Syllabus
Data Link Layer: Data Link Control - Line discipline, Flow control, Error control; Data Link protocols – Asynchronous Protocols, Synchronous protocols, Character oriented protocols, Bit oriented protocols, Link Access Procedures.

Data Link Layer

The goal of the data link layer is to provide reliable, efficient communication between adjacent machines connected by a single communication channel. Specifically:

1. Group the physical layer bit stream into units called frames. Note that frames are nothing more than ``packets'' or ``messages''. By convention, we shall use the term ``frames'' when discussing DLL packets.
2. Sender calculates the checksum and sends checksum together with data. The checksum allows the receiver to determine when a frame has been damaged in transit or received correctly.
3. Receiver recomputes the checksum and compares it with the received value. If they differ, an error has occurred and the frame is discarded.
4. Error control protocol returns a positive or negative acknowledgment to the sender. A positive acknowledgment indicates the frame was received without errors, while a negative acknowledgment indicates the opposite.
5. Flow control prevents a fast sender from overwhelming a slower receiver. For example, a supercomputer can easily generate data faster than a PC can consume it.
6. In general, data link layer provides service to the network layer. The network layer wants to be able to send packets to its neighbors without worrying about the details of getting it there in one piece.

The most important functions of Data Link layer to satisfy the above requirements are **Error Control** and **Flow Control**. Collectively, these functions are known as **data link control**.

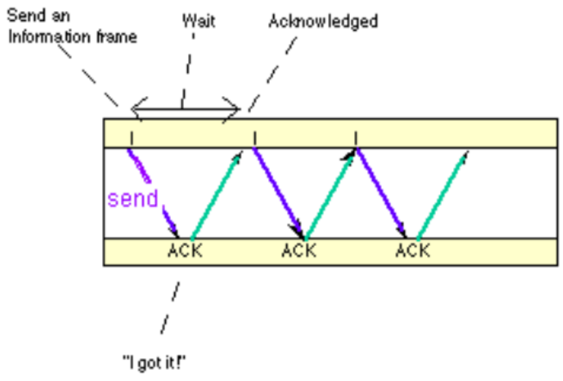
Flow Control is a technique so that transmitter and receiver with different speed characteristics can communicate with each other. Flow control ensures that a transmitting station, such as a server with higher processing capability, does not overwhelm a receiving station, such as a desktop system, with lesser processing capability. This is where there is an orderly flow of transmitted data between the source and the destination.

There are two methods developed for flow control namely **Stop-and-wait** and **Sliding-window**. Stop-and-wait is also known as Request/reply sometimes. Request/reply (Stop-and-wait) flow control requires each data packet to be acknowledged by the remote host before the next packet is sent. This is discussed in detail in the following subsection. **Sliding window** algorithms, used by TCP, permit multiple data packets to be in simultaneous transit, making more efficient use of network bandwidth.

Stop-and-Wait Protocol:

This is the simplest form of flow control where a sender transmits a data frame. After receiving the frame, the receiver indicates its willingness to accept another frame by sending back an ACK frame acknowledging the frame just received. The sender must wait until it receives the ACK frame before sending the next data frame. This is sometimes referred to as ping-pong behavior, request/reply is simple to understand and easy to implement, but not very efficient. In LAN environment with fast links, this isn't much of a concern, but WAN links will spend most of their time idle, especially if several hops are required.

Figure below illustrates the operation of the stop-and-wait protocol. The blue arrows show the sequence of data frames being sent across the link from the sender (top to the receiver (bottom). The protocol relies on two-way transmission (full duplex or half duplex) to allow the receiver at the remote node to return frames acknowledging the successful transmission. The acknowledgements are shown in green in the diagram, and flow back to the original sender. A small processing delay may be introduced between reception of the last byte of a Data PDU and generation of the corresponding ACK.



Major drawback of Stop-and-Wait Flow Control is that only one frame can be in transmission at a time, this leads to inefficiency if propagation delay is much longer than the transmission delay.

Link Utilization in Stop-and-Wait:

Let us assume the following: Transmission time: The time it takes for a station to transmit a frame (normalized to a value of 1).

Propagation delay: The time it takes for a bit to travel from sender to receiver (expressed as *a*).

– $a < 1$:The frame is sufficiently long such that the first bits of the frame arrive at the destination before the source has completed transmission of the frame.
– $a > 1$: Sender completes transmission of the entire frame before the leading bits of the frame arrive at the receiver.

– The link utilization U = 1/(1+2a),

a = Propagation time / transmission time.

It is evident from the above equation that the link utilization is strongly dependent on the ratio of the propagation time to the transmission time. When the propagation time is small, as in case of LAN environment, the link utilization is good. But, in case of long propagation delays, as in case of satellite communication, the utilization can be very poor. To improve the link utilization, we can use the following (sliding-window) protocol instead of using stop-and-wait protocol.

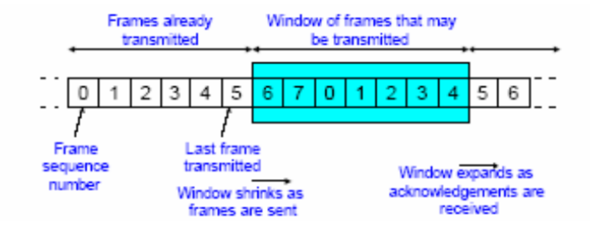
Sliding Window Protocol

With the use of multiple frames for a single message, the stop-and-wait protocol does not perform well. Only one frame at a time can be in transit. In stop-and-wait flow control, if a > 1, serious inefficiencies result. Efficiency can be greatly improved by allowing multiple frames to be in transit at the same time. Efficiency can also be improved by making use of the full duplex line. To keep track of the frames, sender station sends sequentially numbered frames. Since the sequence number to be used occupies a field in the frame, it should be of limited size.

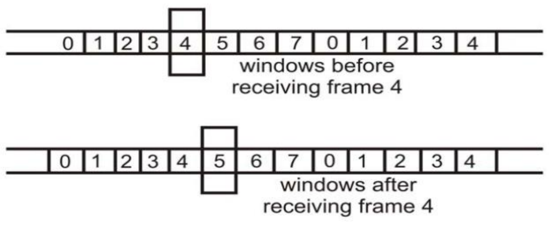
If the header of the frame allows k bits, the sequence numbers range from 0 to $2^k – 1$. Sender maintains a list of sequence numbers that it is allowed to send (sender window). The size of the sender's window is at most $2^k – 1$. The sender is provided with a buffer equal to the window size. Receiver also maintains a window of size $2^k – 1$.

The receiver acknowledges a frame by sending an ACK frame that includes the sequence number of the next frame expected. This also explicitly announces that it is prepared to receive the next N frames, beginning with the number specified. This scheme can be used to acknowledge multiple frames. It could receive frames 2, 3, 4 but withhold ACK until frame 4 has arrived. By returning an ACK with sequence number 5, it acknowledges frames 2, 3, 4 in one go. The receiver needs a buffer of size 1.

Sender sliding Window: At any instant, the sender is permitted to send frames with sequence numbers in a certain range (the sending window) as shown in Fig. below


Frames already transmitted
Window of frames that may be transmitted
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6
Frame sequence number
Last frame transmitted
Window shrinks as frames are sent
Window expands as acknowledgements are received

Receiver sliding Window: The receiver always maintains a window of size 1 as shown in Fig below. It looks for a specific frame (frame 4 as shown in the figure) to arrive in a specific order. If it receives any other frame (out of order), it is discarded and it needs to be resent. However, the receiver window also slides by one as the specific frame is received and accepted as shown in the figure.


0 1 2 3 4 5 6 7 0 1 2 3 4
windows before receiving frame 4
0 1 2 3 4 5 6 7 0 1 2 3 4
windows after receiving frame 4

The receiver acknowledges a frame by sending an ACK frame that includes the sequence number of the next frame expected. This also explicitly announces that it is prepared to receive the next N frames, beginning with the number specified. This scheme can be used to acknowledge multiple frames. It could receive frames 2, 3, 4 but withhold ACK until frame 4 has arrived. By returning an ACK with sequence number 5, it acknowledges frames 2, 3, 4 at one time. The receiver needs a buffer of size 1.

On the other hand, if the local application can process data at the rate it's being transferred; sliding window still gives us an advantage. If the window size is larger than the packet size, then multiple packets can be outstanding in the network, since the sender knows that buffer space is available on the receiver to hold all of them.

Ideally, a steady state condition can be reached where a series of packets (in the forward direction) and window announcements (in the reverse direction) are constantly in transit. As each new window announcement is received by the sender, more data packets are transmitted.

As the application reads data from the buffer (remember, we're assuming the application can keep up with the network), more window announcements are generated. Keeping a series of data packets in transit ensures the efficient use of network resources. Hence, Sliding Window Flow Control

- Allows transmission of multiple frames.
- Assigns each frame a k-bit sequence number.
- Range of sequence number is [0…2_k-1], i.e., frames are counted modulo 2k.

The link utilization in case of Sliding Window Protocol

U = 1, for $N > 2a + 1$

$N/(1+2a)$, for $N < 2a + 1$

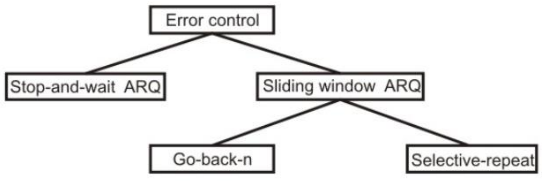
Where N = the window size, and a = Propagation time / transmission time.

Error Control

Error Control involves both error detection and error correction. It is necessary because errors are inevitable in data communication, in spite of the use of better equipment and reliable transmission media based on the current technology. In the preceding lesson we have already discussed how errors can be detected. In this lesson we shall discuss how error control is performed based on retransmission of the corrupted data. When an error is detected, the receiver can have the specified frame retransmitted by the sender. This process is commonly known as **Automatic Repeat Request (ARQ).** For example, Internet's Unreliable Delivery Model allows packets to be discarded if network resources are not available, and demands that ARQ protocols make provisions for retransmission.

Error Control Techniques

When an error is detected in a message, the receiver sends a request to the transmitter to retransmit the ill-fated message or packet. The most popular retransmission scheme is known as Automatic-Repeat-Request (ARQ). Such schemes, where receiver asks transmitter to re-transmit if it detects an error, are known as reverse error correction techniques. There exist three popular ARQ techniques, as shown in Fig. below:



Data Link Protocols

Data Link Protocols are divided into two categories:

Asynchronous Protocols.

Asynchronous protocols treat each character in a bit stream independently.

- These protocols are used in modems.
- They use start and stop bits, and variable gaps between characters.
- They are slower than synchronous protocols in transmitting data.
- The different asynchronous protocols are: XMODEM, YMODEM, ZMODEM and Block Asynchronous Transmission (BLAST)

Synchronous Protocols.

Synchronous Protocols take the whole bit stream and divide it into characters of equal size.

- These protocols have high speed and are used for LAN, WAN and MAN.
- Synchronous protocols are categorized into two groups:
  - Character-Oriented Protocol.
  - Bit-Oriented Protocol.

Character-Oriented Protocol:

It interprets frame as a series of characters.

- These are also known as Byte-Oriented Protocols.
- Control information is inserted as separate control frames or as addition to existing data frame.
- The example of character-oriented protocol is Binary Synchronous Communication (BSC) developed by IBM.

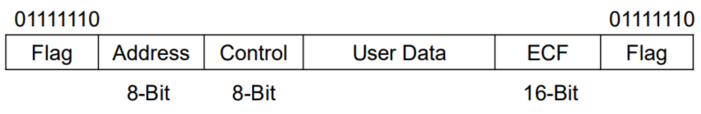
Bit-Oriented Protocol:

It interprets frame as a series of bits.

- Control information can be inserted as bits depending on the information to be contained in the frame.
- Bit-oriented protocol can pack more information into shorter frames.
- The examples of bit-oriented protocol are:
    - Synchronous Data Link Control (SDLC).
    - High Level Data Link Control (HDLC).

Synchronous Data Link Control (SDLC) Protocol:

- SDLC protocol was developed by IBM in 1975.
- After developing SDLC, IBM submitted it to American National Standard Institute (ANSI) and to International Standard Organization (ISO) for acceptance.
- ANSI modified it to ADCCP (Advanced Data Communication Control Procedure.
- ISO modified it to HDLC (High Level Data Link Control).
- The frame format of SDLC is:



- The flag sequence of 8-bits 01111110 marks the beginning and ending of the frame.
- Address field contains the address of the receiver.
- Control field carries the sequence number, acknowledgement, requests and responses.
- The user data field carries the data and is of variable length.
- ECF stands for Error Checking Field and is of 16- bits. It is used for error control.

High Level Data Link Control (HDLC) Protocol:

HDLC came into existence after ISO modified the SDLC protocol.

- It is a bit-oriented protocol that supports both half and full duplex communication.
- Systems using HDLC are characterized by:
    - Station Types.
    - Configuration.
    - Response Modes

Station Types

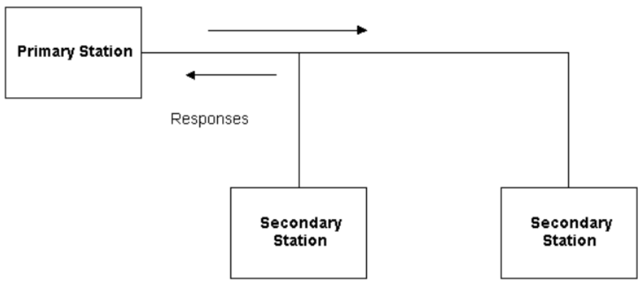
To make HDLC protocol applicable to various network configurations, three types of stations have been defined:

- Primary Station
    - It has complete control over the link at any time.
    - It has the responsibility of connecting & disconnecting the link.
    - The frames sent by primary station are called commands.
- Secondary Station
    - All the secondary stations work under the control of primary station.
    - The frames sent by secondary station are called responses.
- Combined Station
    - A combined station can behave either as primary or as secondary station.
    - It can send commands as well as responses.

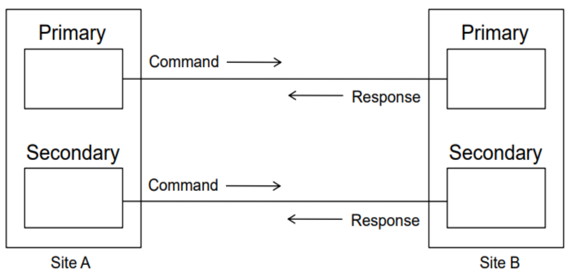
Configuration

Configuration defines how the various stations are connected to a link.

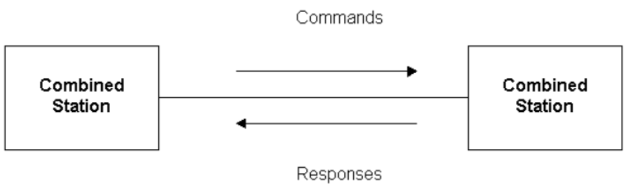
- There are three possible configurations:
    - Unbalanced Configuration.
        - This type of configuration exists if one station is primary and other is secondary.



    - Symmetrical Configuration.
        - It can further be of two types:
            - Point-to-Point Unbalanced Configuration:
                - If there is one primary and one secondary station.
            - Multipoint Unbalanced Configuration:
                - If there is one primary and many secondary stations.



    - Balanced Configuration
        - In this configuration, both sites have combined stations.
        - These combined stations are connected with single link.
        - This single link can be controlled by either station.



Link Access

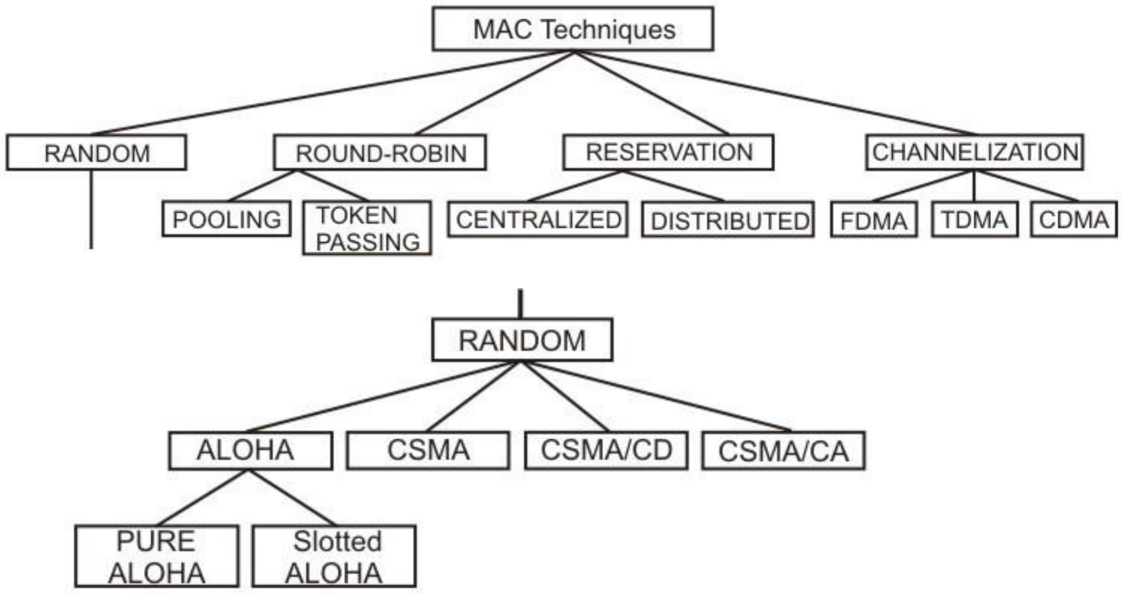
A network of computers based on multi-access medium requires a protocol for effective sharing of the media. As only one node can send or transmit signal at a time using the broadcast mode, the main problem here is how different nodes get control of the medium to send data, that is "who goes next?". The protocols used for this purpose are known as Medium Access Control (MAC) techniques. The key issues involved here are - Where and How the control is exercised.

'Where' refers to whether the control is exercised in a centralised or distributed manner. In a centralized system a master node grants access of the medium to other nodes. A centralized scheme has a number of advantages as mentioned below:

- Greater control to provide features like priority, overrides, and guaranteed bandwidth.
- Simpler logic at each node.
- Easy coordination.

Although this approach is easier to implement, it is vulnerable to the failure of the master node and reduces efficiency. On the other hand, in a distributed approach all the nodes collectively perform a medium access control function and dynamically decide which node to be granted access. This approach is more reliable than the former one.

'How' refers to in what manner the control is exercised. It is constrained by the topology and tradeoff between cost-performance and complexity. Various approaches for medium access control are shown in Fig. below. The MAC techniques can be broadly divided into four categories; *Contention-based*, *Round-Robin*, *Reservation-based* and. *Channelization-based*. Under these four broad categories there are specific techniques, as shown in Fig.



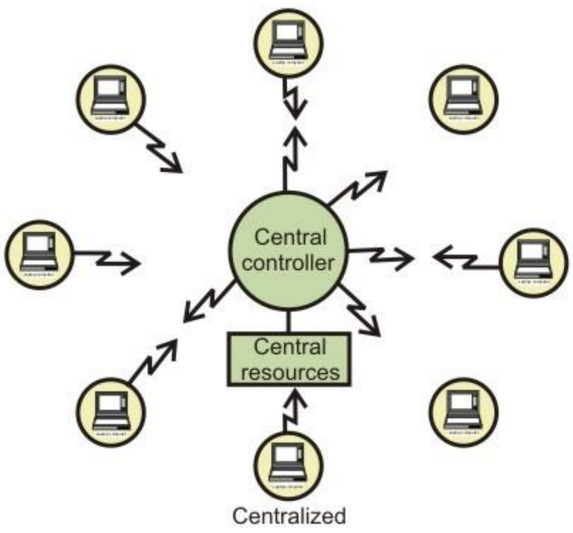
Round Robin Techniques

In Round Robin techniques, each and every node is given the chance to send or transmit by rotation. When a node gets its turn to send, it may either decline to send, if it has no data or may send if it has got data to send. After getting the opportunity to send, it must relinquish its turn after some maximum period of time. The right to send then passes to the next node based on a predetermined logical sequence. The right to send may be controlled in a centralised or distributed manner. *Polling* is an example of centralised control and *token passing* is an example of distributed control as discussed below.

Polling

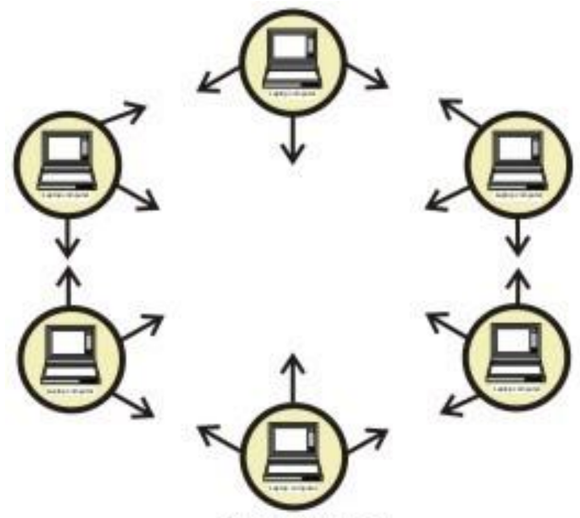
The mechanism of polling is similar to the roll-call performed in a classroom. Just like the teacher, a controller sends a message to each node in turn. The message contains the address of the node being selected for granting access. Although all nodes receive the message, only the addressed node responds and then it sends data, if any. If there is no data, usually a *"poll reject"* message is sent back. In this way, each node is interrogated in a round-robin fashion, one after the other, for granting access to the medium. The first node is again polled when the controller finishes with the remaining codes.

The polling scheme has the flexibility of either giving equal access to all the nodes, or some nodes may be given higher priority than others. In other words, priority of access can be easily implemented.



Polling can be done using a central controller, which may use a frequency band to send outbound messages as shown in Fig. Other stations share a different frequency to send inbound messages. The technique is called frequency-division duplex approach (FDD). Main drawbacks of the polling scheme are high overhead of the polling messages and high dependence on the reliability of the controller.

Polling can also be accomplished without a central controller. Here, all stations receive signals from other stations as shown in Fig. Stations develop a polling order list, using some protocol.

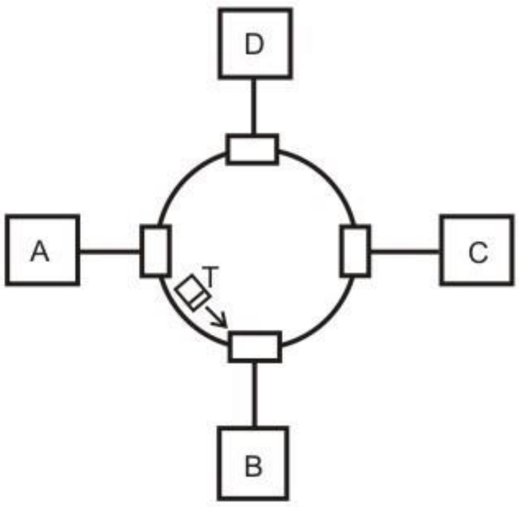


Distributed

Token Passing

In token passing scheme, all stations are logically connected in the form of a ring and control of the access to the medium is performed using a *token*. A *token* is a special bit pattern or a small packet, usually several bits in length, which circulate from node to node. Token passing can be used with both broadcast (token bus) and sequentially connected (token ring) type of networks with some variation in the details as considered in the next lesson.

In case of token ring, token is passed from a node to the physically adjacent node. On the other hand, in the token bus, token is passed with the help of the address of the nodes, which form a logical ring. In either case a node currently holding the token has the 'right to transmit'. When it has got data to send, it removes the token and transmits the data and then forwards the token to the next logical or physical node in the ring. If a node currently holding the token has no data to send, it simply forwards the token to the next node. The token passing scheme is efficient compared to the polling technique, but it relies on the correct and reliable operation of all the nodes. There exists a number of potential problems, such as *lost token*, *duplicate token*, *and insertion of a node*, *removal of a node*, which must be tackled for correct and reliable operation of this scheme.



Contention-based Approaches

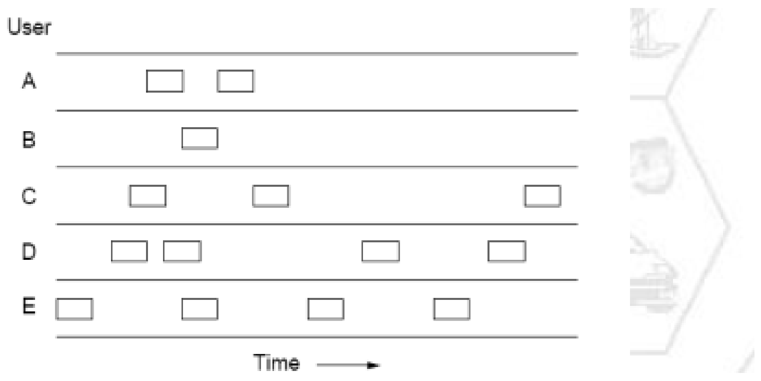
Round-Robin techniques work efficiently when majority of the stations have data to send most of the time. But, in situations where only a few nodes have data to send for brief periods of time, Round-Robin techniques are unsuitable. Contention techniques are suitable for bursty nature of traffic. In contention techniques, there is no centralised control and when a node has data to send, it contends for gaining control of the medium. The principle advantage of contention techniques is their simplicity. They can be easily implemented in each node. The techniques work efficiently under light to moderate load, but performance rapidly falls under heavy load.

ALOHA

ALOHA, the earliest random access method, was developed at the University of Hawaiiin early 1970. It was designed for a radio (wireless) LAN, but it can be used on anyshared medium.It is obvious that there are potential collisions in this arrangement. The medium is shared between the stations. When a station sends data, another station may attempt to do so at the same time. The data from the two stations collide and become garbled.

Pure ALOHA

The original ALOHA protocol is called pure ALOHA. This is a simple, but elegant protocol. The idea is that each station sends a frame whenever it has a frame to send. However, since there is only one channel to share, there is the possibility of collision between frames from different stations. In pure ALOHA, frames are transmitted at completely arbitrary times. It is obvious that we need to resend the frames that have been destroyed during transmission. The pure ALOHA protocol relies on acknowledgments from the receiver. When a station sends a frame, it expects the receiver to send an acknowledgment. If the acknowledgment does not arrive after a time-out period, the station assumes that the frame (or the acknowledgment) has been destroyed and resends the frame. A collision involves two or more stations. If all these stations try to resend their frames after the time-out, the frames will collide again. Pure ALOHA dictates that when the time-out period passes, each station waits a random amount of time before resending its frame. The randomness will help avoid more collisions.



Slotted ALOHA

Pure ALOHA has a vulnerable time of 2 x Tfr . This is so because there is no rule that defines when the station can send. A station may send soon after another station has started or soon before another station has finished. Slotted ALOHA was invented to improve the efficiency of pure ALOHA. In slotted ALOHA we divide the time into slots of Tfr s and force the station to send only at the beginning of the time slot.

CSMA

The poor efficiency of the ALOHA scheme can be attributed to the fact that a node start transmission without paying any attention to what others are doing. In situations where propagation delay of the signal between two nodes is small compared to the transmission time of a packet, all other nodes will know very quickly when a node starts transmission. This observation is the basis of the carrier-sense multiple-access (CSMA) protocol. In this scheme, a node having data to transmit first listens to the medium to check whether another transmission is in progress or not. The node starts sending only when the channel is free, that is there is no carrier. That is why the scheme is also known as listen-before talk. There are three variations of this basic scheme as outlined below.

- 1-persistent CSMA: In this case, a node having data to send, start sending, if the channel is sensed free. If the medium is busy, the node continues to monitor until the channel is idle. Then it starts sending data.
- Non-persistent CSMA: If the channel is sensed free, the node starts sending the packet. Otherwise, the node waits for a random amount of time and then monitors the channel.
- p-persistent CSMA: If the channel is free, a node starts sending the packet. Otherwise the node continues to monitor until the channel is free and then it sends with probability p.

CSMA/CD

CSMA/CD protocol can be considered as a refinement over the CSMA scheme. It has evolved to overcome one glaring inefficiency of CSMA. In CSMA scheme, when two packets collide the channel remains unutilized for the entire duration of transmission time of both the packets. If the propagation time is small (which is usually the case) compared to the packet transmission time, wasted channel capacity can be considerable. This wastage of channel capacity can be reduced if the nodes continue to monitor the channel while transmitting a packet and immediately cease transmission when collision is detected. This refined scheme is known as Carrier Sensed Multiple Access with Collision Detection (CSMA/CD) or Listen-While-Talk.

On top of the CSMA, the following rules are added to convert it into CSMA/CD:

- If a collision is detected during transmission of a packet, the node immediately ceases transmission and it transmits jamming signal for a brief duration to ensure that all stations know that collision has occurred.
- After transmitting the jamming signal, the node waits for a random amount of time and then transmission is resumed.

The random delay ensures that the nodes, which were involved in the collision are not likely to have a collision at the time of retransmissions. To achieve stability in the back off scheme, a technique known as binary exponential back off is used. A node will attempt to transmit repeatedly in the face of repeated collisions, but after each collision, the mean value of the random delay is doubled. After 15 retries (excluding the original try), the unlucky packet is discarded and the node reports an error.