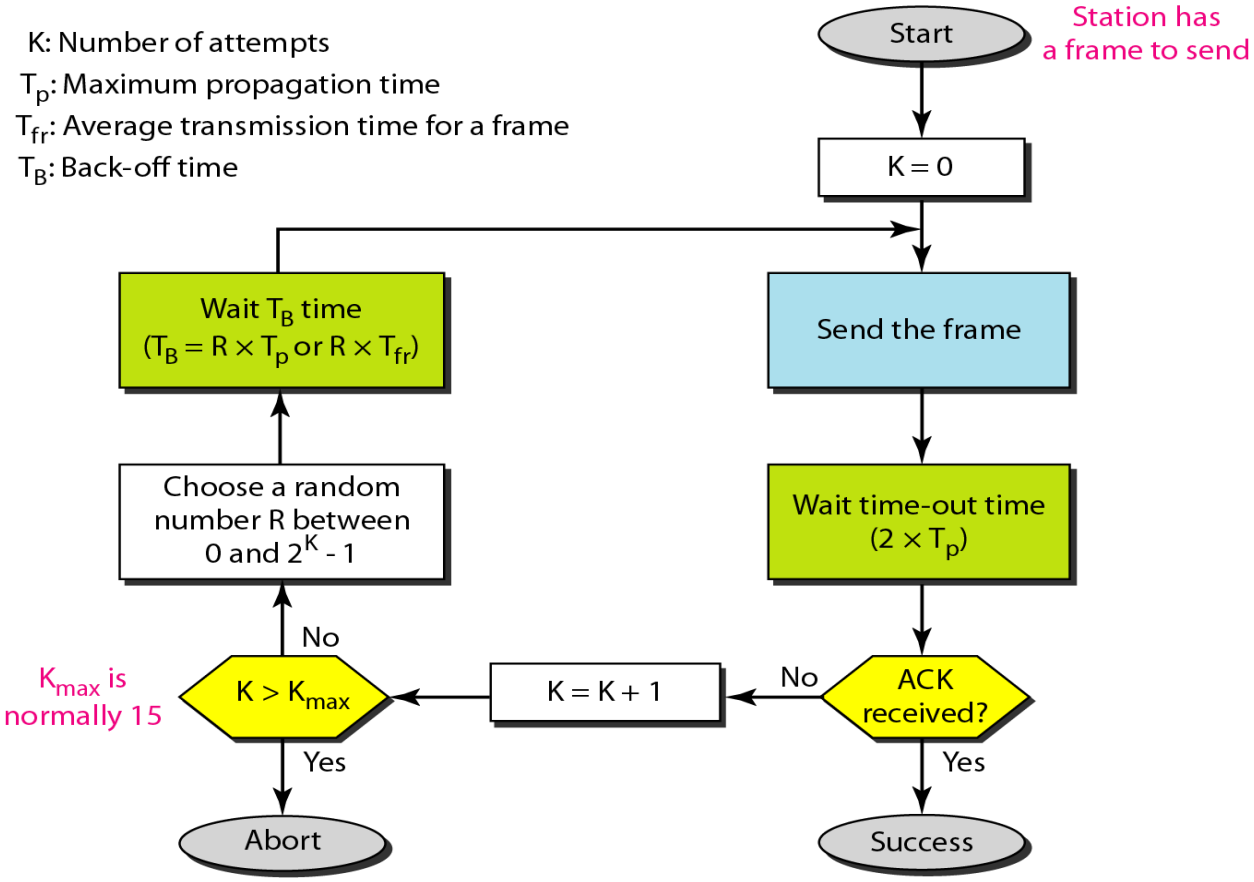
- In random access or contention methods, no station is superior to another station and none is assigned the control over another.
- No station permits, or does not permit, another station to send.
- At each instance, a station that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send.

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



- There are four stations (unrealistic assumption) that contend with one another for access to the shared channel.
- The figure above 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 above shows that only two frames survive:
  - ✓ one frame from station 1 and
  - ✓ one frame from station 3.
- We need to mention that even if one bit of a frame coexists on the channel with one bit from another frame, there is a collision and both will be destroyed.
- 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 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 the backoff time TB.
- Pure ALOHA used to prevent congesting the channel with retransmitted frames.
- After a maximum number of retransmission attempts Kmax, a station must give up and try later.
- Figure below shows the procedure for pure ALOHA.



- 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 formula for T_B depends on the implementation. One common formula is the binary exponential backoff.
- In this method, for each retransmission, a multiplier R = 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 value of K_{max} is usually chosen as 15.

Example

The stations on a wireless ALOHA network are a maximum of 600 km apart. Assume that signals propagate at 3×10^8 m/s, Then

$$Tp = (6\times 10^5) / (3 \times 10^8) = 2 \text{ ms.}$$

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 × 2) or 2 ms (1 × 2), based on the outcome of the random variable.

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

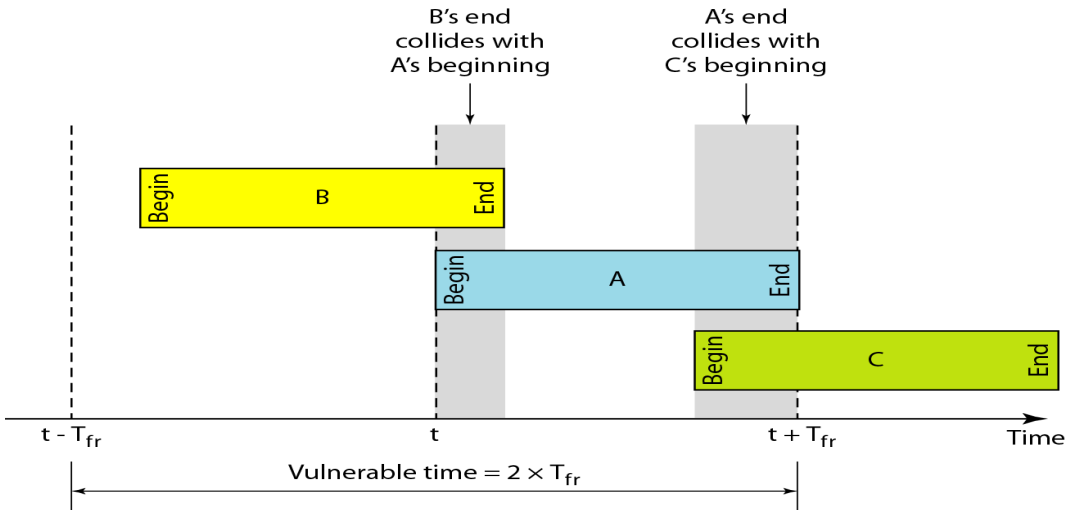
c. For K = 3, the range is {0, 1, 2, 3, 4, 5, 6, 7}. This means that TB can be 0, 2, 4, . . . , 14 ms, based on the outcome of the random variable.

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

Vulnerable time

- Is the length of time in which there is a possibility of collision.
- Assume that the stations send fixed-length frames with each frame taking Tfr seconds to send. Figure below shows the vulnerable time for station B.
- Station B starts to send a frame at time t.
- Imagine station A has started to send its frame after t − Tfr. 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 + Tfr. Here, there is also a collision between frames from station B and station C. Shown below figure.
- The vulnerable time during which a collision may occur in pure ALOHA is 2 times the frame transmission time

Pure ALOHA vulnerable time = 2 X Tfr



Example

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 2×1 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

- Assume G is the average number of frames generated by the system during one frame transmission time.
- The average number of successfully transmitted frames for pure ALOHA is $S = G \times e^{-2G}$.
- The maximum throughput Smax is 0.184, for G = ½

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

The maximum throughput

S_{max} = 0.184 when G= (1/2).

Example

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.

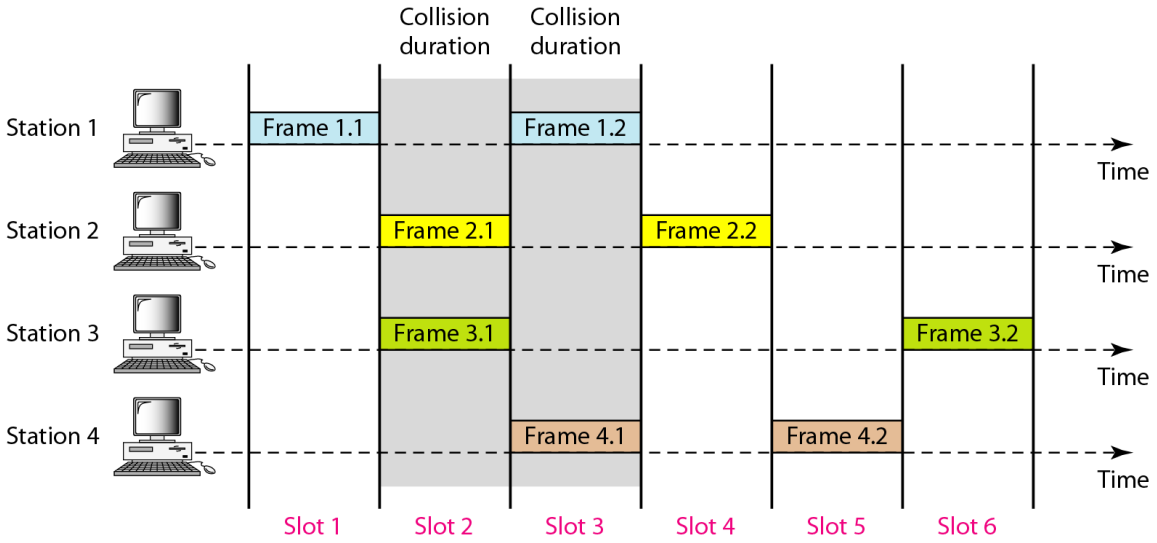
a. If the system creates 1000 frames per second, this is 1 frame per millisecond. The load is 1. In this case $S = G \times e^{-2G}$ or S = 0.135 (13.5 percent). This means that the throughput is 1000 × 0.135 = 135 frames. Only 135 frames out of 1000 will probably survive.

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

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

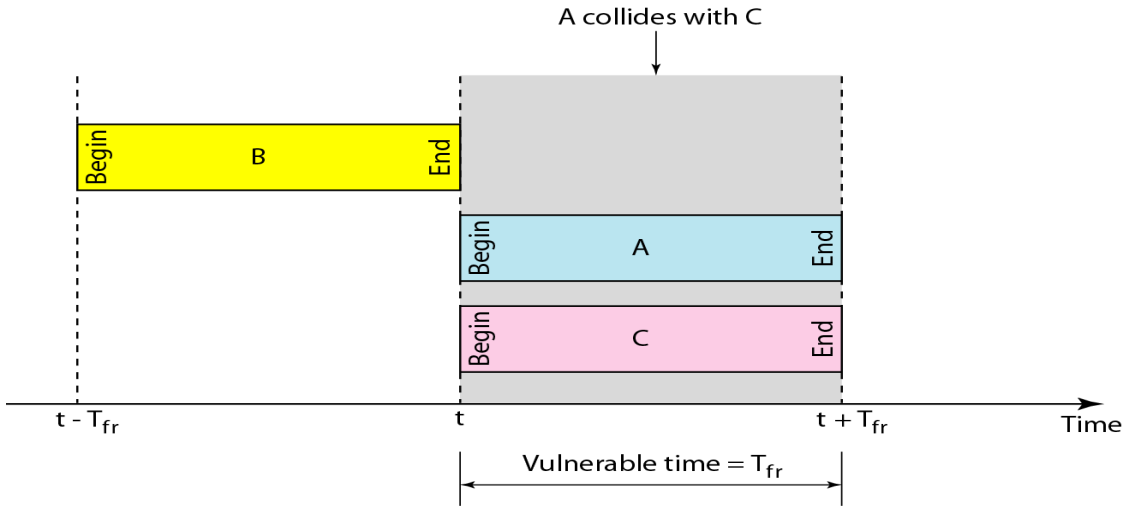
SLOTTED ALOHA

- Slotted ALOHA was invented to improve the efficiency of pure ALOHA.
- In slotted ALOHA, divide the time into slots of Tfr seconds and force the station to send only at the beginning of the time slot.
- Below Figure shows an example of frame collisions in slotted ALOHA
- Because a station is allowed to send only at the beginning of the synchronized time slot, if a station misses this moment, it must wait until the beginning of the next time slot.
- There is still the possibility of collision if two stations try to send at the beginning of the same time slot



Vulnerable time

- The vulnerable time is now reduced to one-half, equal to Tfr shown in Figure



Throughput

The throughput for slotted ALOHA is

$$S = G \times e^{-G}.$$

The maximum throughput

$S_{max} = 0.368$ when G = 1.

Example

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

a. **1000 frames per second**

b. **500 frames per second**

c. **250 frames per second.**

d. ***Solution***

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

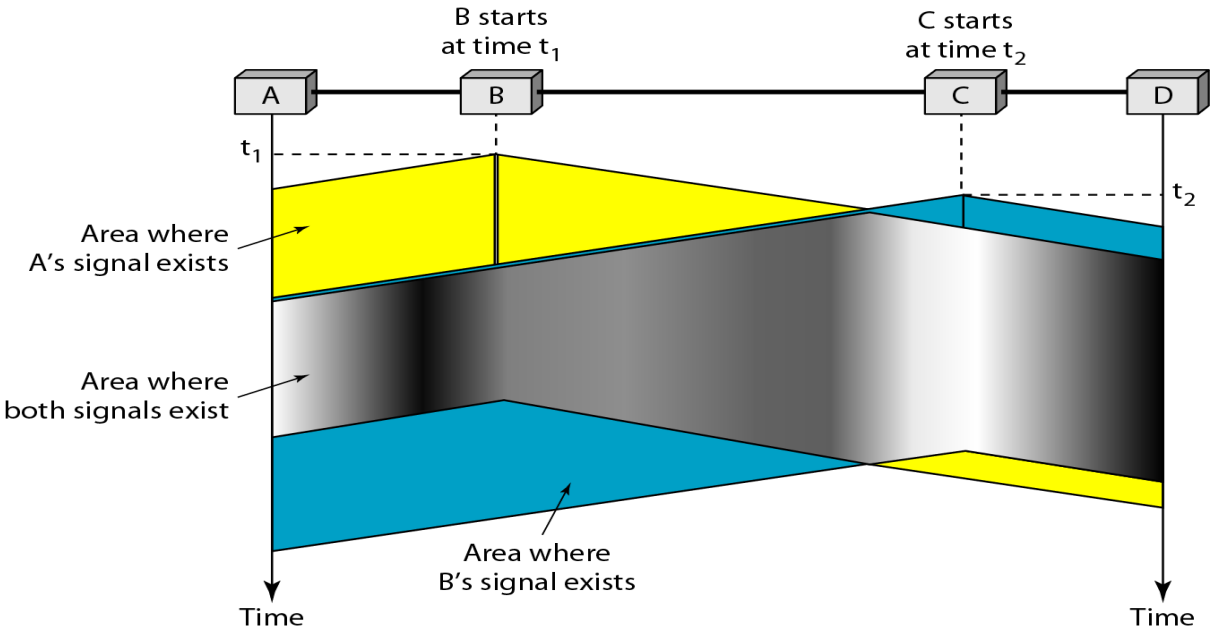
a. If the system creates 1000 frames per second, this is 1 frame per millisecond. The load is 1. In this case $S = G \times e^{-G}$ or S = 0.368 (36.8 percent). This means that the throughput is 1000 × 0.0368 = 368 frames. Only 386 frames out of 1000 will probably survive

b. If the system creates 500 frames per second, this is (1/2) frame per millisecond. The load is (1/2). In this case $S = G \times e^{-G}$ or S = 0.303 (30.3 percent). This means that the throughput is 500 × 0.0303 = 151. Only 151 frames out of 500 will probably survive.

c. If the system creates 250 frames per second, this is (1/4) frame per millisecond. The load is (1/4). In this case $S = G \times e^{-G}$ or S = 0.195 (19.5 percent). This means that the throughput is 250 × 0.195 = 49. 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. Shown in below Figure



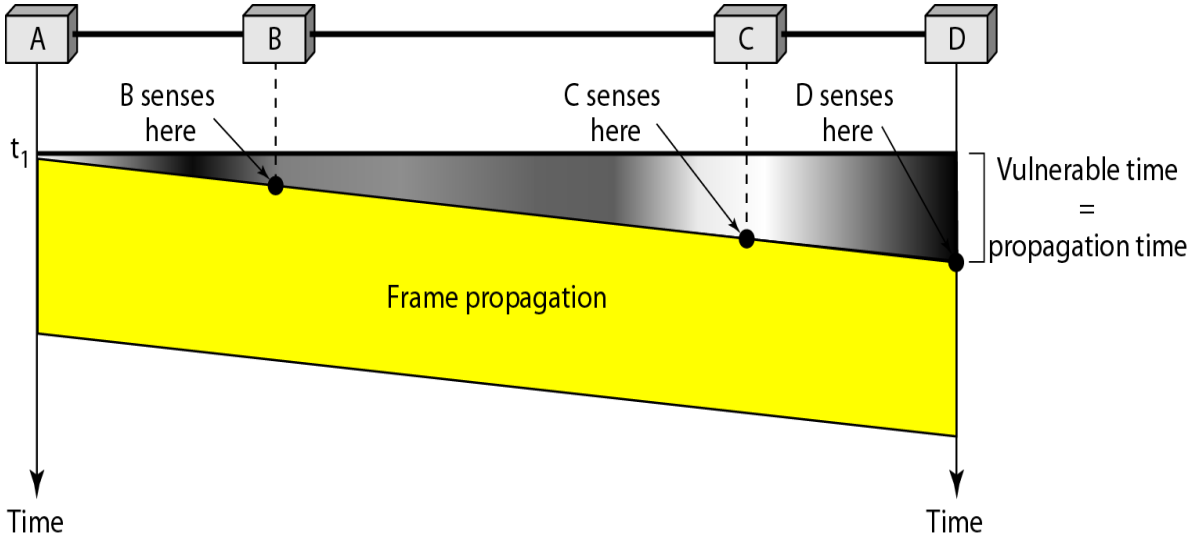
- The possibility of collision still exists because of propagation delay, when a station sends a frame, it still takes time for the first bit to reach every station and for every station to sense it.
- A station may sense the medium and find it idle, only because the first bit sent by

another station has not yet been received At time t1, station B senses the medium and finds it idle, so it sends a frame.

- At time t2 (t2 > t1), station C senses the medium and finds it idle because, at this time, the first bits from station B have not reached station C.
- Station C also sends a frame. The two signals collide and both frames are destroyed

Vulnerable time

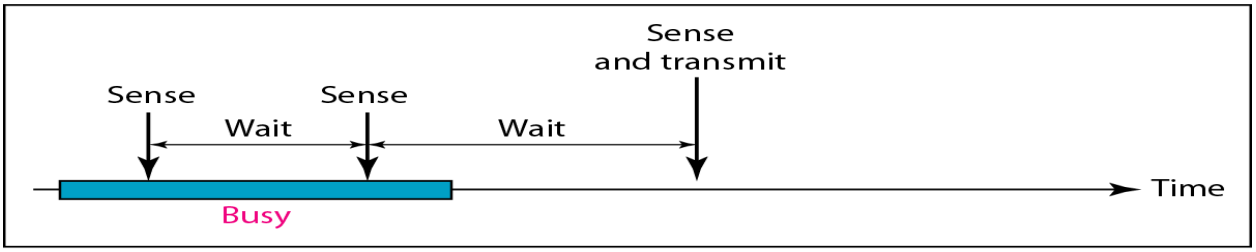
- The vulnerable time for CSMA is the propagation time Tp. This is the time needed for a signal to propagate from one end of the medium to the other.
- When a station sends a frame and any other station tries to send a frame during this time, a collision will result. But if the first bit of the frame reaches the end of the medium, every station will already have heard the bit and will refrain from sending.
- Below Figure shows the worst case. The leftmost station, A, sends a frame at time t1, which reaches the rightmost station D, at time t1 + Tp.
- The gray area shows the vulnerable area in time and space.



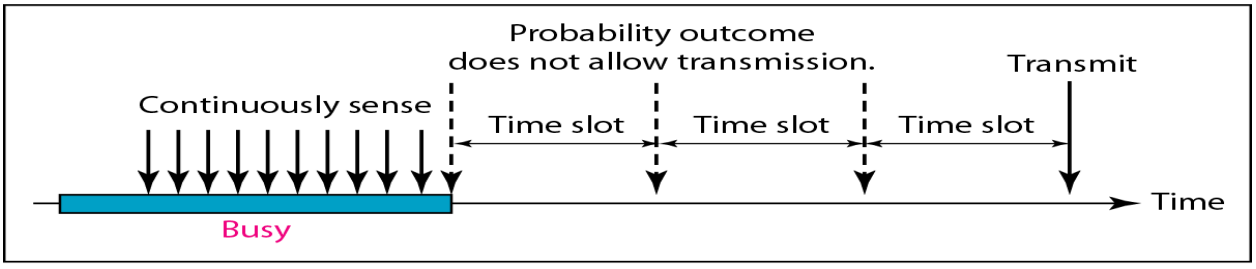
Persistence Methods

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

a. 1-persistent



b. Nonpersistent



c. p-persistent

1-Persistent

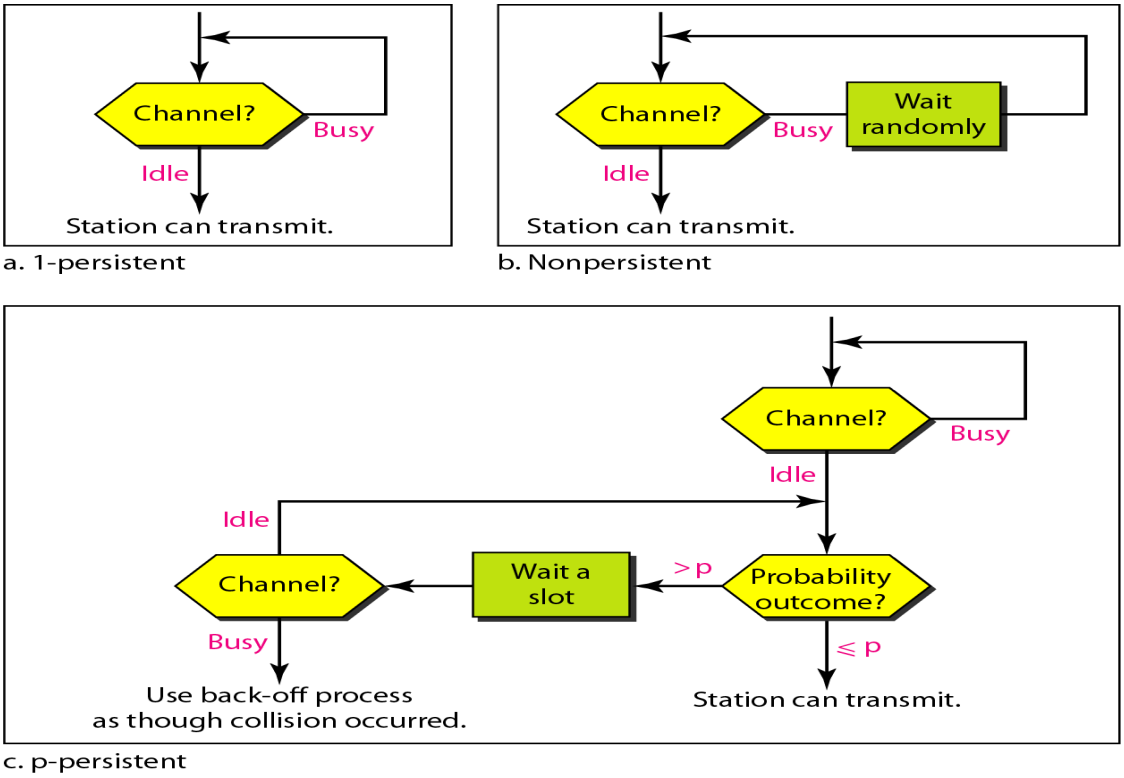
- Is simple and straightforward. After the station finds the line idle, it sends its frame immediately (with probability 1).
- This method has the highest chance of collision because two or more stations may find the line idle and send their frames immediately.

Nonpersistent

- A station that has a frame to send senses the line. If the line is idle, it sends immediately.
- If the line is not idle, it waits a random amount of time and then senses the line again.
- This approach reduces the chance of collision because it is unlikely that two or more stations will wait the same amount of time and retry to send simultaneously, But it reduces the efficiency.

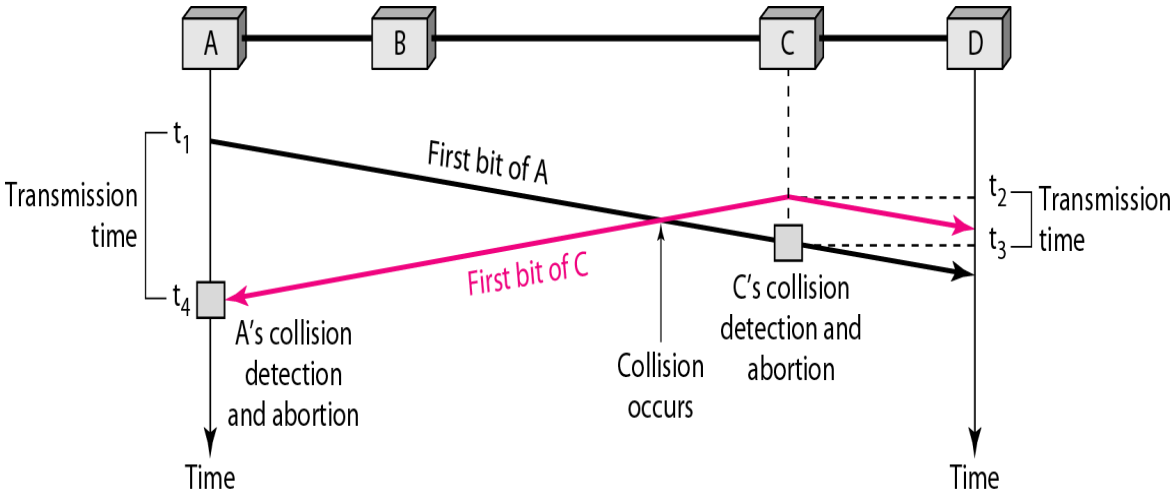
p-Persistent

- It 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.
- After the station finds the line idle it follows these steps:
- With probability p, the station sends its frame
- With probability q = 1 − p, the station waits for the beginning of the next time slot and checks the line again.
- If the line is idle, it goes to step 1.
- If the line is busy, it acts as though a collision has occurred and uses the backoff procedure.

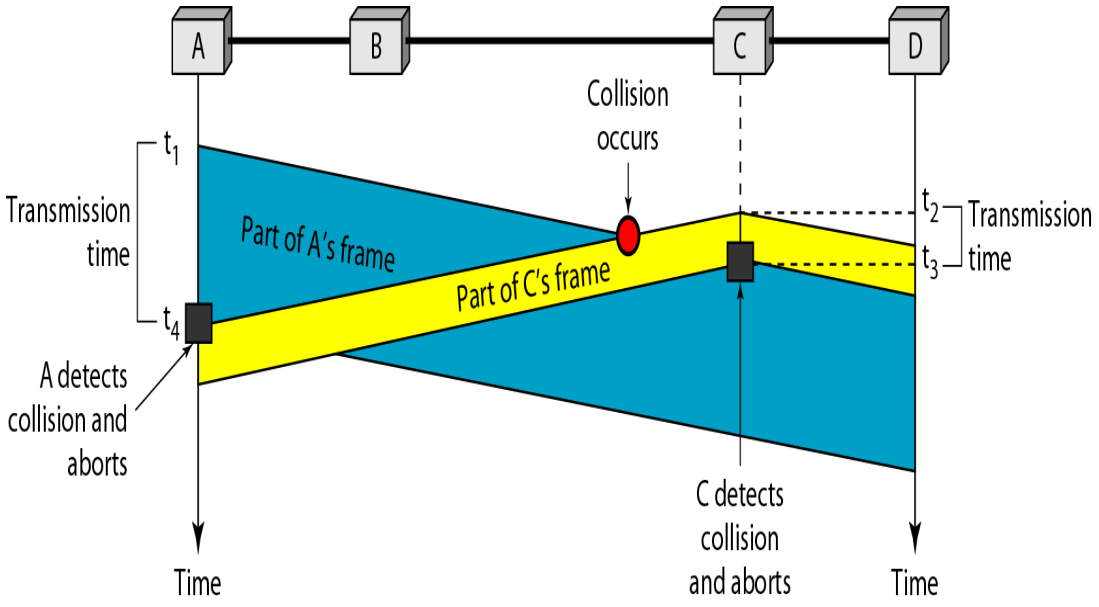


CSMA/CD

- 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.
- Each station continues to send bits in the frame until it detects the collision.
- Below Figure stations A and C are involved in the collision.



- At time t1, station A has executed its persistence procedure and starts sending the bits of its frame.
- At time t2, station C has not yet sensed the first bit sent by A.
- Station C executes its persistence procedure and starts sending the bits in its frame, which propagate both to the left and to the right. The collision occurs sometime after time t2.
- Station C detects a collision at time t3 when it receives the first bit of A's frame.
- Station C immediately aborts transmission.
- Station A detects collision at time t4 when it receives the first bit of C's frame, it also immediately aborts transmission.
- Here A transmits for the duration t4 − t1, C transmits for the duration t3 − t2.



Minimum Frame Size

- 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.
- The entire frame is sent, does not keep a copy of the frame and does not monitor the line for collision detection.
- Therefore, the frame transmission time Tfr must be at least two times the maximum propagation time Tp. That is 2Tp.

Example

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, as we see later) is 25.6 μs, what is the minimum size of the frame?

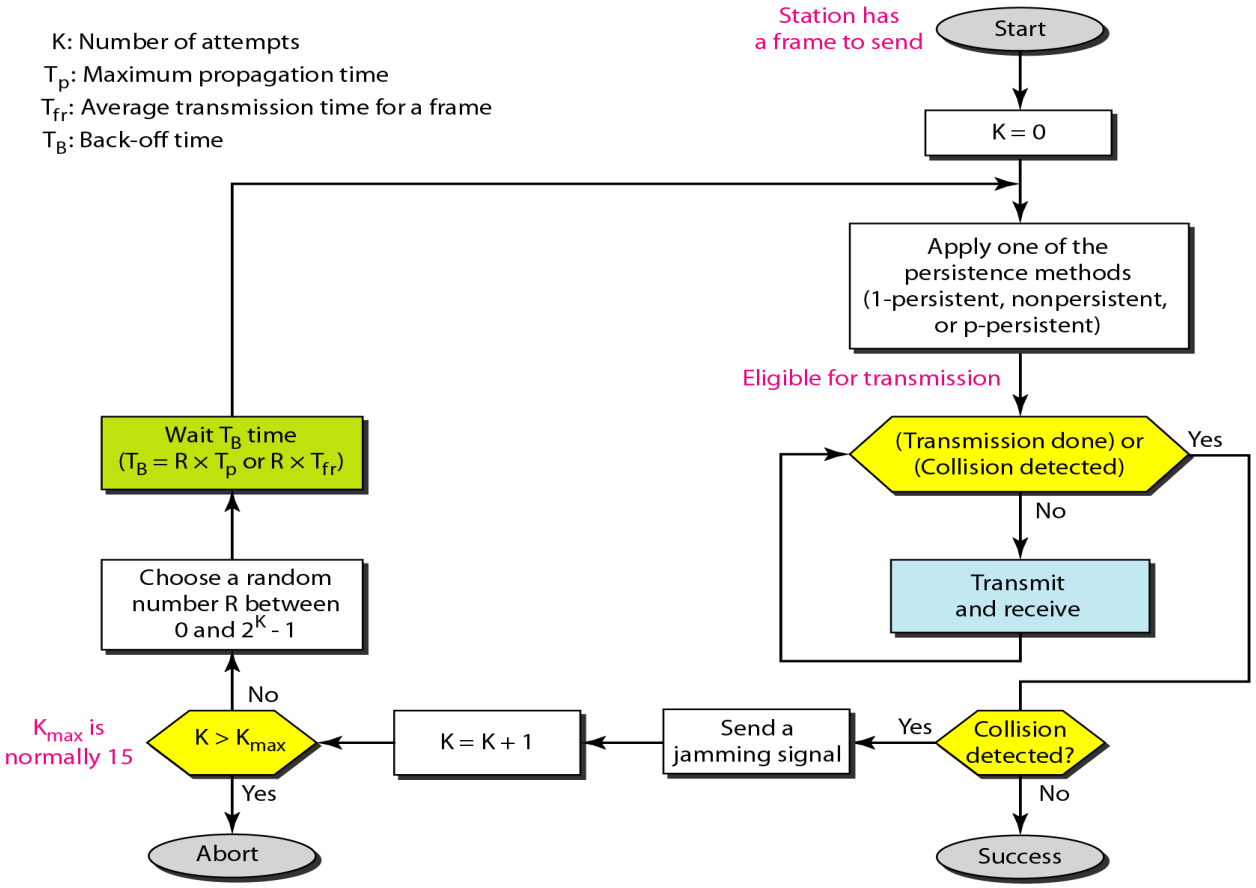
Solution

The frame transmission time is $T_{fr} = 2 \times T_p = 51.2$ 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 × 51.2 µs = 512 bits or 64 bytes. This is actually the minimum size of the frame for Standard Ethernet.

Procedure

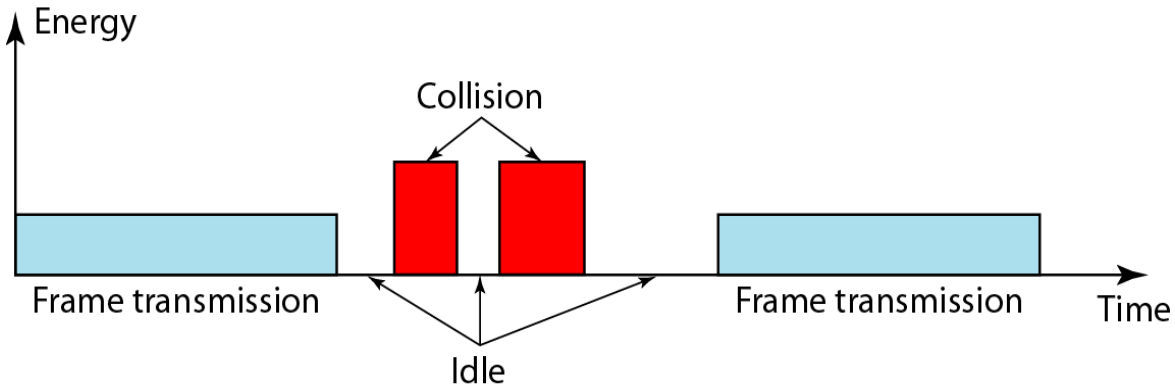
- Below is the flow diagram for CSMA/CD. It is similar to the one for the ALOHA protocol, but there are differences.

1. In addition of the persistence process. sense the channel before start sending the frame by using one of the persistence processes.
2. The frame transmission. In ALOHA, first transmit the entire frame and then wait for an acknowledgment. In CSMA/CD, transmission and collision detection are continuous processes. Here not send the entire frame and then look for a collision. The station transmits and receives continuously and simultaneously. constantly monitor in order to detect one of two conditions: either transmission is finished or a collision is detected. Either event stops transmission. When come out of the loop, if a collision has not been detected, it means that transmission is complete. Otherwise, a collision has occurred.
3. The sending of a short jamming signal to make sure that all other stations become aware of the collision.



Energy Level

- 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.
- Below Figure shows the situation.

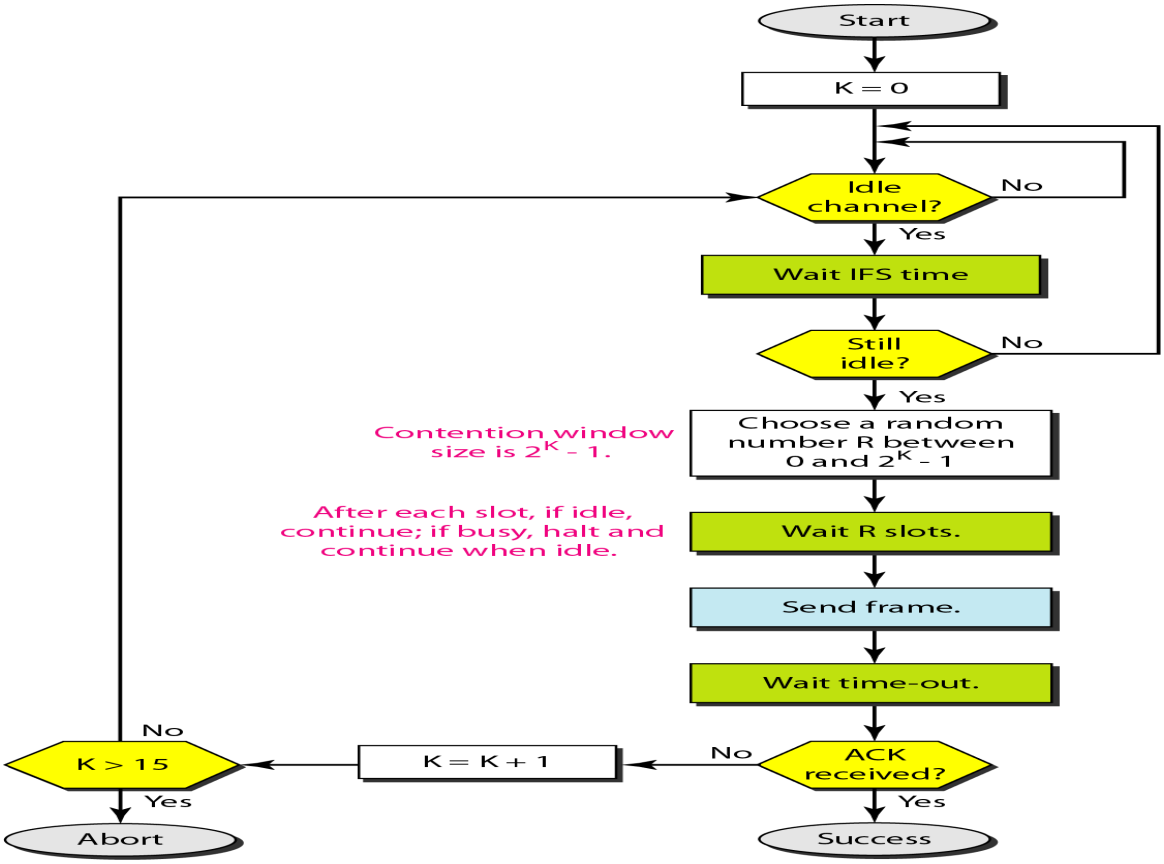


CSMA/CA

- 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 interframe space, the contention window, and acknowledgments, as shown Below Figure.

Interframe Space (IFS).

- ✓ First, collisions are avoided by deferring transmission even if the channel is found idle.
- ✓ When an idle channel is found, the station does not send immediately. It waits for a period of time called the interframe space or IFS.
- ✓ Even though the channel may appear idle when it is sensed, a distant station may have already started transmitting.
- ✓ The IFS time allows the front of the transmitted signal by the distant station to reach 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.

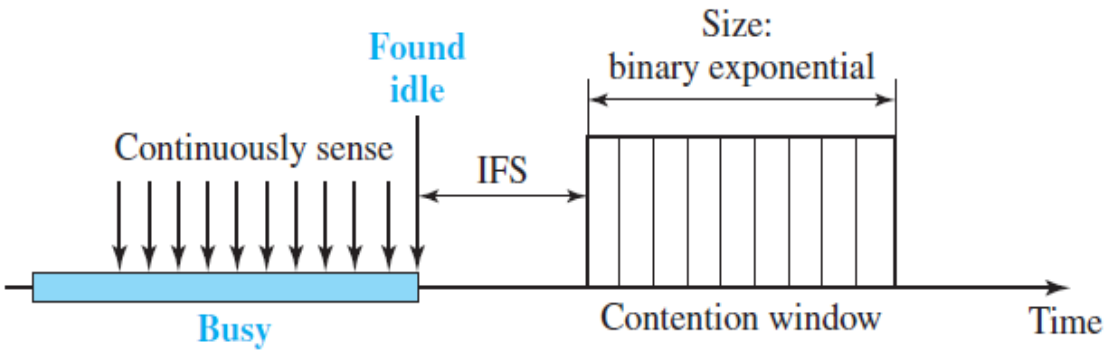


Contention Window.

- ✓ 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 backoff 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.
- ✓ 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. As shown in below figure.

Acknowledgment.

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

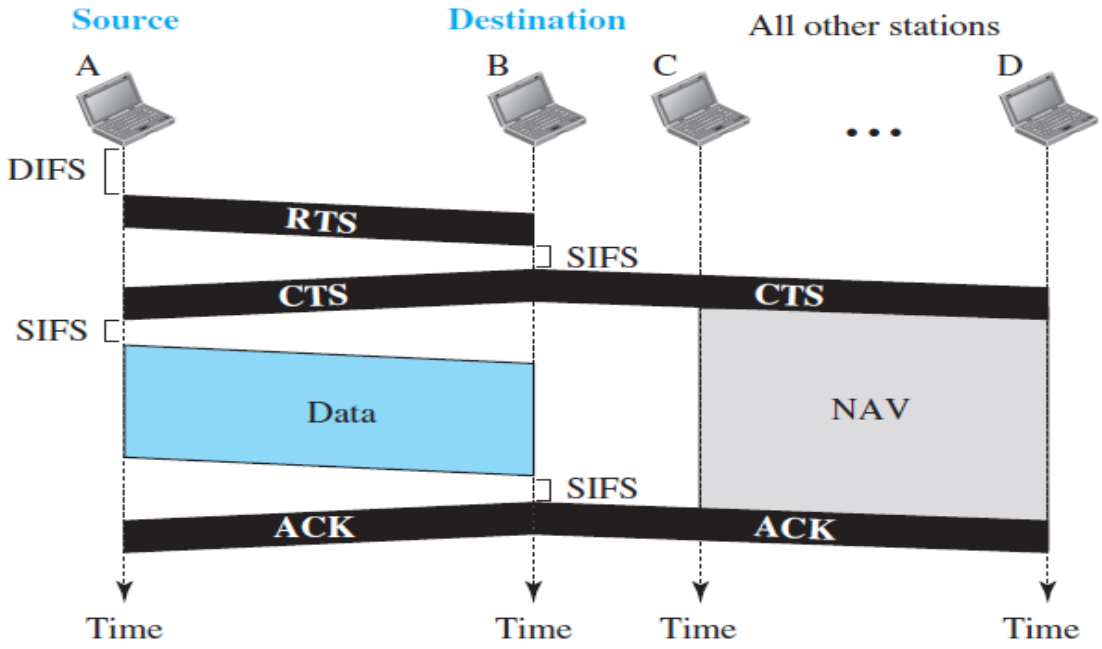


Frame Exchange Time Line

- ❖ Below Figure shows the exchange of data and control frames in time.

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 backoff 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 interframe 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 interframe 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.



Network Allocation Vector

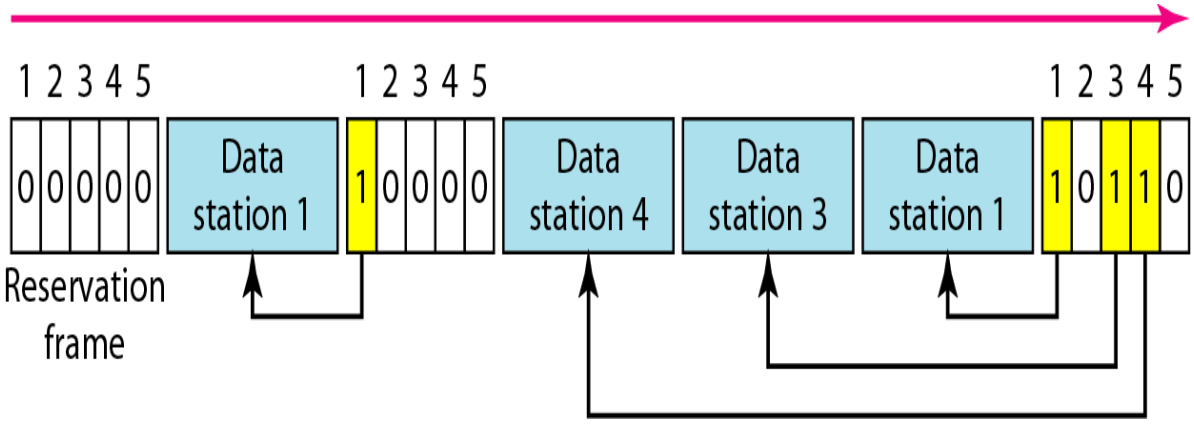
- How do other stations defer sending their data if one station acquires access?
- The key is a feature called NAV.
- When a station sends an RTS frame, it includes the duration of time that it needs to occupy the channel.
- The stations that are affected by this transmission create a timer called a 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.

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.
- We have three types of controlled access
  - i. Reservation
  - ii. Polling
  - iii. Token passing

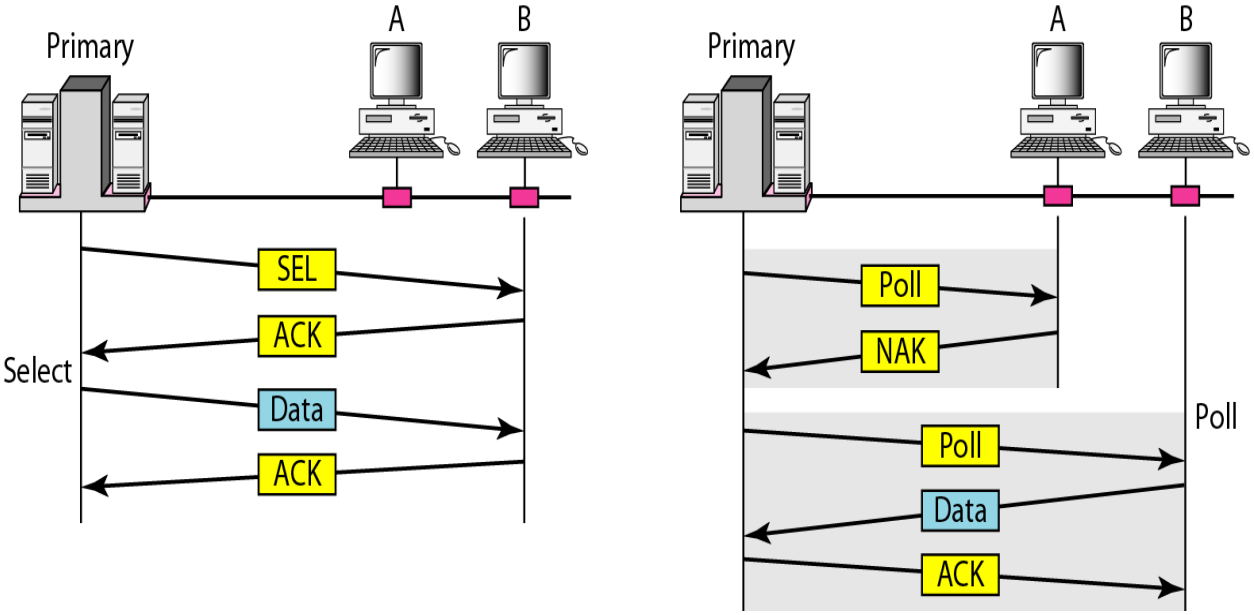
Reservation

- 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 mini slots 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 mini slot.
- The stations that have made reservations can send their data frames after the reservation frame. Below Figure shows a situation with five stations and a five-mini slot reservation frame.
- In the first interval, only stations 1, 3, and 4 have made reservations.
- In the second interval, only station 1 has made a reservation.



Polling

- Polling works with topologies in which one device is designated as a primary station and the other devices are secondary stations.
- All data exchanges must be made through the primary device even when the ultimate destination is a secondary device.
- The primary device controls the link, the secondary devices follow its instructions.
- It is up to the primary device to determine which device is allowed to use the channel at a given time.



Select

- The select function is used whenever the primary device has something to send.
- If the primary is neither sending nor receiving data, it knows the link is available.
- If it has something to send, the primary device sends it. 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

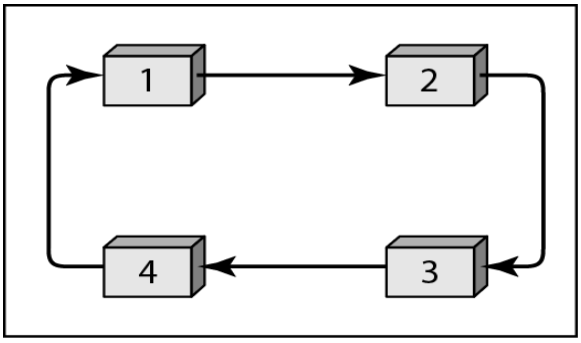
- The poll function is used by the primary device to solicit transmissions from the secondary devices.
- When the primary is ready to receive data, it must ask (poll) each device in turn if it has anything to send.
- When the first secondary is approached, it responds either with a NAK frame if it has nothing to send or with data if it does.
- If the response is negative, then the primary polls the next secondary in the same manner until it finds one with data to send.
- When the response is positive, the primary reads the frame and returns an acknowledgment, verifying its receipt.

Token Passing

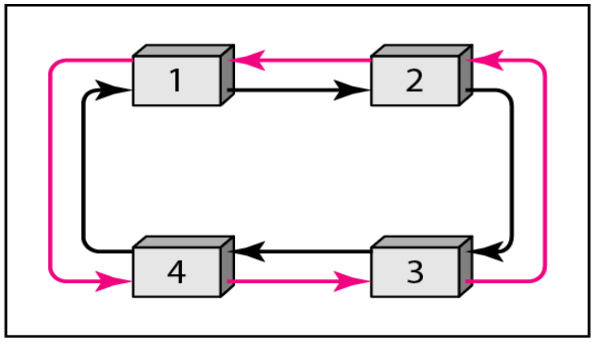
- The stations in a network are organized in a logical ring. For each station, there is a predecessor and a successor.
- The predecessor is the station which is logically before the station in the ring, the successor is the station which is after the station in the ring.
- When a station has some data 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.
- 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.

Logical Ring

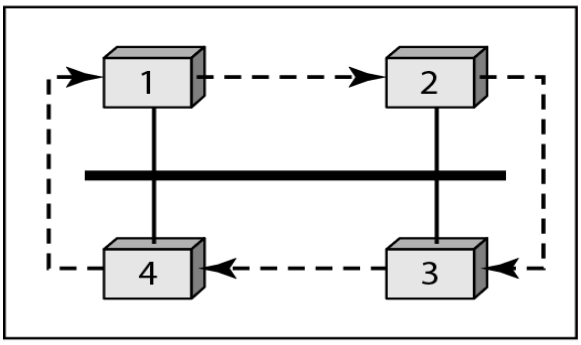
- ✓ In a token-passing network, stations do not have to be physically connected in a ring, the ring can be a logical one.
- ✓ Below Figure shows four different physical topologies that can create a logical ring.



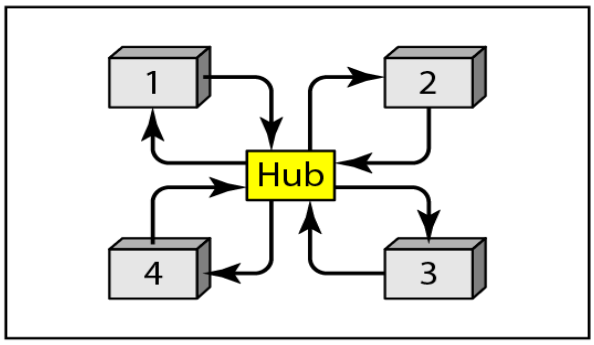
a. Physical ring



b. Dual ring



c. Bus ring



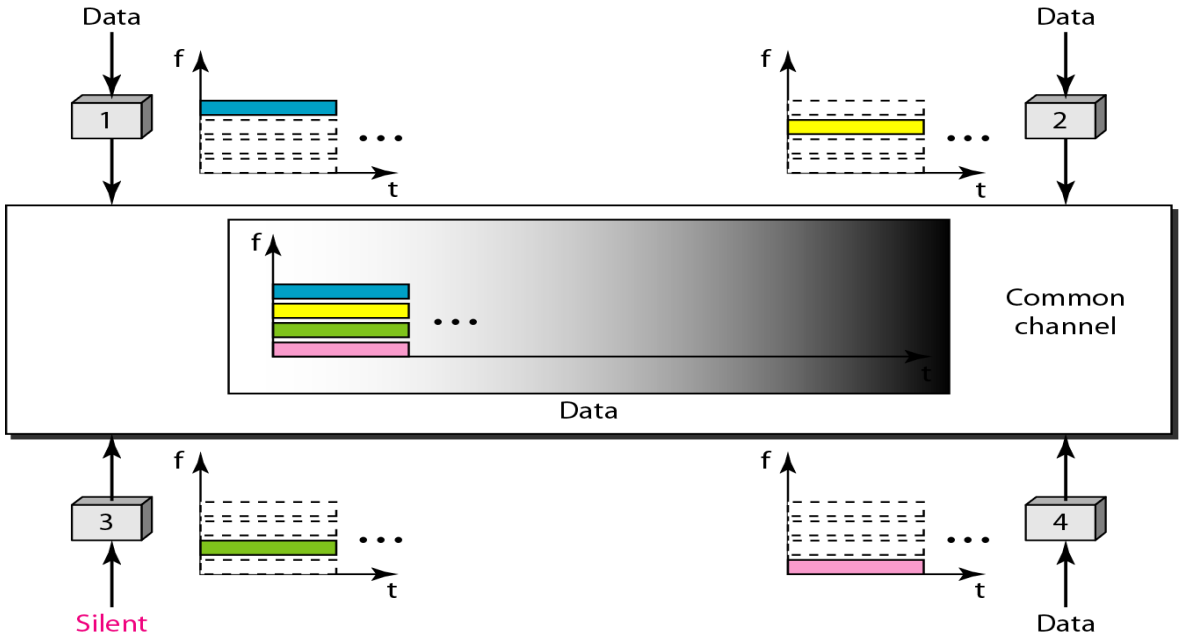
d. Star ring

CHANNELIZATION

- Channelization is a multiple-access method in which the available bandwidth of a link is shared in time, frequency, or through code, among different stations.
- We have Three types
  i. FDMA
  ii. TDMA
  iii. CDMA

FDMA

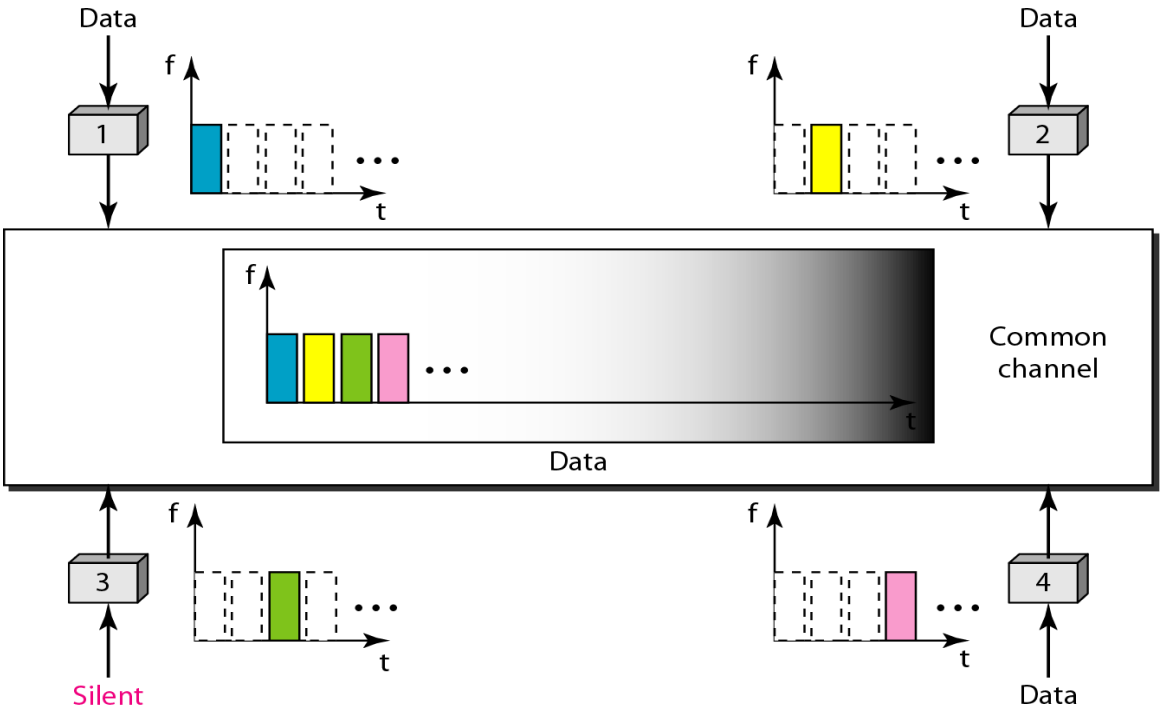
- In frequency-division multiple access (FDMA), the available bandwidth is divided into frequency bands. Each station is allocated a band to send its data.
- Each band is reserved for a specific station, and it belongs to the station all the time.
- Each station also uses a bandpass filter to confine the transmitter frequencies.
- To prevent station interferences, the allocated bands are separated from one another by small guard bands.
- Below Figure shows the idea of FDMA.



- FDMA specifies a predetermined frequency band for the entire period of communication.
- The channels that are combined are low-pass. The multiplexer modulates the signals, combines them, and creates a bandpass signal.
- The signal must be created in the allocated band.
- The signals created at each station are automatically bandpass-filtered.
- They are mixed when they are sent to the common channel.

TDMA

- In time-division multiple access (TDMA), the stations share the bandwidth of the channel in time. Each station is allocated a time slot during which it can send data.
- Each station transmits its data in its assigned time slot. Below Figure shows the idea behind TDMA.
- The main problem with TDMA lies in achieving synchronization between the different stations.
- Each station needs to know the beginning of its slot and the location of its slot.
- This may be difficult because of propagation delays introduced in the system if the stations are spread over a large area.
- To compensate for the delays, insert guard times.
- Synchronization is normally accomplished by having some synchronization bits at the beginning of each slot

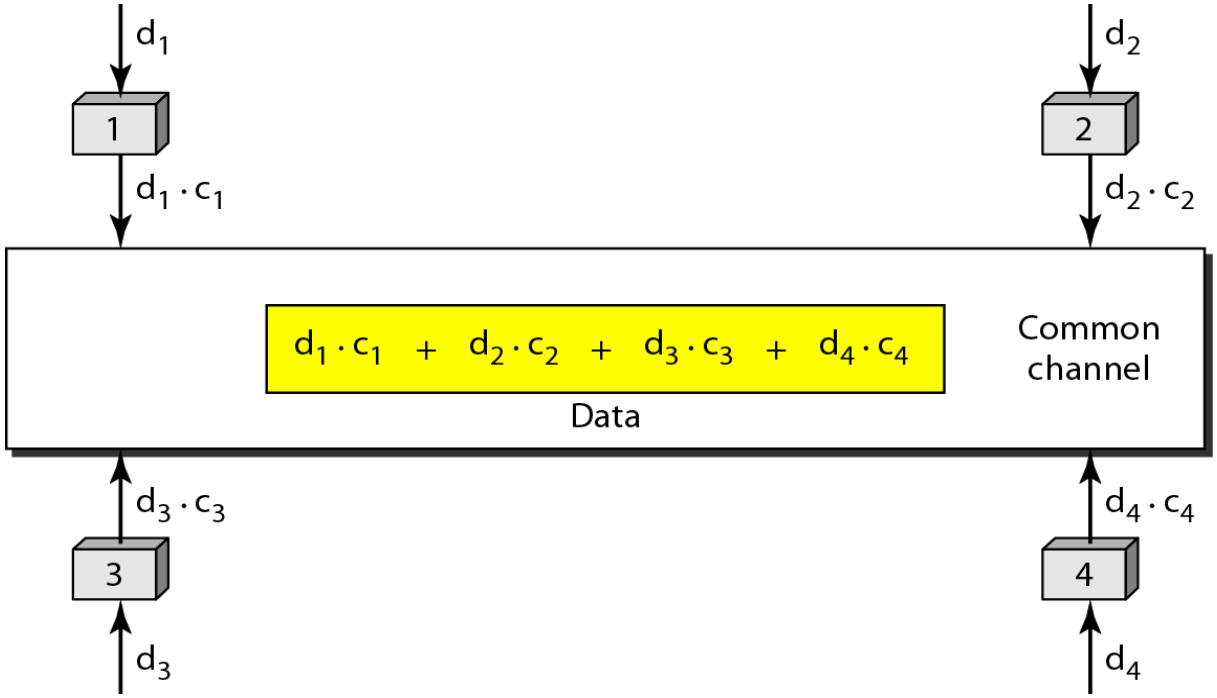


CDMA

- Code-division multiple access (CDMA) was conceived several decades ago.
- CDMA differs from FDMA in that only one channel occupies the entire bandwidth of the link.
- It differs from TDMA in that all stations can send data simultaneously; there is no timesharing.
- Assume we have four stations, 1, 2, 3, and 4, connected to the same channel.
- The data from station 1 are d1, from station 2 are d2, and so on.
- The code assigned to the first station is c1, to the second is c2, and so on.
- We assume that the assigned codes have two properties.

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

- This is shown in below Figure.



$$\text{data} = (d_1 \cdot c_1 + d_2 \cdot c_2 + d_3 \cdot c_3 + d_4 \cdot c_4) \cdot c_1$$
$$= d_1 \cdot c_1 \cdot c_1 + d_2 \cdot c_2 \cdot c_1 + d_3 \cdot c_3 \cdot c_1 + d_4 \cdot c_4 \cdot c_1 = 4 \times d_1$$

1. Each sequence is made of N elements, where N is the number of stations.
2. If we multiply a sequence by a number, every element in the sequence is multiplied by that element. This is called multiplication of a sequence by a scalar. For example,

$$2 \cdot [+1 +1 -1 -1] = [+2 +2 -2 -2]$$

3. If we multiply two equal sequences, element by element, and add the results, we get N, where N is the number of elements in each sequence. This is called the inner product of two equal sequences. For example

$$[+1 +1 -1 -1] \cdot [+1 +1 -1 -1] = 1 + 1 + 1 + 1 = 4$$

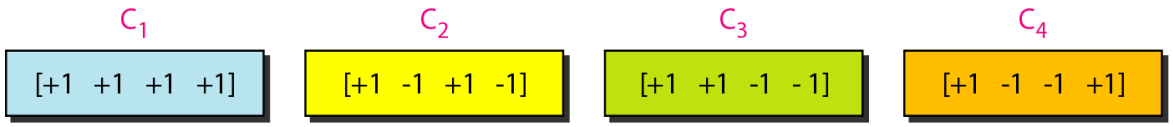
4. **If we multiply two different sequences, element by element, and add the results, we get 0. This is called the inner product of two different sequences. For example,**

$$[+1 +1 -1 -1] \cdot [+1 +1 +1 +1] = 1 + 1 - 1 - 1 = 0$$

5. **Adding two sequences means adding the corresponding elements. The result is another sequence. For example,**

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

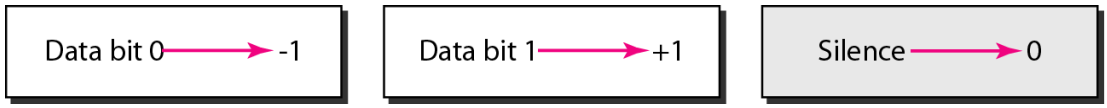
Chip sequences



Data Representation

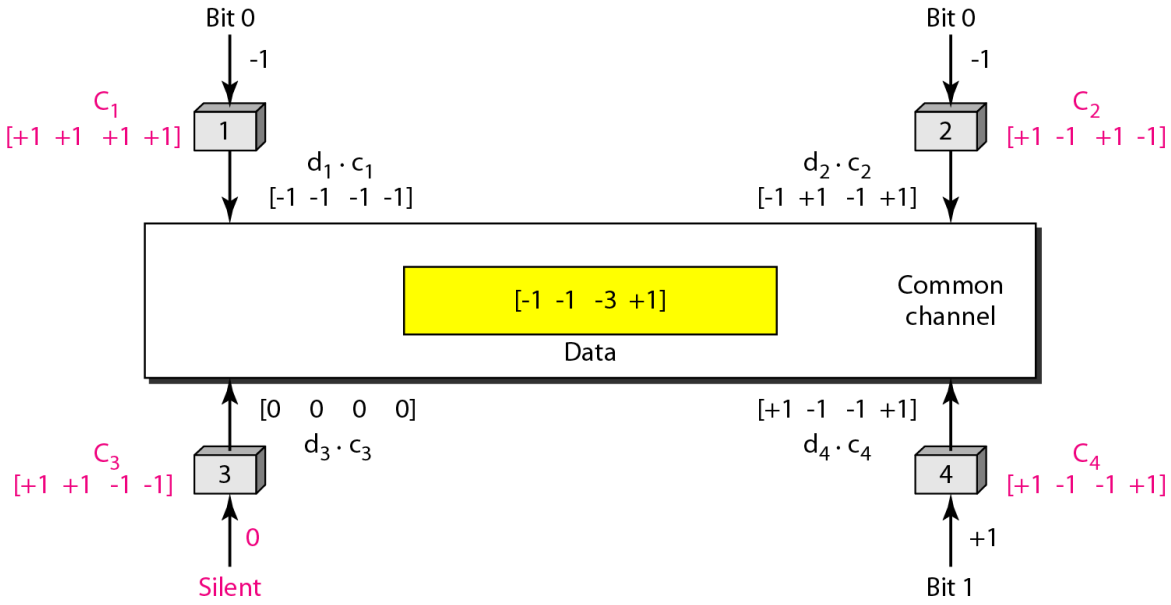
- Rules for encoding:
  - If a station needs to send a 0 bit, it encodes it as −1
  - if it needs to send a 1 bit, it encodes it as +1.
  - When a station is idle, it sends no signal, which is interpreted as a 0.

  These are shown below Figure.



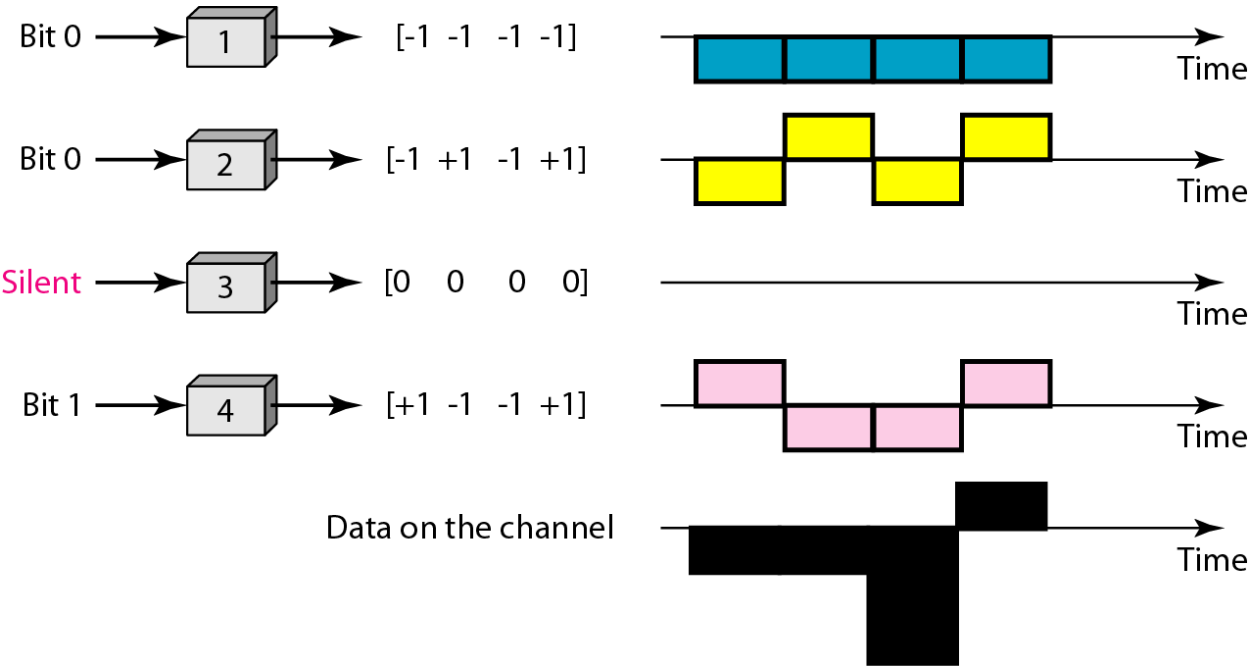
- Encoding and Decoding
- Assume that stations 1 and 2 are sending a 0 bit and channel 4 is sending a 1 bit.
- Station 3 is silent.
- The data at the sender site are translated to −1, −1, 0, and +1. Each station multiplies the corresponding number by its chip (its orthogonal sequence), which is unique for each station.
- The result is a new sequence which is sent to the channel.
- The sequence on the channel is the sum of all four sequences as defined before. Below Figure shows the situation.
- Imagine that station 3, which we said is silent, is listening to station 2. Station 3 multiplies the total data on the channel by the code for station 2, which is [+1 −1 +1 −1], to get

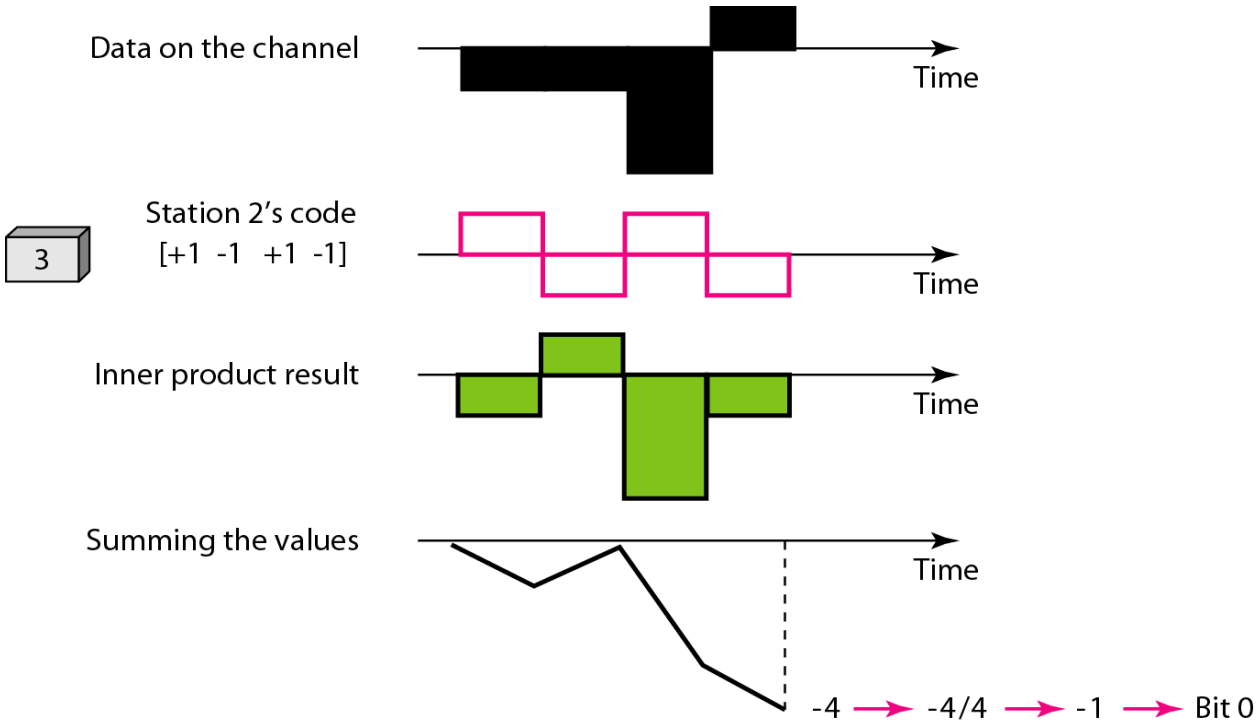
[-1-1-3+1].[+1-1+1-1]=-4/4=-1--→Bit 0



Signal Level

- The process of how the digital signal produced by each station and the data recovered at the destination .
- The figure shows the corresponding signals for each station (using NRZ-L for simplicity) and the signal that is on the common channel.





Sequence Generation

- To generate chip sequences, we use a Walsh table, which is a two-dimensional table with an equal number of rows and columns, as shown in Figure below.
- Each row is a sequence of chips.
- W1 for a one-chip sequence has one row and one column.
- We can choose −1 or +1 for the chip for this trivial table (we chose +1).
- According to Walsh, if we know the table for N sequences WN, we can create the table for 2N sequences W2N.
- The WN with the overbar WN stands for the complement of WN, where each +1 is changed to −1 and vice versa.
- we can create W2 and W4 from W1.
- After we select W1, W2 can be made from four W1s, with the last one the complement of W1.
- After W2 is generated, W4 can be made of four W2s, with the last one the complement of W2.

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

a. Two basic rules

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

b. Generation of W_1, W_2, and W_4

We can use the rows of W_2 and W_4 in Figure

a. For a two-station network, we have
[+1 +1] and [+1 −1].

b. For a four-station network we have
[+1 +1 +1 +1], [+1 −1 +1 −1],
[+1 +1 −1 −1], *and* [+1 −1 −1 +1].

Example

What is the number of sequences if we have 90 stations in our network?

Solution

The number of sequences needs to be 2^m. We need to choose m = 7 and N = 2^7 or 128. We can then use 90 of the sequences as the chips.

Example

Prove that a receiving station can get the data sent by a specific sender if it multiplies the entire data on the channel by the sender's chip code and then divides it by the number of stations.

Solution

Let us prove this for the first station, using our previous four-station example. We can say that the data on the channel

$D = (d_1 \cdot c_1 + d_2 \cdot c_2 + d_3 \cdot c_3 + d_4 \cdot c_4)$.

The receiver which wants to get the data sent by station 1 multiplies these data by c_1.

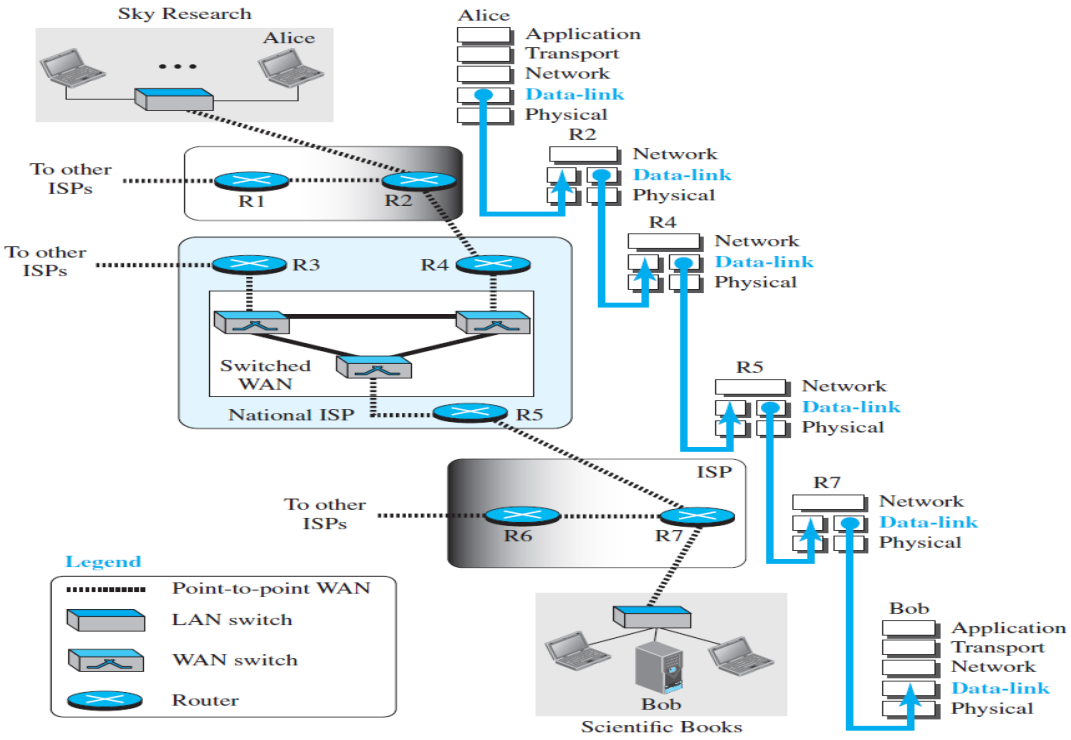
$$\begin{aligned} D \cdot c_1 &= (d_1 \cdot c_1 + d_2 \cdot c_2 + d_3 \cdot c_3 + d_4 \cdot c_4) \cdot c_1 \\ &= d_1 \cdot c_1 \cdot c_1 + d_2 \cdot c_2 \cdot c_1 + d_3 \cdot c_3 \cdot c_1 + d_4 \cdot c_4 \cdot c_1 \\ &= d_1 \times N + d_2 \times 0 + d_3 \times 0 + d_4 \times 0 \\ &= d_1 \times N \end{aligned}$$

When we divide the result by N, we get d_1

DATA LINK LAYER

INTRODUCTION

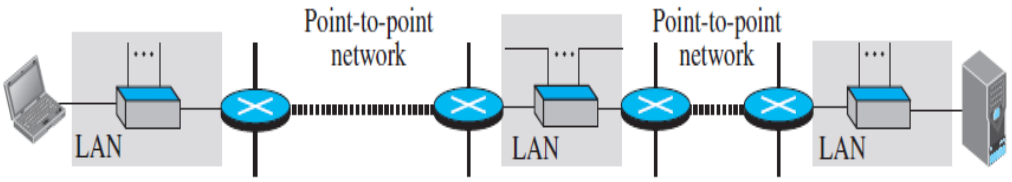
- The Internet is a combination of networks glued together by connecting devices (routers or switches).
- If a packet is to travel from a host to another host, it needs to pass through these networks



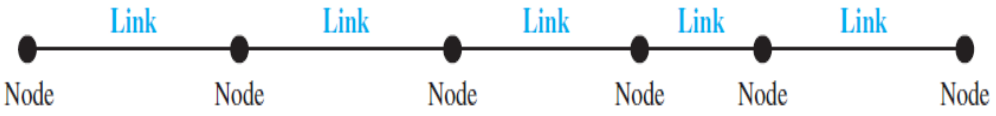
Nodes and Links

- Communication at the data-link layer is node-to-node.
- A data unit from one point in the Internet needs to pass through many networks (LANs and WANs) to reach another point. Theses LANs and WANs are connected by routers.

- Below Figure is a simple representation of links and nodes when the path of the data unit is only six nodes
- The first node is the source host, the last node is the destination host.
- The other four nodes are four routers.
- The first, the third, and the fifth links represent the three LANs.
- The second and the fourth links represent the two WANs.



a. A small part of the Internet

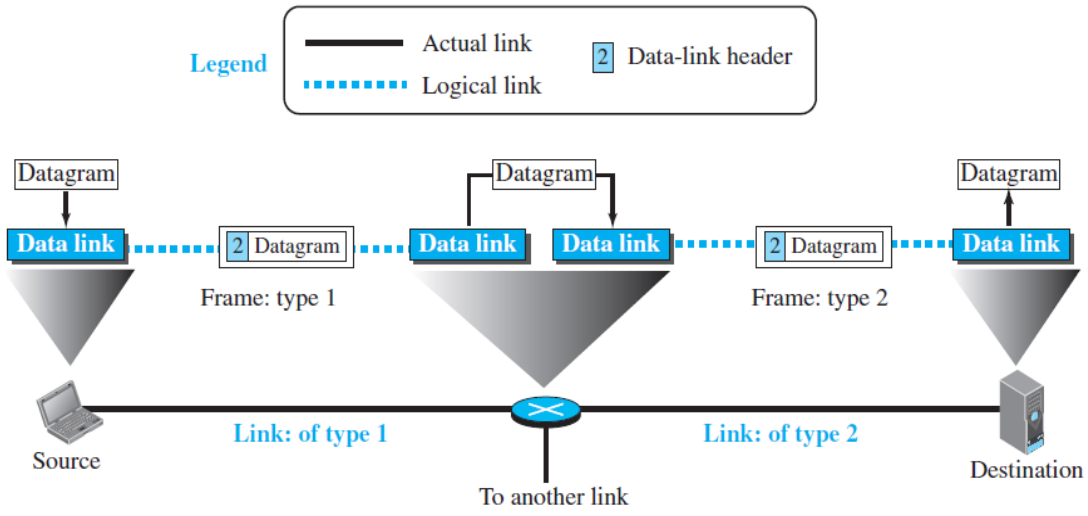


b. Nodes and links

Services

- The duty scope of the data-link layer is node-to-node.
- When a packet is travelling in the Internet, the data-link layer of a node (host or router) is responsible for delivering a datagram to the next node in the path.
- The data-link layer of the sending node needs to encapsulate the datagram received from the network in a frame, and the data-link layer of the receiving node needs to decapsulate the datagram from the frame.
- Below Figure shows the encapsulation and decapsulation at the data-link layer.
- Assume that we have only one router between the source and destination.
- The datagram received by the data-link layer of the source host is encapsulated in a frame.

- The frame is logically transported from the source host to the router.
- The frame is decapsulated at the data-link layer of the router and encapsulated at another frame.
- The new frame is logically transported from the router to the destination host.



Framing

- A packet at the data-link layer is normally called a frame.

Flow Control

- If the producer produces items that cannot be consumed, accumulation of items occurs.

Error Control

- Since electromagnetic signals are susceptible to error, a frame is susceptible to error.
- The error needs first to be detected. After detection, it needs to be either corrected at the receiver node or discarded and retransmitted by the sending node

Congestion Control

- Congestion control is considered an issue in the network layer or the transport layer because of its end-to-end nature.

Two Categories of Links

Point-to-Point link

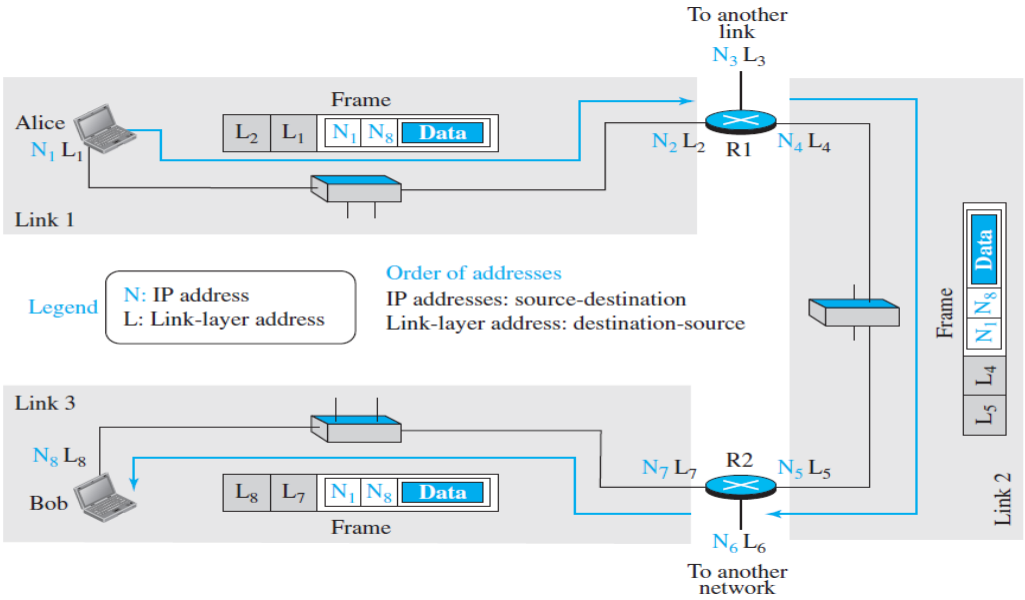
- In a point-to-point link, the link is dedicated to the two devices.

Broadcast link

- Broadcast link, the link is shared between several pairs of devices.

LINK-LAYER ADDRESSING

- The link-layer addresses of the two nodes.
- A link-layer address is sometimes called a link address, sometimes a physical address, and sometimes a MAC address
- When a datagram passes from the network layer to the data-link layer, the datagram will be encapsulated in a frame and two data-link addresses are added to the frame header.
- These two addresses are changed every time the frame moves from one link to another. Below Figure demonstrates the concept in a small internet.



Three Types of addresses

Unicast Address

- Each host or each interface of a router is assigned a unicast address.
- Unicasting means one-to-one communication.
- A frame with a unicast address destination is destined only for one entity in the link.

Example :-

The unicast link-layer addresses in the most common LAN, Ethernet, are 48 bits (six bytes) that are presented as 12 hexadecimal digits separated by colons; for example, the following is a link-layer address of a computer

A3:34:45:11:92:F1

Multicast Address

- Some link-layer protocols define multicast addresses.
- Multicasting means one-to-many communication.

Example:- The multicast link-layer addresses in the most common LAN, Ethernet, are 48 bits (six bytes) that are presented as 12 hexadecimal digits separated by colons. The second digit, however, needs to be an even number in hexadecimal. The following shows a multicast address

A2:34:45:11:92:F1

Broadcast Address

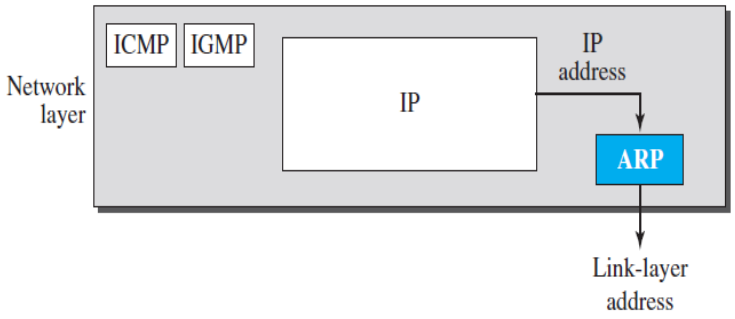
- Some link-layer protocols define a broadcast address.
- Broadcasting means one-to-all communication.
- A frame with a destination broadcast address is sent to all entities in the link.

Example: - The broadcast link-layer addresses in the most common LAN, Ethernet, are 48 bits, all 1s, that are presented as 12 hexadecimal digits separated by colons. The following shows a broadcast address:

FF:FF:FF:FF:FF:FF

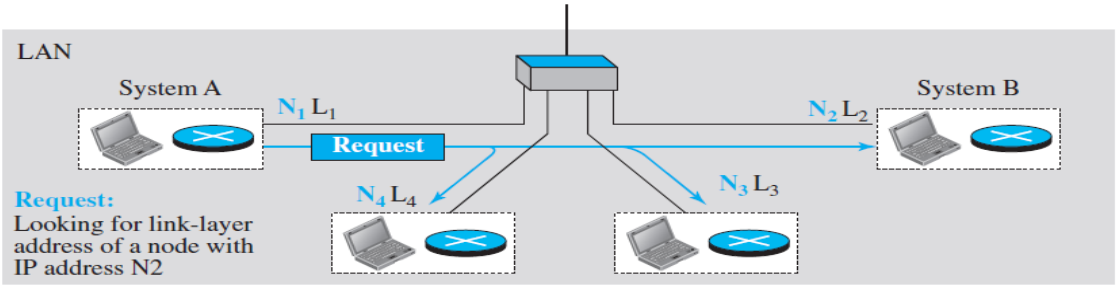
Address Resolution Protocol (ARP)

- Anytime a node has an IP datagram to send to another node in a link, it has the IP address of the receiving node.
- The source host knows the IP address of the default router.
- Each router except the last one in the path gets the IP address of the next router by using its forwarding table.
- The last router knows the IP address of the destination host.
- The IP address of the next node is not helpful in moving a frame through a link.
- This is the time when the Address Resolution Protocol (ARP) becomes helpful.
- The ARP protocol is one of the auxiliary protocols defined in the network layer, as shown in below Figure.
- ARP accepts an IP address from the IP protocol, maps the address to the corresponding link-layer address, and passes it to the data-link layer.

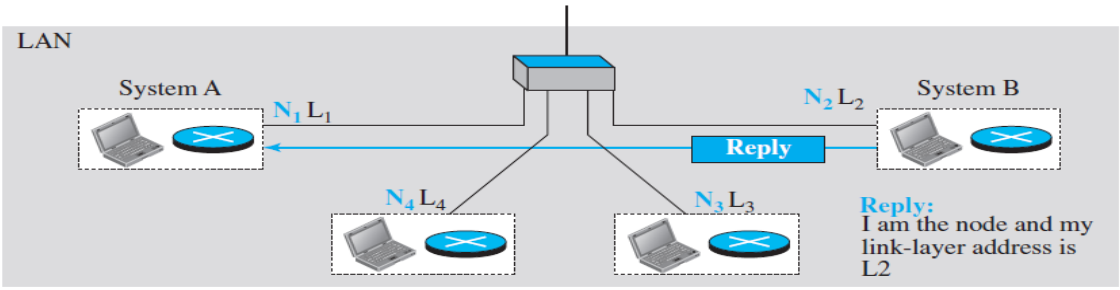


- Anytime a host or a router needs to find the link-layer address of another host or router in its network, it sends an ARP request packet.
- The packet includes the link-layer and IP addresses of the sender and the IP address of the receiver.
- Because the sender does not know the link-layer address of the receiver, the query is broadcast over the link using the link-layer broadcast address.

- Every host or router on the network receives and processes the ARP request packet, but only the intended recipient recognizes its IP address and sends back an ARP response packet.
- The response packet contains the recipient's IP and link-layer addresses.
- The packet is unicast directly to the node that sent the request packet.



a. ARP request is broadcast



b. ARP reply is unicast

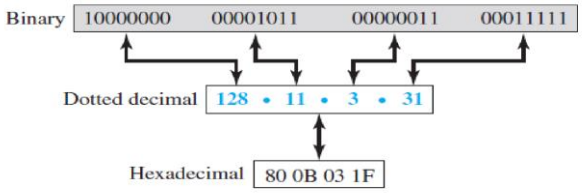
- In Figure a, the system on the left (A) has a packet that needs to be delivered to another system (B) with IP address N2.
- System A needs to pass the packet to its data-link layer for the actual delivery, but it does not know the physical address of the recipient.
- It uses the services of ARP by asking the ARP protocol to send a broadcast ARP request packet to ask for the physical address of a system with an IP address of N2.
- This packet is received by every system on the physical network, but only system B will answer it, as shown in Figure b.
- System B sends an ARP reply packet that includes its physical address.

IPV4 ADDRESSING

- A protocol like IPv4 that defines addresses has an address space.
- An address space is the total number of addresses used by the protocol.
- If a protocol uses b bits to define an address, the address space is 2^b because each bit can have two different values (0 or 1).
- IPv4 uses 32-bit addresses, which means that the address space is 232 or 4,294,967,296 (more than four billion).
- If there were no restrictions, more than 4 billion devices could be connected to the Internet.

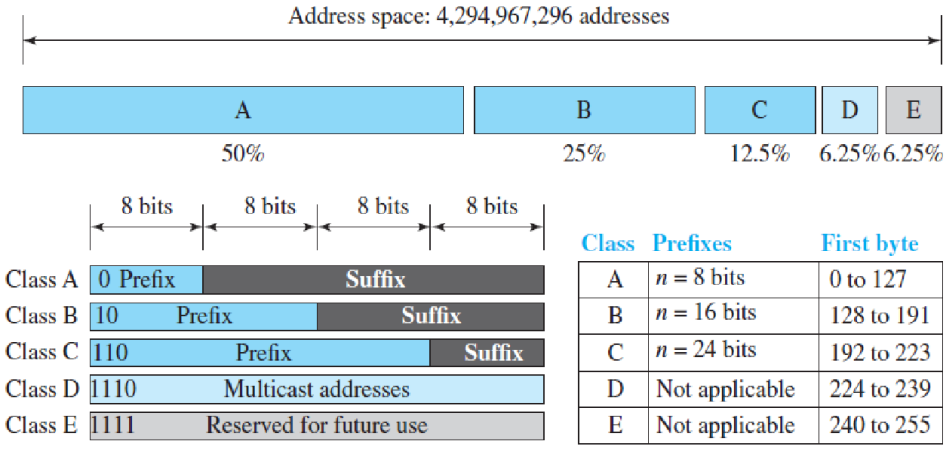
Notation

- There are three common notations to show an IPv4 address: binary notation (base 2), dotted-decimal notation (base 256), and hexadecimal notation (base 16).
- In binary notation, an IPv4 address is displayed as 32 bits. To make the address more readable, one or more spaces are usually inserted between each octet (8 bits).
- Each octet is often referred to as a byte. To make the IPv4 address more compact and easier to read, it is usually written in decimal form with a decimal point (dot) separating the bytes.
- This format is referred to as dotted-decimal notation.
- Each byte (octet) is only 8 bits, each number in the dotted-decimal notation is between 0 and 255.
- Each hexadecimal digit is equivalent to four bits. This means that a 32-bit address has 8 hexadecimal digits.
- This notation is often used in network programming. Below Figure shows an IP address in the three discussed notations



Classful Addressing

- When the Internet started, an IPv4 address was designed with a fixed-length prefix, but to accommodate both small and large networks, three fixed-length prefixes were designed instead of one (n = 8, n = 16, and n = 24).
- The whole address space was divided into five classes (class A, B, C, D, and E), as shown in below Figure.
- This scheme is referred to as classful addressing.
- In class A, the network length is 8 bits, but since the first bit, which is 0, defines the class, only seven bits as the network identifier.
- This means there are only 2^7 =128 Networks in the world that can have a class A address.
- In class B, the network length is 16 bits, but since the first two bits, which are (10)2, define the class, only 14 bits as the network identifier.
- This means there are only 2^{14} = 16,384 networks in the world that can have a class B address.
- All addresses that start with (110)2 belong to class C.
- In class C, the network length is 24 bits, but since three bits define the class, only 21 bits as the network identifier. There are 2^{21} = 2,097,152 networks in the world that can have a class C address.
- Class D is not divided into prefix and suffix. It is used for multicast addresses.
- All addresses that start with 1111 in binary belong to class E.
- As in Class D, Class E is not divided into prefix and suffix and is used as reserve.



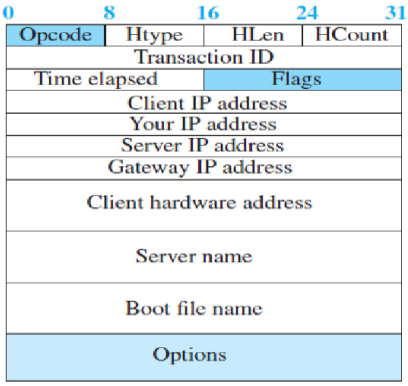
Class	Prefixes	First byte
A	$n = 8$ bits	0 to 127
B	$n = 16$ bits	128 to 191
C	$n = 24$ bits	192 to 223
D	Not applicable	224 to 239
E	Not applicable	240 to 255

Dynamic Host Configuration Protocol (DHCP)

- A block of addresses are assigned to an organization, the network administration can manually assign addresses to the individual hosts or routers.
- Address assignment in an organization can be done automatically using the Dynamic Host Configuration Protocol (DHCP).
- DHCP is an application-layer program, using the client-server paradigm, that actually helps TCP/IP at the network layer.
- DHCP has found such widespread use in the Internet that it is often called a plugand-play protocol.
- A network manager can configure DHCP to assign permanent IP addresses to the host and routers.
- DHCP can also be configured to provide temporary, on demand, IP addresses to hosts.
- The second capability can provide a temporary IP address to a traveler to connect her laptop to the Internet while she is staying in the hotel.
- It also allows an ISP with 1000 granted addresses to provide services to 4000 households, assuming not more than one-fourth of customers use the Internet at the same time.

DHCP Message Format

- DHCP is a client-server protocol in which the client sends a request message and the server returns a response message.
- The general format of the DHCP message in below Figure.
- The 64-byte option field has a dual purpose.
- It can carry either additional information or some specific vendor information.
- The server uses a number, called a magic cookie, in the format of an IP address with the value of 99.130.83.99.



Fields:
Opcode: Operation code, request (1) or reply (2)
Htype: Hardware type (Ethernet, ...)
HLen: Length of hardware address
HCount: Maximum number of hops the packet can travel
Transaction ID: An integer set by the client and repeated by the server
Time elapsed: The number of seconds since the client started to boot
Flags: First bit defines unicast (0) or multicast (1); other 15 bits not used
Client IP address: Set to 0 if the client does not know it
Your IP address: The client IP address sent by the server
Server IP address: A broadcast IP address if client does not know it
Gateway IP address: The address of default router
Server name: A 64-byte domain name of the server
Boot file name: A 128-byte file name holding extra information
Options: A 64-byte field with dual purpose described in text

- When the client finishes reading the message, it looks for this magic cookie.
- If present, the next 60 bytes are options.
- An option is composed of three fields: a 1-byte tag field, a 1-byte length field, and a variable-length value field.
- There are several tag fields that are mostly used by vendors.
- If the tag field is 53, the value field defines one of the 8 message types shown in below Figure.