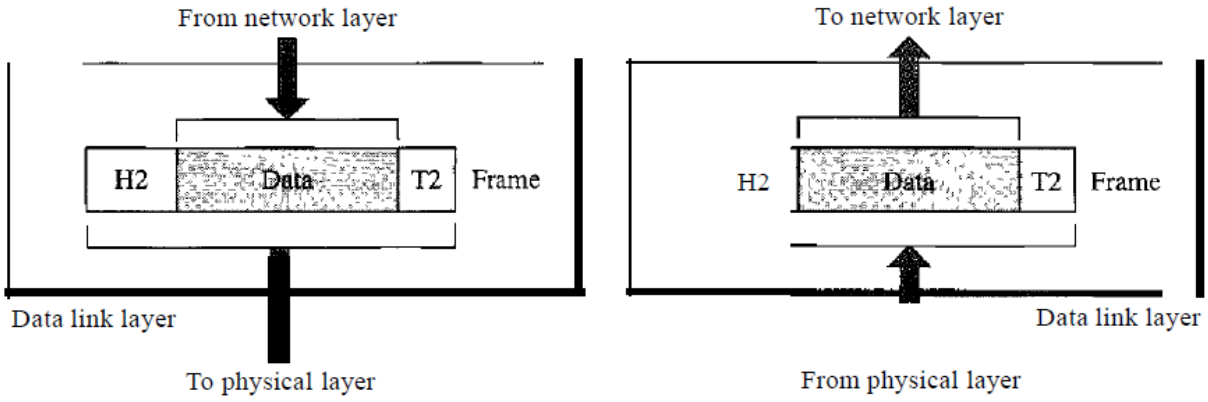
mood-book

Unit- II – Data Link Layer

- The data link layer transforms the physical layer, a raw transmission facility, to a link responsible for node-to-node (hop-to-hop) communication.
- Specific responsibilities of the data link layer include framing, addressing, flow control, error control, and media access control.
- The data link layer divides the stream of bits received from the network layer into manageable data units called frames. The data link layer adds a header to the frame to define the addresses of the sender and receiver of the frame.
- If the rate at which the data are absorbed by the receiver is less than the rate at which data are produced in the sender, the data link layer imposes a flow control mechanism to avoid overwhelming the receiver.
- The data link layer also adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged, duplicate, or lost frames.
- When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has control over the link at any given time.

The data link layer transforms the physical layer, and raw information's is transmitted to a reliable link. It makes physical layer appear error – free to the upper layer i.e. network layer.



Above figure shows that relationship between the data link layer to n/w layer and physical layer.

The responsibility of the data link layer is to move frames one hop (node) to the next hop.

The data link layer uses the services of the physical layer to send and receive bits over the communication channel.

1. It has to provide well – defined service interface to network layer.
2. Dealing with transmission errors
3. Regulating the flow of data so that slow receiver are not swamped by the fast senders.

Framing

The data link layer divides the stream of bits received from the n/w layer into manageable data units called frames.

Each frame contains a head, a pay load field for holding the packet a frame trainer.

Physical Addressing

If frames are to be distributed to different systems on the network, the data link layer adds a header to the frame to define the sender and or receiver of the frame.

Flow control

When a sender systematically wants to send frames faster than the receiver can accept them.

The rate at which the data are absorbed by the receiver less than the rate at which the data are produced in the sender

The data link layer imposes a flow control mechanism to avoid over whelming the receiver.

Error Control

The data link layer adds reliability to the physical layer by adding mechanism to defect and retransmit damaged or lost frames.

It also uses a mechanism to recognize duplicate frames

Error control is normally achieved through a trailer adder to the end of the frame.

Access Control

When two or more devices are connected to the same link, the data link layer protocols are necessary to determine which device has control over the link at any given time.

Error Detection and Correction:

Networks must be able to transfer data from one device to another with acceptable accuracy. For most applications, a system must guarantee that the data received are identical to the data transmitted. Any time data are transmitted from one node to the next, they can become corrupted in passage. Many factors can alter one or more bits of a message.

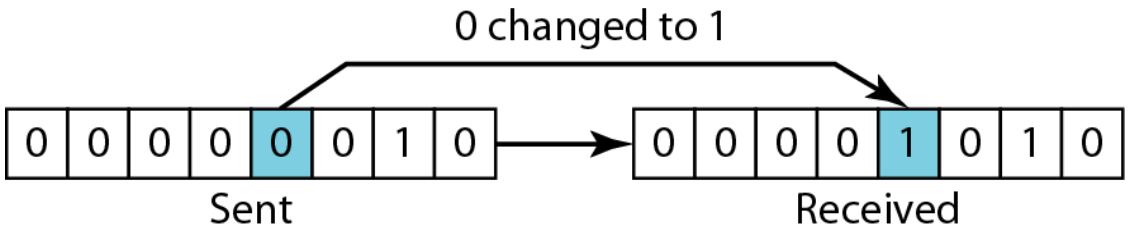
Data can be corrupted during transmission due to some applications it require that errors be detected and corrected.

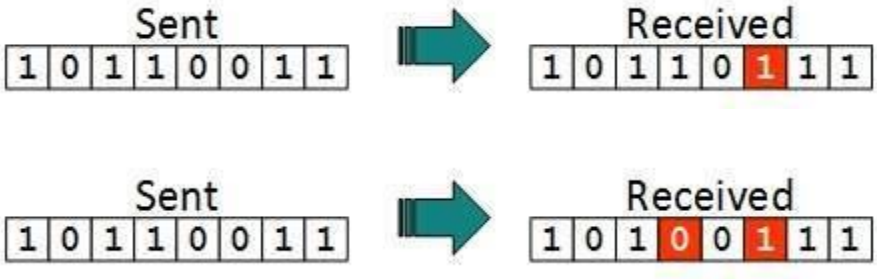
Types of Errors:

Whenever bits flow from one point to another, they are subject to unpredictable changes because of interference. This interference can change the shape of the signal.

Single Bit Error: In a single-bit error, a 0 is changed to a 1 or a 1 to a 0. In a burst error, multiple bits are changed. For example, a 11100 s burst of impulse noise on a transmission with a data rate of 1200 bps might change all or some of the12 bits of information.

The term single-bit error means that only 1 bit of a given data unit (such as a byte, character, or packet) is changed from 1 to 0 or from 0 to 1.

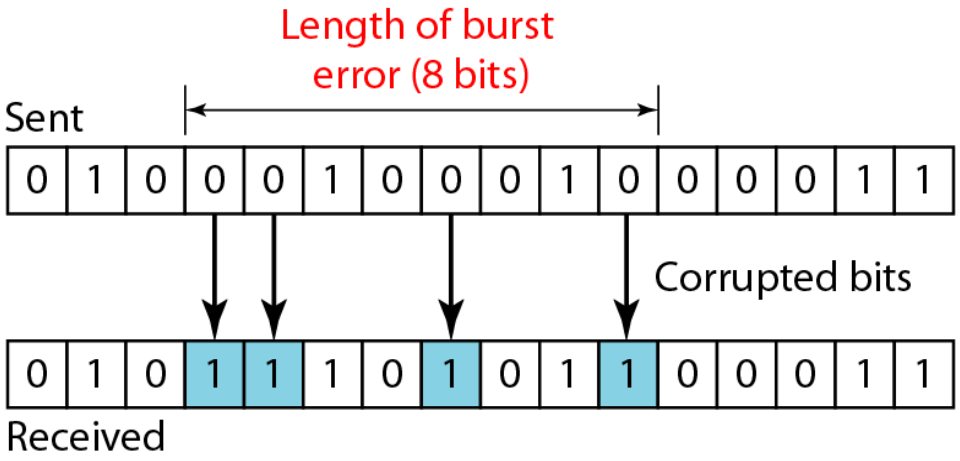




Burst Error: The term burst error means that 2 or more bits in the data unit have changed from 1 to 0 or from 0 to 1.

It shows the effect of a burst error on a data unit. In this case, 0100010001000011 was sent, but 0101110101100011 was received.

Note that a burst error does not necessarily mean that the errors occur in consecutive bits. The length of the burst is measured from the first corrupted bit to the last corrupted bit. Some bits in between may not have been corrupted.



Redundancy: The central concept in detecting or correcting errors is redundancy. To be able to detect or correct errors, we need to send some extra bits with our data. These redundant bits are added by the sender and removed by the receiver. Their presence allows the receiver to detect or correct corrupted bits.

Detection Versus Correction:

The correction of errors is more difficult than the detection. In error detection, we are looking only to see if any error has occurred. The answer is a simple yes or no. We are not even interested in the number of errors. A single-bit error is the same for us as a burst error. In error correction, we need to know the exact number of bits that are corrupted and more importantly, their location in the message. The number of the errors and the size of the message are important factors. If we need to correct one single error in an 8-bit data unit, we need to consider eight possible error locations.

Forward Error Correction Versus Retransmission:

There are two main methods of error correction. Forward error correction is the process in which the receiver tries to guess the message by using redundant bits. This is possible, if the number of errors is small.

Correction by retransmission is a technique in which the receiver detects the occurrence of an error and asks the sender to resend the message. Resending is repeated until a message arrives that the receiver believes is error-free.

Error control mechanism may involve two possible ways:

- Error detection
- Error correction

Error Detection:

Errors in the received frames are detected by means of Parity Check and Cyclic Redundancy Check (CRC). In both cases, few extra bits are sent along with actual data to confirm that bits received at other end are same as they were sent. If the counter-check at receiver' end fails, the bits are considered corrupted.

Parity Check:

One extra bit is sent along with the original bits to make number of 1s either even in case of even parity, or odd in case of odd parity.

The sender while creating a frame counts the number of 1s in it. For example, if even parity is used and number of 1s is even then one bit with value 0 is added. This way number of 1s remains even. If the number of 1s is odd, to make it even a bit with value 1 is added.

Even Parity:

The receiver simply counts the number of 1s in a frame. If the count of 1s is even and even parity is used, the frame is considered to be not-corrupted and is accepted. If the count of 1s is odd and odd parity is used, the frame is still not corrupted.

If a single bit flips in transit, the receiver can detect it by counting the number of 1s. But when more than one bits are erroneous, then it is very hard for the receiver to detect the error.

Checksum:

This is a block code method where a checksum is created based on the data values in the data blocks to be transmitted using some algorithm and appended to the data. When the receiver gets this data, a new checksum is calculated and compared with the existing checksum. A non-match indicates an error.

Error Detection by Checksums

For error detection by checksums, data is divided into fixed sized frames or segments.

Sender's End − The sender adds the segments using 1's complement arithmetic to get the sum. It then complements the sum to get the checksum and sends it along with the data frames.

Receiver's End − The receiver adds the incoming segments along with the checksum using 1's complement arithmetic to get the sum and then complements it.

If the result is zero, the received frames are accepted; otherwise they are discarded.

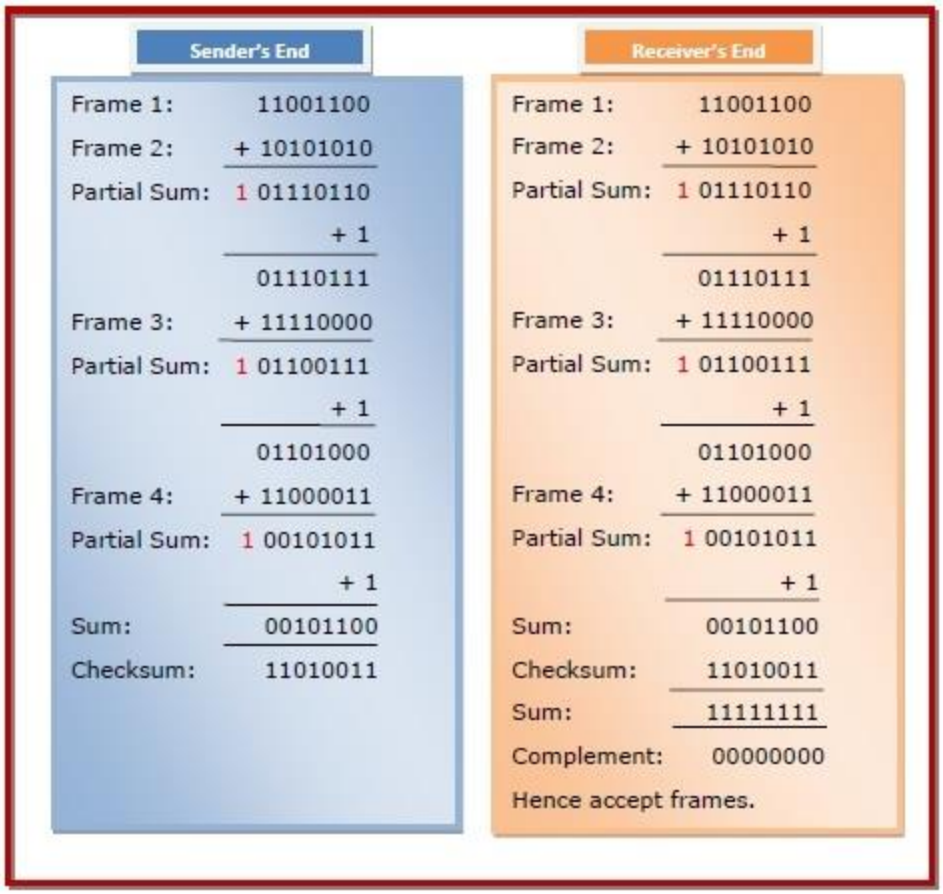
Example:

Suppose that the sender wants to send 4 frames each of 8 bits, where the frames are 11001100, 10101010, 11110000 and 11000011.

The sender adds the bits using 1s complement arithmetic. While adding two numbers using 1s complement arithmetic, if there is a carry over, it is added to the sum.

After adding all the 4 frames, the sender complements the sum to get the checksum, 11010011, and sends it along with the data frames.

The receiver performs 1s complement arithmetic sum of all the frames including the checksum. The result is complemented and found to be 0. Hence, the receiver assumes that no error has occurred.



Cyclic Redundancy Check (CRC):

CRC is a different approach to detect if the received frame contains valid data. This technique involves binary division of the data bits being sent. The divisor is generated using polynomials. The sender performs a division operation on the bits being sent and calculates the remainder. Before sending the actual bits, the sender adds the remainder at the end of the actual bits. Actual data bits plus the remainder is called a codeword. The sender transmits data bits as codewords.

At the other end, the receiver performs division operation on codewords using the same CRC divisor. If the remainder contains all zeros the data bits are accepted, otherwise it is considered as there some data corruption occurred in transit.

Cyclic Redundancy Check Codes:

In order to send the data from sender to receiver, the data must be checked and preceded with zero errors to receiver.

CRC (cyclic redundancy check) is also called as polynomial code.

Polynomial codes are based on the treating bit string as representations of polynomials with co – efficient of 0 and 1 only.

It is used in networks such as LANS and WANS.

In sender or encoder, the data word has k bits (4 – bits) the code word has n bits (7 – bits) are given by as per the structure.

Data word is : 1 1 0 1

Code word is : 1 0 1 1 0 1 1

The size of the data word is appended by m – k (bits) 0s to the right hand side of the code word.

The CRC generator of data word is 4 – bits the (h-1) is the CRC generator by adding to code word.

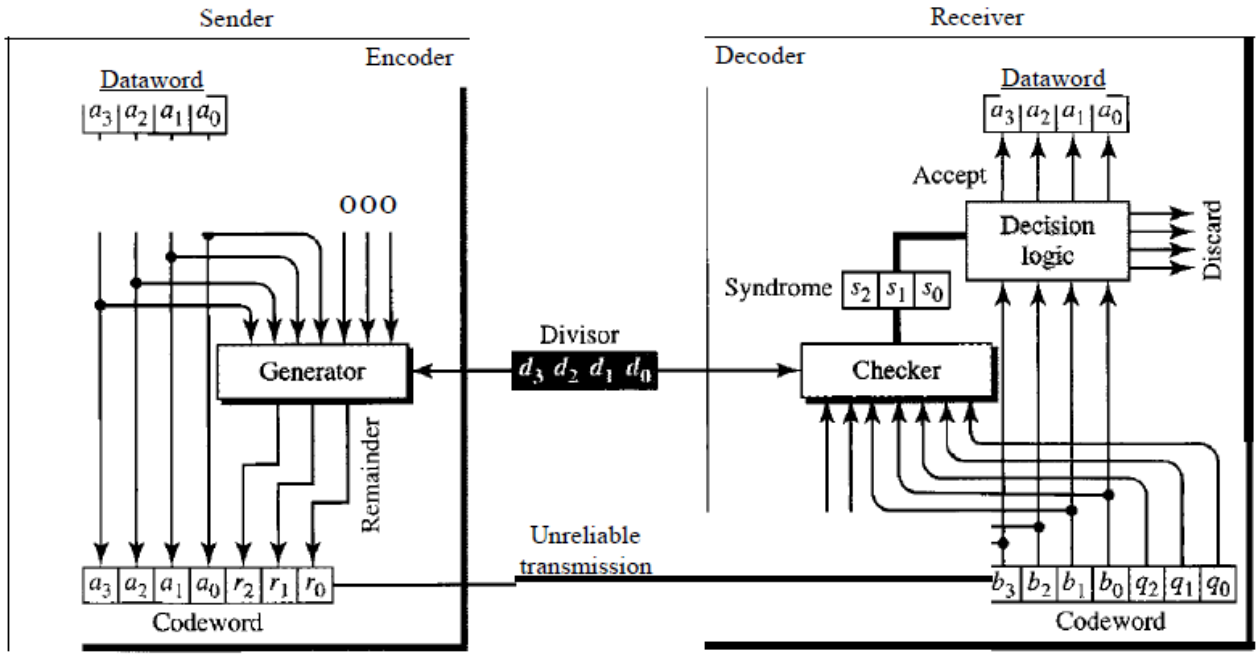
The code word is given by: 1 0 1 1 0 1 1 0 0 0

The generator divides the augmented data word by divisor (modulo – 2 divisions), the quotient of the division is discarded.

The reminder is appended to the data word to create the code word.

A copy of all n – bits is fed to the checker which is replica of the generator.

The remainder produced by the checker is a syndrome of n – k (3 – bits) which is fed to decision logic analyzer.



The analyzer has to perform simple function like if the syndrome bits are all 0s, the 4 left most bits of the code word are accepted as the data word (interpreted as no error) otherwise the bits are non – zero mean (interpreted as error) and are discarded.

Example:

Perform CRC check for error force data transmission to receiver. Data word is: 1 1 0 1 and code word is given by: 1 0 1 1 0 1 1.

Solution:

Data word : 1 1 0 1 (k – bits)

Code word : 1 0 1 1 0 1 1 (n – bits)

Cyclic redundancy generators is: (k-1) = 0 0 0 (3 – bits)

The 3 – bits are append to the code word to check CRC

Here data word taken as divisor.

Perform a division operation with code word.

i.e., by modulo – 2 division it also represents EX – OR like operation

```
          1 1 1
        ---------
1 1 0 1) 1 0 1 1 0 1 1 0 0 0  (
         1 1 0 1
        ---------------------
          0 1 1 0 0 1 1 0 0 0
            1 1 0 1
          ---------------------
            0 0 0 1 1 1 0 0 0
                1 1 0 1
              -----------------
                  0 0 1 1 0 0
                    1 1 0 1
                  -----------
                    0 0 0 1   → CRC – bit
```

Case 1:

In the above operation last bit is '1' the CRC both is append to the code word to check error free transmission.

Checking at the receiver side

```
          1 1 1 1
        ---------
1 1 0 1) 1 0 1 1 0 1 1 0 0 1  (
         1 1 0 1
        ---------------------
          0 1 1 0 0 1 1 0 0 1
            1 1 0 1
          ---------------------
            0 0 0 1 1 1 0 0 1
                1 1 0 1
              -----------------
                  0 0 1 1 0 1
                    1 1 0 1
                  -----------
                    0 0 0 0   → Syndrome is '0'.
                  -----------
```

If code word have a corrupted bit, for such cases these is an single error may occur, which will be discarded.

Case 2:

```
                  1
1 1 0 1) 1 0   1   1 0 1 1 0 0 1   (
         0
```

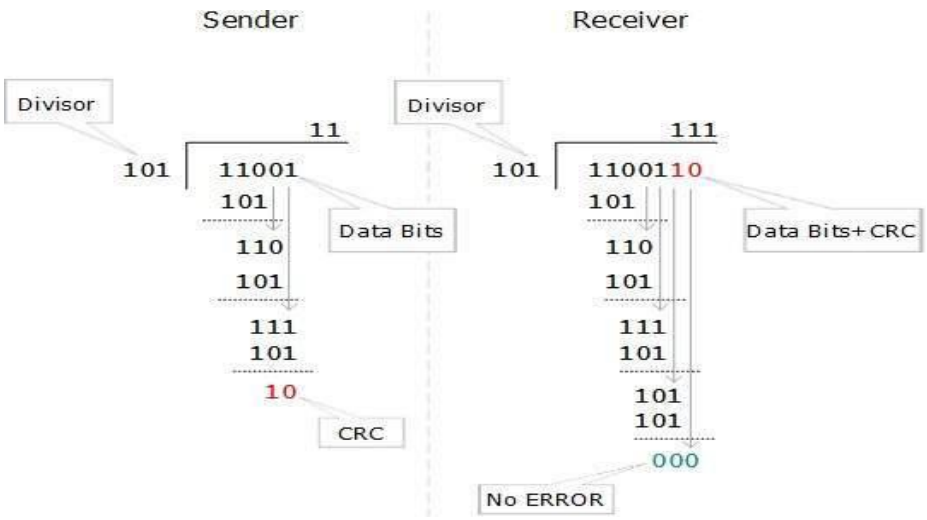
→ Corrupted bit – replaced as '0'

```
          1 1
1 1 0 1) 1 0 0 1 0 1 1 0 0 1   (
           1 1 0 1
         ---------------------
           0 1 0 0 0 1 1 0 0 1
             1 1 0 1
         ---------------------
           0 1 0 1 1 1 0 0 1
             1 1 0 1
         ---------------------
           0 1 1 0 1 0 0 1
             1 1 0 1
           -----------------
           0 0 0 0 0 0 0 1
```

→ Syndrome is '1'→ Error Bit

In this case data word not accepted, discarded due to non – zero CRC – bit occur.

For an successful transmission of data is selected by the checking of CRC must done with above manner.



Error Correction:

In the digital world, error correction can be done in two ways:

Backward Error Correction: When the receiver detects an error in the data received, it requests back the sender to retransmit the data unit.

Forward Error Correction: When the receiver detects some error in the data received, it executes error-correcting code, which helps it to auto-recover and to correct some kinds of errors.

The first one, Backward Error Correction, is simple and can only be efficiently used where retransmitting is not expensive. For example, fiber optics. But in case of wireless transmission retransmitting may cost too much. In the latter case, Forward Error Correction is used.

To correct the error in data frame, the receiver must know exactly which bit in the frame is corrupted. To locate the bit in error, redundant bits are used as parity bits for error detection. For example, we take ASCII words (7 bits data), then there could be 8 kind of information we need: first seven bits to tell us which bit is error and one more bit to tell that there is no error.

For m data bits, r redundant bits are used. r bits can provide 2r combinations of information. In m+r bit codeword, there is possibility that the r bits themselves may get corrupted. So the number of r bits used must inform about m+r bit locations plus no-error information, i.e. m+r+1.

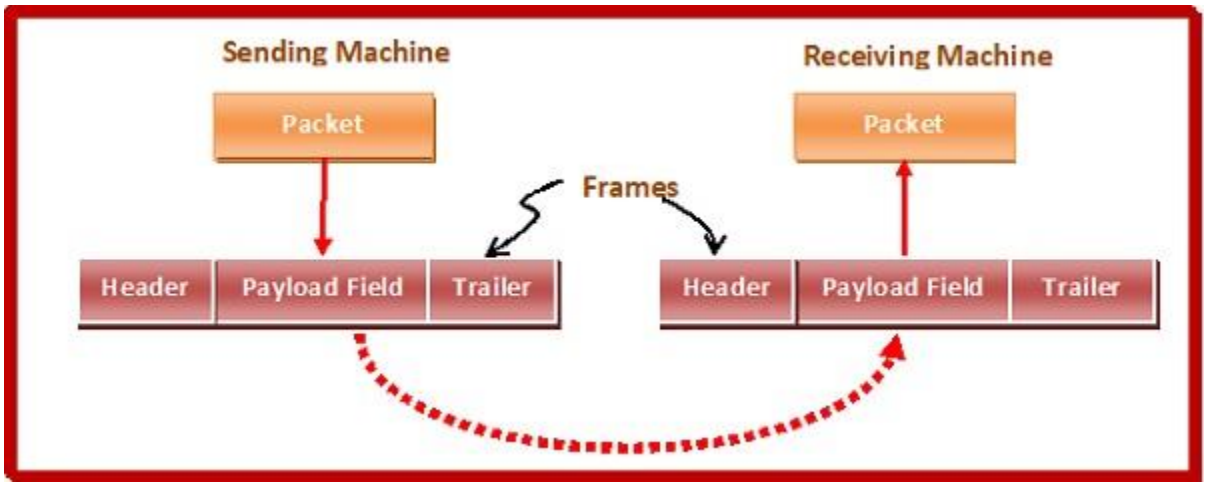
$$2^r >= m+r+1$$

Framing:

In the physical layer, data transmission involves synchronised transmission of bits from the source to the destination. The data link layer packs these bits into frames.

Data-link layer takes the packets from the Network Layer and encapsulates them into frames. If the frame size becomes too large, then the packet may be divided into small sized frames. Smaller sized frames makes flow control and error control more efficient.

Then, it sends each frame bit-by-bit on the hardware. At receiver's end, data link layer picks up signals from hardware and assembles them into frames.



Parts of a Frame

A frame has the following parts −

- Frame Header − It contains the source and the destination addresses of the frame.
- Payload field − It contains the message to be delivered.
- Trailer − It contains the error detection and error correction bits.
- Flag − It marks the beginning and end of the frame.

Types of Framing

Framing can be of two types, fixed sized framing and variable sized framing.

Fixed-sized Framing

Here the size of the frame is fixed and so the frame length acts as delimiter of the frame. Consequently, it does not require additional boundary bits to identify the start and end of the frame.

Example − ATM Wide area network cells.

Variable – Sized Framing

Here, the size of each frame to be transmitted may be different. So additional mechanisms are kept to mark the end of one frame and the beginning of the next frame.

It is used in local area networks.

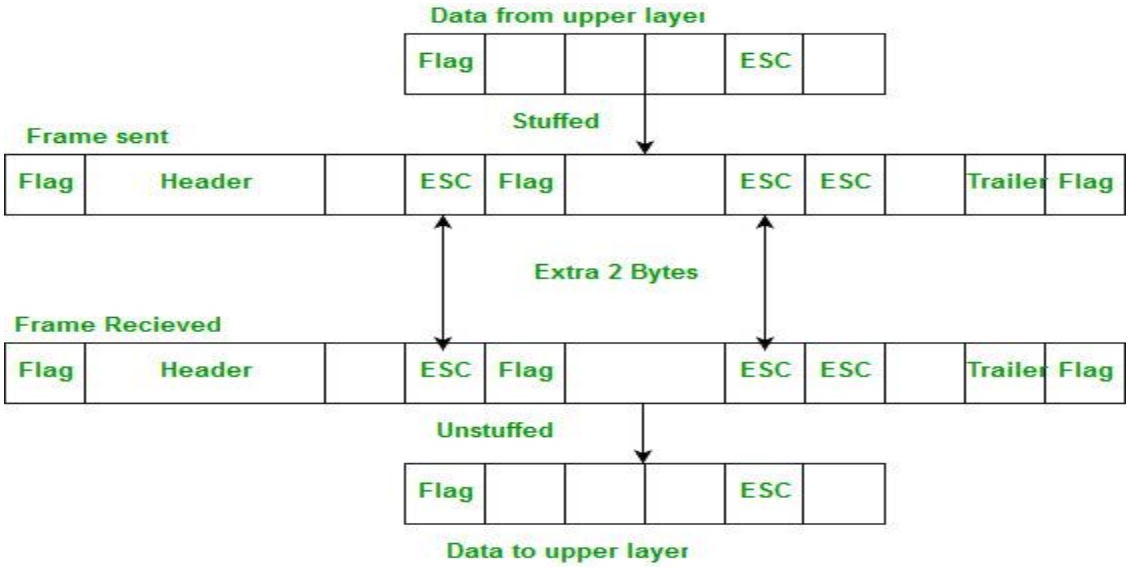
Two ways to define frame delimiters in variable sized framing are −

- **Length Field** − Here, a length field is used that determines the size of the frame. It is used in Ethernet (IEEE 802.3).
- **End Delimiter** − Here, a pattern is used as a delimiter to determine the size of frame. It is used in Token Rings. If the pattern occurs in the message, then two approaches are used to avoid the situation −

Byte – Stuffing − A byte is stuffed in the message to differentiate from the delimiter. This is also called character-oriented framing.

A byte (usually escape character (ESC)), which has a predefined bit pattern is added to the data section of the frame when there is a character with the same pattern as the flag. Whenever the receiver encounters the ESC character, it removes from the data section and treats the next character as data, not a flag.

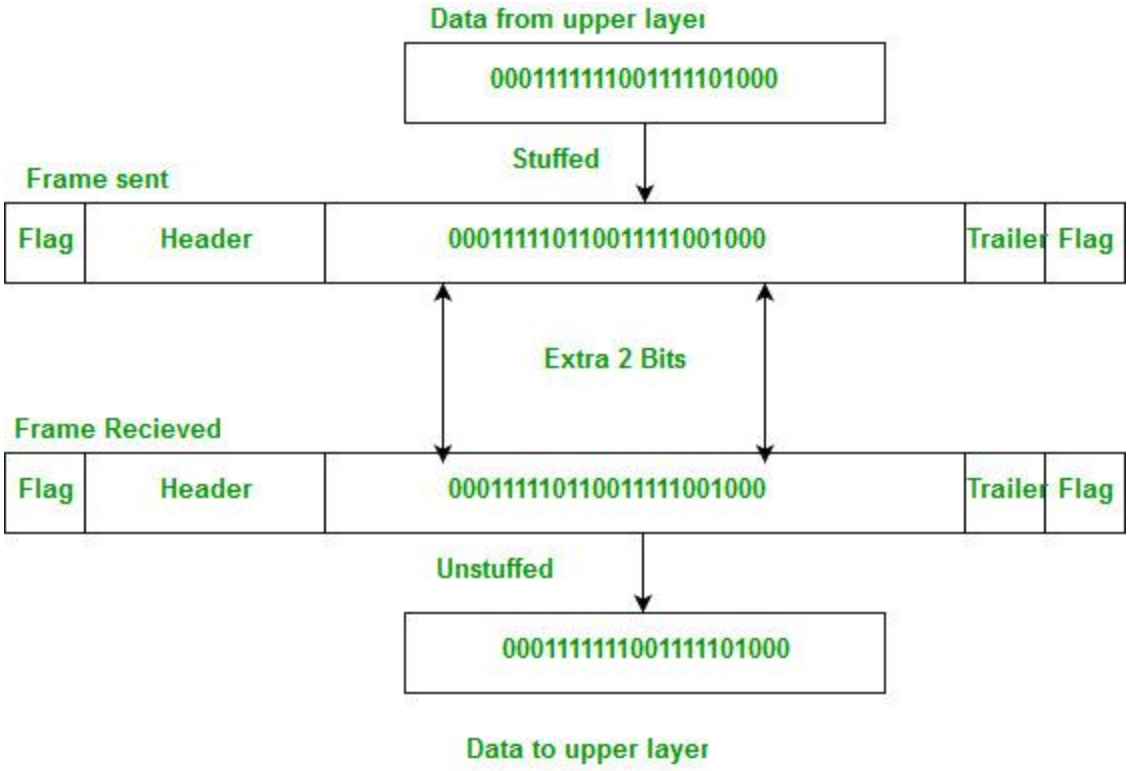
But the problem arises when the text contains one or more escape characters followed by a flag. To solve this problem, the escape characters that are part of the text are marked by another escape character i.e., if the escape character is part of the text, an extra one is added to show that the second one is part of the text.



Bit – Stuffing − A pattern of bits of arbitrary length is stuffed in the message to differentiate from the delimiter. This is also called bit – oriented framing.

Mostly flag is a special 8-bit pattern “01111110” used to define the beginning and the end of the frame.

Problem with the flag is the same as that was in case of byte stuffing. So, in this protocol what we do is, if we encounter 0 and five consecutive 1 bits, an extra 0 is added after these bits. This extra stuffed bit is removed from the data by the receiver.



The extra bit is added after one 0 followed by five 1 bits regardless of the value of the next bit. Also, as the sender side always knows which sequence is data and which is flag it will only add this extra bit in the data sequence, not in the flag sequence.

Data-link layer is responsible for implementation of point-to-point flow and error control mechanism.

Flow Control:

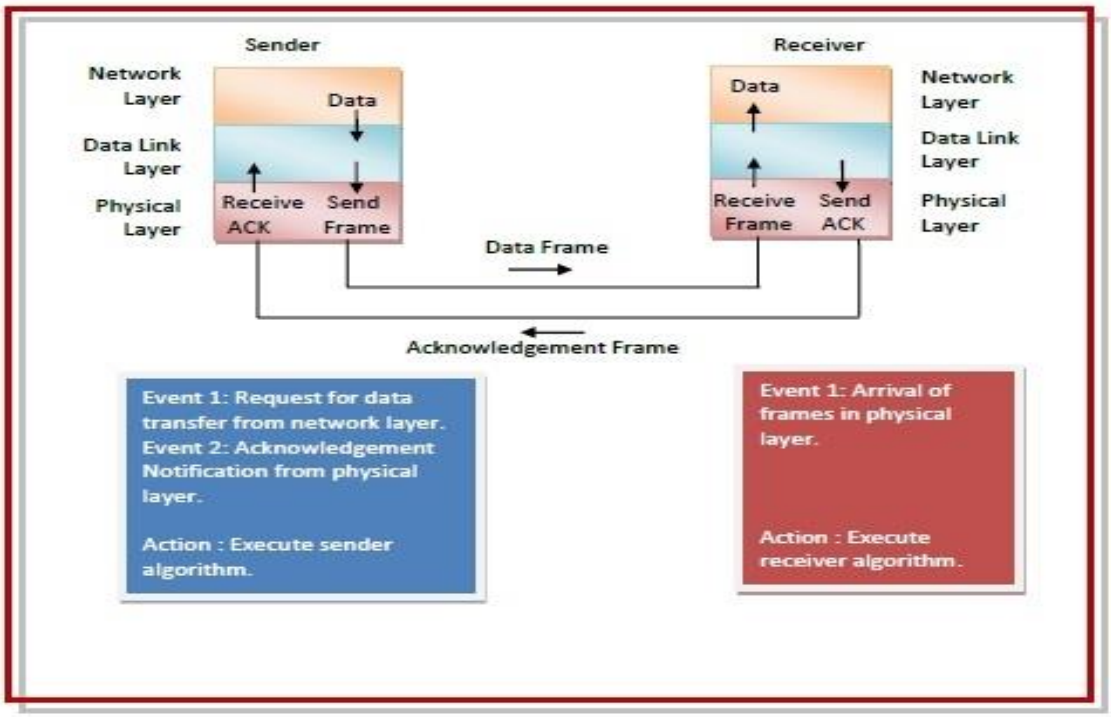
When a data frame (Layer-2 data) is sent from one host to another over a single medium, it is required that the sender and receiver should work at the same speed. That is, sender sends at a speed on which the receiver can process and accept the data. What if the speed (hardware/software) of the sender or receiver differs? If sender is sending too fast the receiver may be overloaded, (swamped) and data may be lost.

Simplex Stop – and – Wait protocol for noisy channel is data link layer protocol for data communications with error control and flow control mechanisms. It is popularly

known as Stop – and –Wait Automatic Repeat Request (Stop – and –Wait ARQ) protocol. It adds error control facilities to Stop – and – Wait protocol.

This protocol takes into account the facts that the receiver has a finite processing speed and that frames may get corrupted while transmission. If data frames arrive at the receiver's end at a rate which is greater than its rate of processing, frames can be dropped out. Also, frames may get corrupted or entirely lost when they are transmitted via network channels. So, the receiver sends an acknowledgment for each valid frame that it receives. The sender sends the next frame only when it has received a positive acknowledgment from the receiver that it is available for further data processing. Otherwise, it waits for a certain amount of time and then resends the frame.

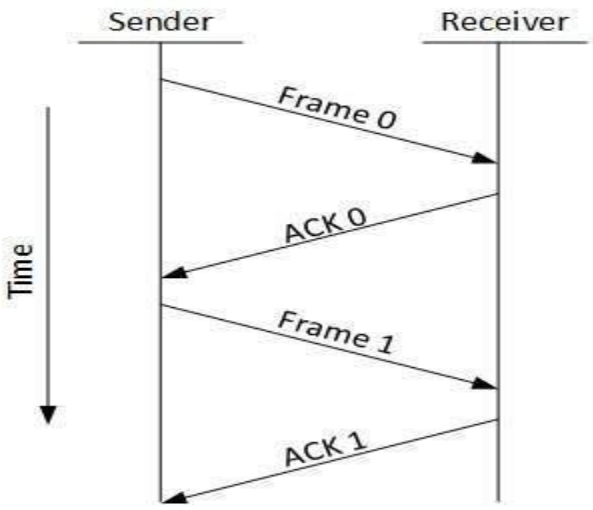
- **Sender Site** − At the sender site, a field is added to the frame to hold a sequence number. If data is available, the data link layer makes a frame with the certain sequence number and sends it. The sender then waits for arrival of acknowledgment for a certain amount of time. If it receives a positive acknowledgment for the frame with that sequence number within the stipulated time, it sends the frame with next sequence number. Otherwise, it resends the same frame.
- **Receiver Site** − The receiver also keeps a sequence number of the frames expected for arrival. When a frame arrives, the receiver processes it and checks whether it is valid or not. If it is valid and its sequence number matches the sequence number of the expected frame, it extracts the data and delivers it to the network layer. It then sends an acknowledgement for that frame back to the sender along with its sequence number.



Two types of mechanisms can be deployed to control the flow:

Stop and Wait

This flow control mechanism forces the sender after transmitting a data frame to stop and wait until the acknowledgement of the data-frame sent is received.



Sliding Window:

In this flow control mechanism, both sender and receiver agree on the number of data-frames after which the acknowledgement should be sent. As we learnt, stop and wait flow control mechanism wastes resources, this protocol tries to make use of underlying resources as much as possible.

Error Control:

When data-frame is transmitted, there is a probability that data-frame may be lost in the transit or it is received corrupted. In both cases, the receiver does not receive the correct data-frame and sender does not know anything about any loss.In such case, both sender and receiver are equipped with some protocols which helps them to detect transit errors such as loss of data-frame. Hence, either the sender retransmits the data-frame or the receiver may request to resend the previous data-frame.

Requirements for error control mechanism:

Error detection - The sender and receiver, either both or any, must ascertain that there is some error in the transit.

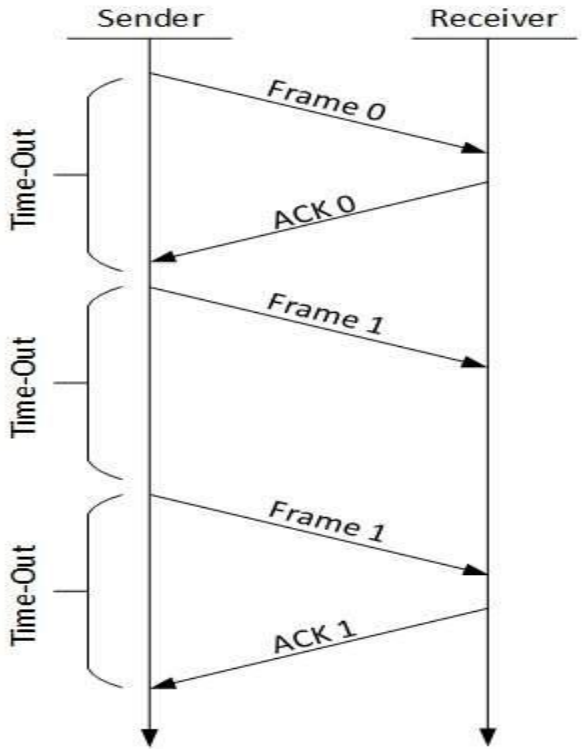
Positive ACK - When the receiver receives a correct frame, it should acknowledge it.

Negative ACK - When the receiver receives a damaged frame or a duplicate frame, it sends a NACK back to the sender and the sender must retransmit the correct frame.

Retransmission: The sender maintains a clock and sets a timeout period. If an acknowledgement of a data-frame previously transmitted does not arrive before the timeout the sender retransmits the frame, thinking that the frame or it's acknowledgement is lost in transit.

There are three types of techniques available which Data-link layer may deploy to control the errors by Automatic Repeat Requests (ARQ):

Stop-and-wait ARQ:

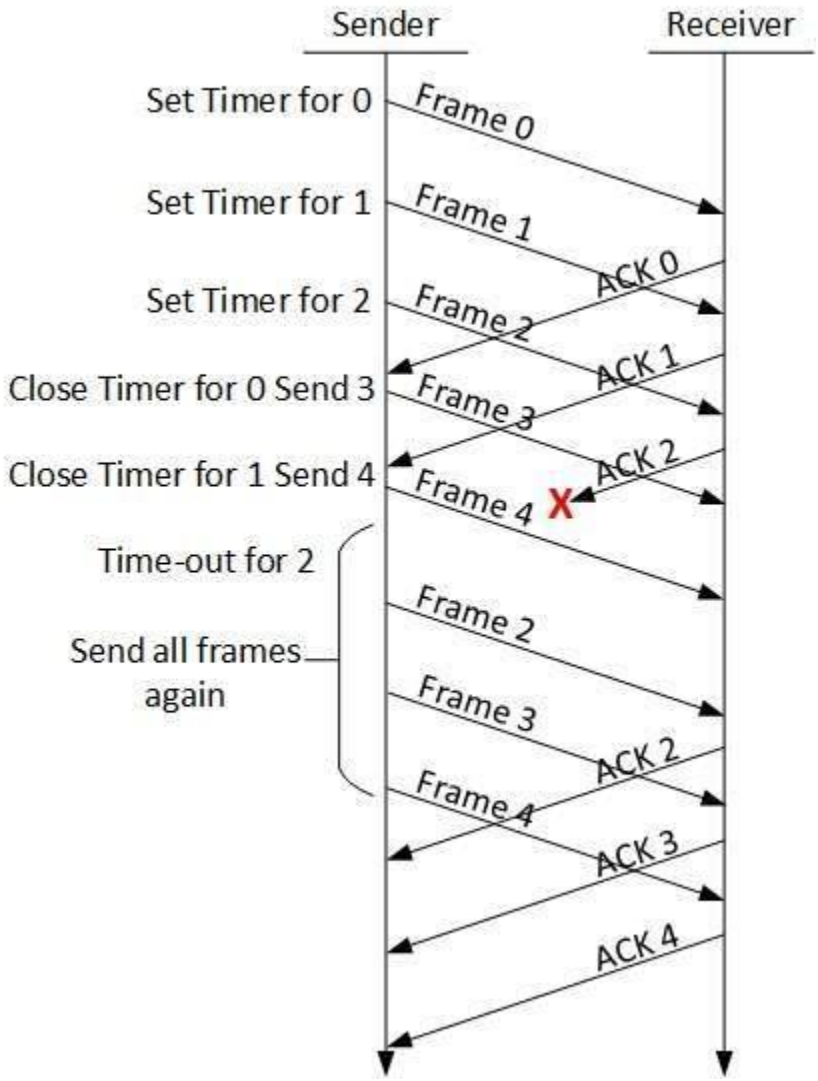


The following transition may occur in Stop-and-Wait ARQ:

- The sender maintains a timeout counter.
- When a frame is sent, the sender starts the timeout counter.
- If acknowledgement of frame comes in time, the sender transmits the next frame in queue.
- If acknowledgement does not come in time, the sender assumes that either the frame or its acknowledgement is lost in transit. Sender retransmits the frame and starts the timeout counter.
- If a negative acknowledgement is received, the sender retransmits the frame.

Go-Back-N ARQ:

Stop and wait ARQ mechanism does not utilize the resources at their best. When the acknowledgement is received, the sender sits idle and does nothing. In Go-Back-N ARQ method, both sender and receiver maintain a window.

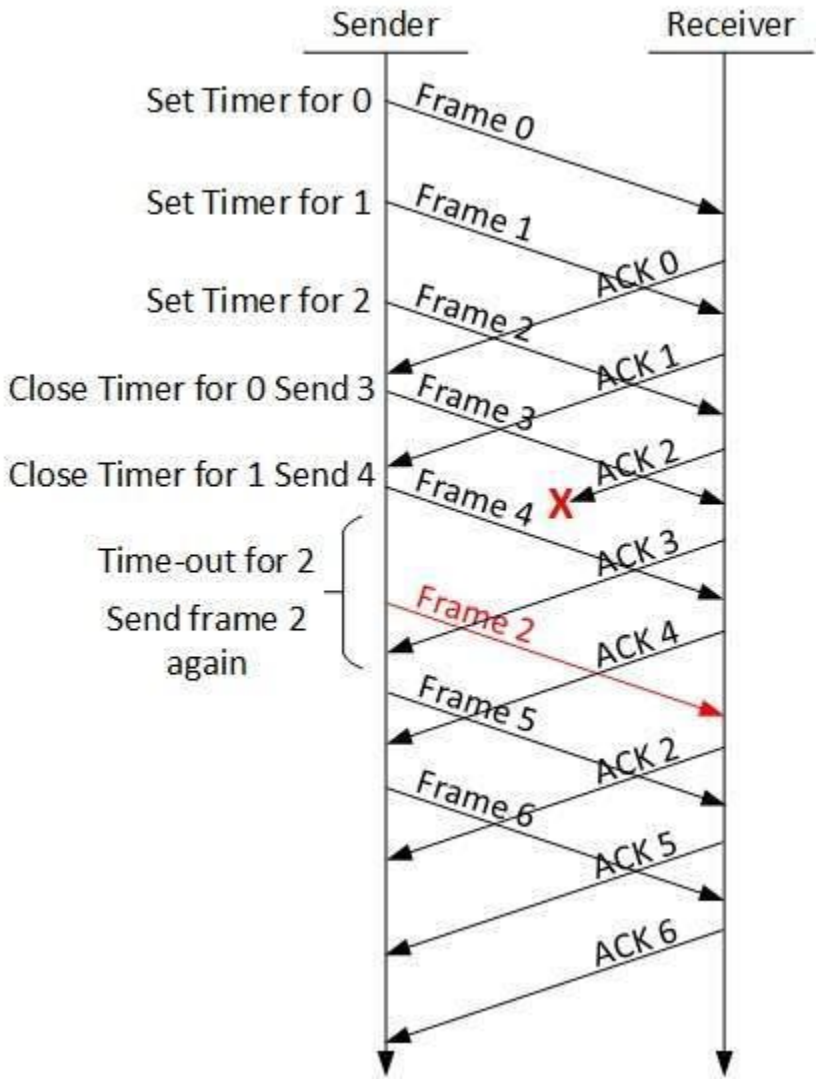


The sending-window size enables the sender to send multiple frames without receiving the acknowledgement of the previous ones. The receiving-window enables the receiver to receive multiple frames and acknowledge them. The receiver keeps track of incoming frame's sequence number.

When the sender sends all the frames in window, it checks up to what sequence number it has received positive acknowledgement. If all frames are positively acknowledged, the sender sends next set of frames. If sender finds that it has received NACK or has not receive any ACK for a particular frame, it retransmits all the frames after which it does not receive any positive ACK.

Selective Repeat ARQ:

In Go-back-N ARQ, it is assumed that the receiver does not have any buffer space for its window size and has to process each frame as it comes. This enforces the sender to retransmit all the frames which are not acknowledged.



In Selective-Repeat ARQ, the receiver while keeping track of sequence numbers, buffers the frames in memory and sends NACK for only frame which is missing or damaged.

The sender in this case, sends only packet for which NACK is received.

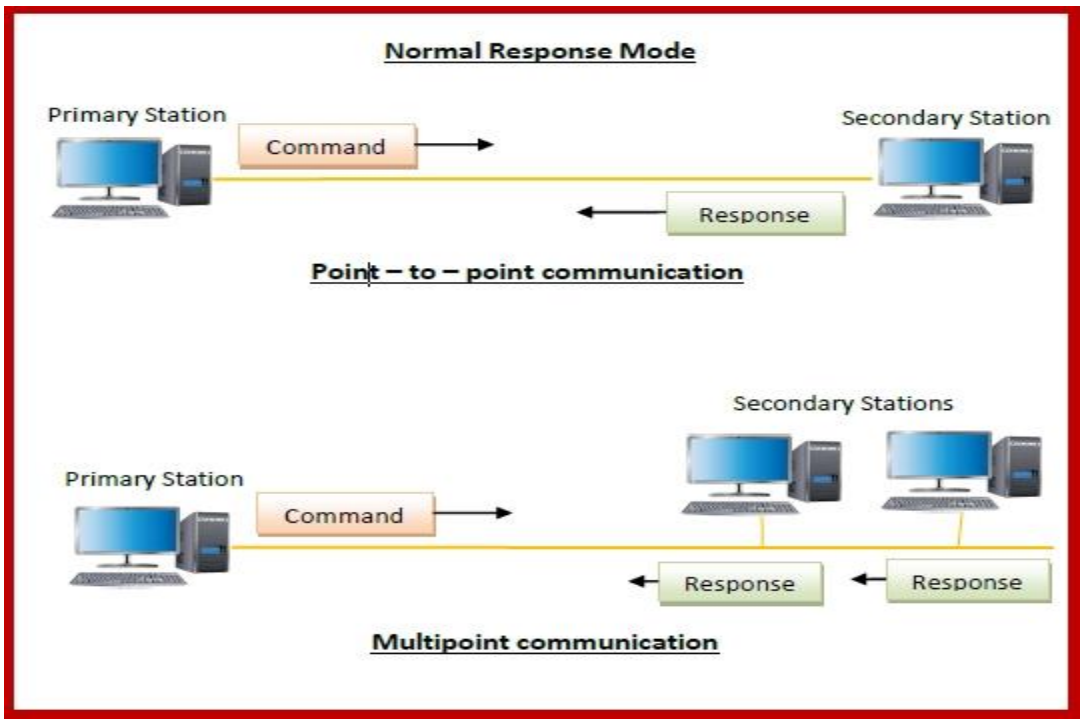
High-level Data Link Control (HDLC) :

It is a group of communication protocols of the data link layer for transmitting data between network points or nodes. Since it is a data link protocol, data is organized into frames. A frame is transmitted via the network to the destination that verifies its successful arrival. It is a bit - oriented protocol that is applicable for both point - to - point and multipoint communications.

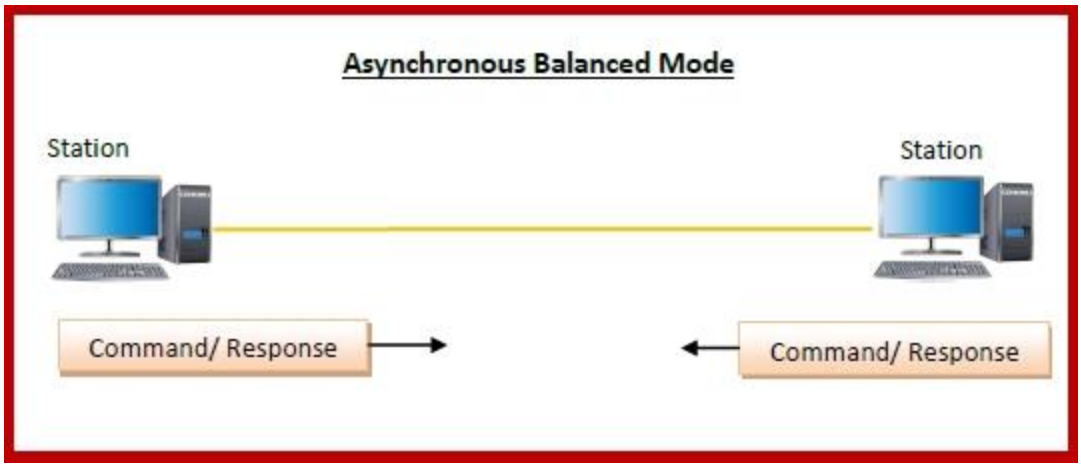
Transfer Modes

HDLC supports two types of transfer modes, normal response mode and asynchronous balanced mode.

- **Normal Response Mode (NRM)** – Here, two types of stations are there, a primary station that send commands and secondary station that can respond to received commands. It is used for both point - to - point and multipoint communications.



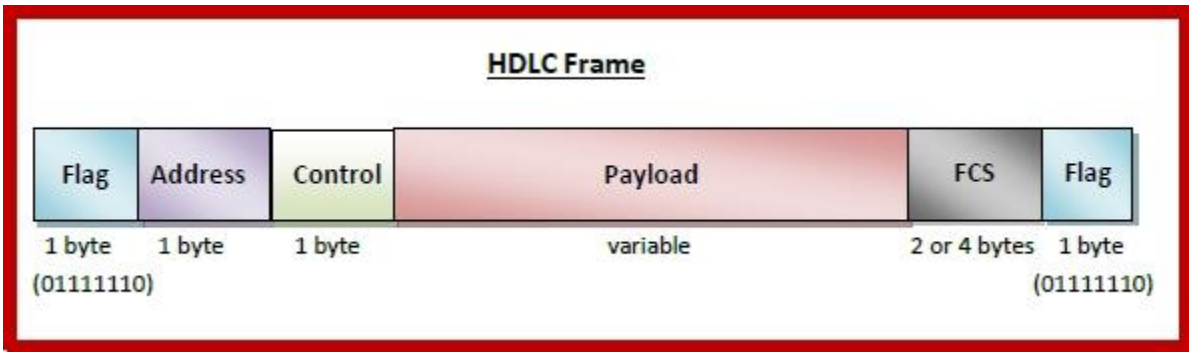
- **Asynchronous Balanced Mode (ABM)** – Here, the configuration is balanced, i.e. each station can both send commands and respond to commands. It is used for only point - to - point communications.



HDLC Frame

HDLC is a bit - oriented protocol where each frame contains up to six fields. The structure varies according to the type of frame. The fields of a HDLC frame are −

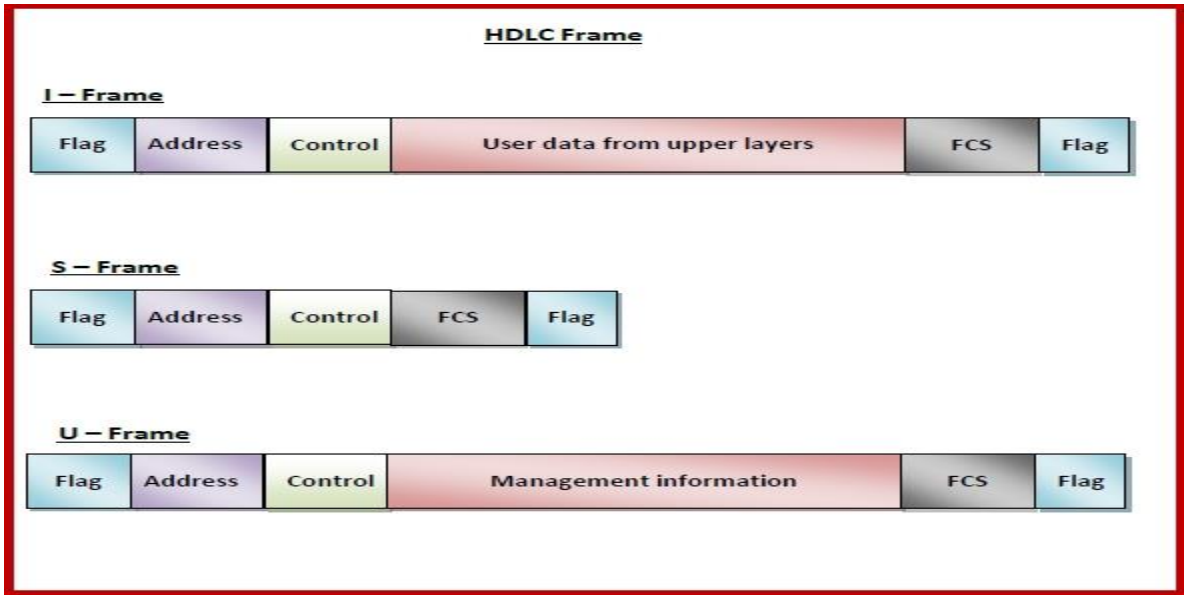
- **Flag** − It is an 8-bit sequence that marks the beginning and the end of the frame. The bit pattern of the flag is 01111110.
- **Address** − It contains the address of the receiver. If the frame is sent by the primary station, it contains the address(es) of the secondary station(s). If it is sent by the secondary station, it contains the address of the primary station. The address field may be from 1 byte to several bytes.
- **Control** − It is 1- or 2-bytes containing flow and error control information.
- **Payload** − This carries the data from the network layer. Its length may vary from one network to another.
- **FCS** − It is a 2 byte or 4 bytes frame check sequence for error detection. The standard code used is CRC (cyclic redundancy code)



Types of HDLC Frames

There are three types of HDLC frames. The type of frame is determined by the control field of the frame −

- **I-frame** – I-frames or Information frames carry user data from the network layer. They also include flow and error control information that is piggybacked on user data. The first bit of control field of I-frame is 0.
- **S-frame** – S-frames or Supervisory frames do not contain information field. They are used for flow and error control when piggybacking is not required. The first two bits of control field of S-frame is 10.
- **U-frame** – U-frames or Un-numbered frames are used for myriad miscellaneous functions, like link management. It may contain an information field, if required. The first two bits of control field of U-frame is 11.



Multiple Access Protocol:

In LAN's the back bone is a street channel (or) transmission link, which provides all users to access the transmission facility. It may be possible that two (or) more stations transmitting simultaneously, causing their signals to interfere and becomes garbled.

In order to resolve this conflict no. of different control mechanism or access protocols have been given.

Access the medium from many entry prints is called contention. It is controlled with in contention protocol.

In a random-access method, each station has the right to the medium without being controlled by other station.

However if more than one station tries to send, there is an access conflict. i.e. collision and the frames will be either destroyed (or) modified.

The random access techniques are given by

1. ALOHA
2. Carrier sense multiple access
3. CSNAA/CD (CD – collision detection)

1. ALOHA:

The ALOHA protocol was developed at the university of Hawaii in the earth 1970's. ALOHA was developed for packet radio networks it is used in any shared transmission medium.

In this multiple users try to send messages to other stations through a common broad cast medium i.e random access or contention technique.

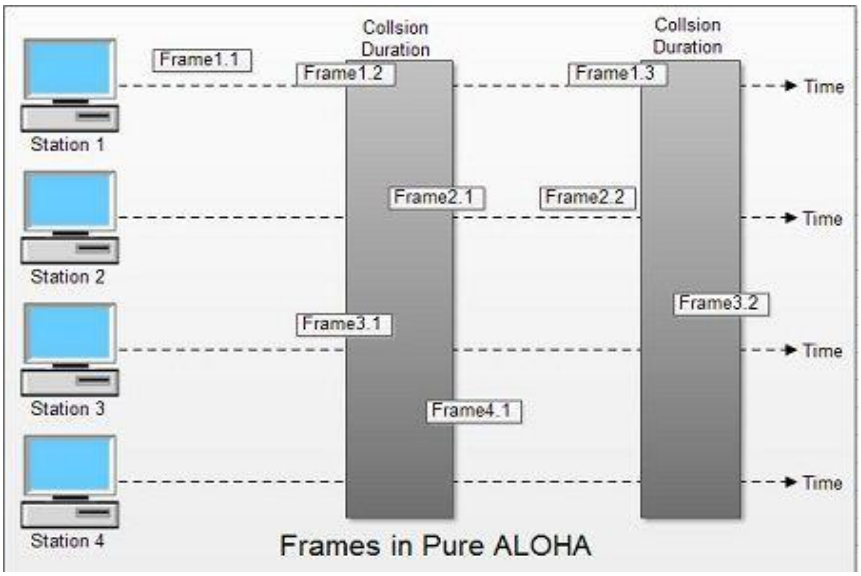
When a station sends data, another station may attempt to do so at the same time. The data from the two stations collide and becomes garbled

If two signals collided so, each station would simply wait a random time and try again.

PUREALOHA:

The original ALOHA protocol is called pure ALOHA. it is a simple protocol, the idea given by the each station sends a frame wherever it has a frame to send.

Since there is only one channel to share, there is possibility of collision between frames from different stations. The below figure shows that frame collisions in pure ALOHA



The pure ALOHA relies on acknowledgements from the receiver. When a user sends a frame, it expects the receiver to send an acknowledgment.

If acknowledgement does not arrive after a time out period, the station assumes that the frame has been destroyed and resends the frames.

Whenever two frames try to occupy the channel at the same time, there will be collision and both will be garbled. If the first bit of a new frame over laps with just the last bit of a frame almost finished, both frames will be totally destroyed and both will be transmitted again.

If all users try to send their frames after the time out the frames will collide again. Pure ALOHA dictates that when the time out period passes, each user waits a random amount of time before resending its frames.

The randomness will help to avoid more collisions. This is called back – off time (T_B)

Procedure For Pure ALOHA:

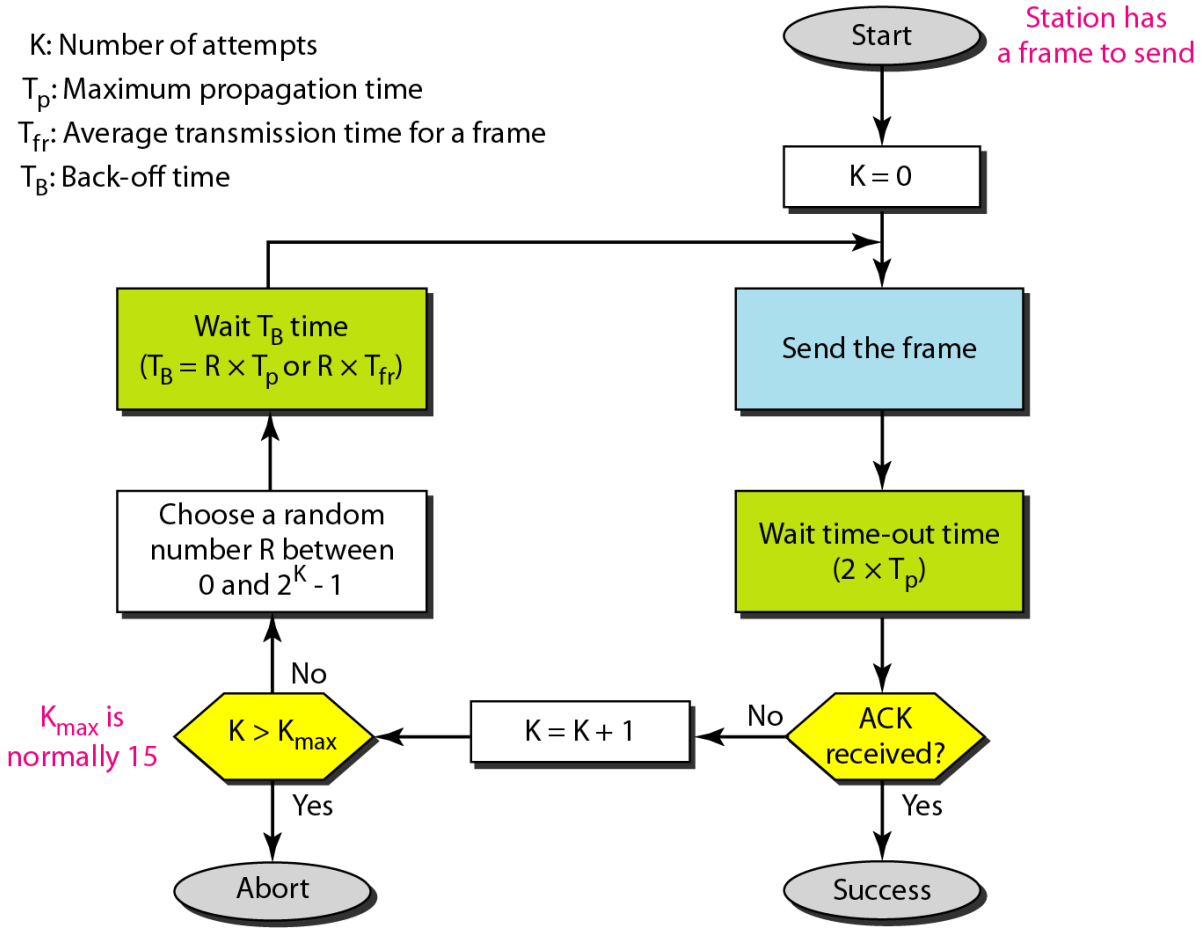
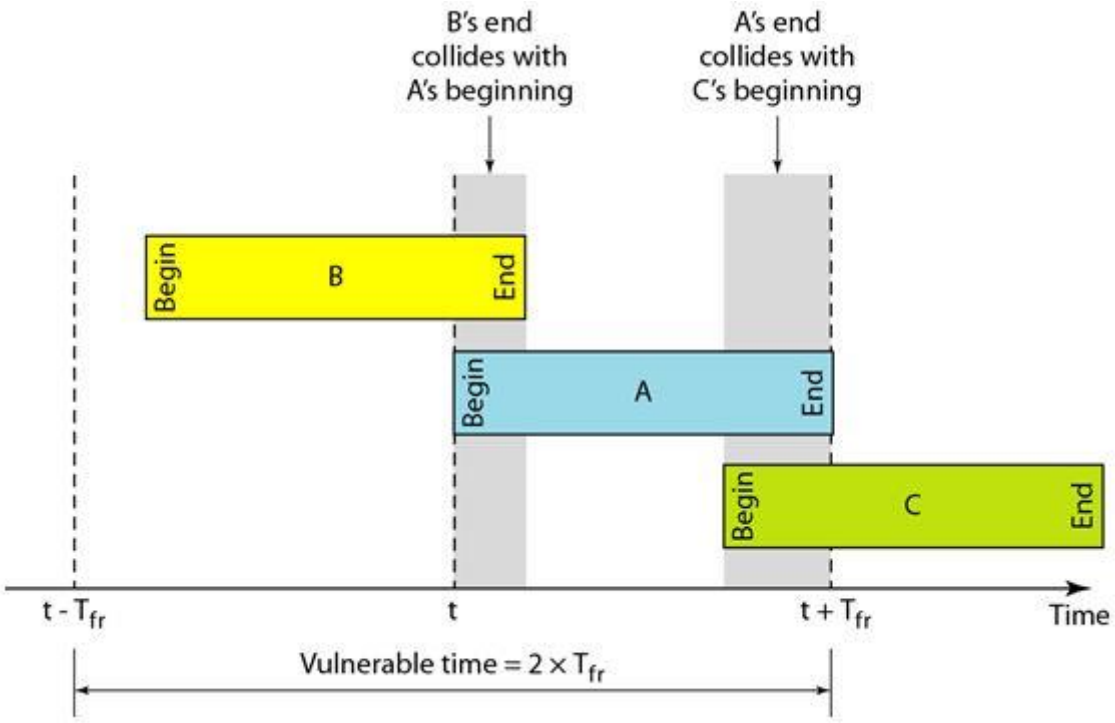


Figure: working for pure ALOHA

From an above flow chart, 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($2 \times Tp$).

Let all the packets have same length and each required one time unit for transmission (t_p).

Consider any user to send a packet A at time to if any other user B has generated a packet between time to and $t_0 + t_p$, the end of packet B will collide with the beginning of packet A.



Since in pure ALOHA packet, a station does not listen to the channel before transmitting it has no way of knowing that above frames was already under way.

Similarly if another user wants to transmit between (t_0+t_p) and (t_0+2t_p) i.e, packet C, the beginning of packet C will collide with the end of packet A.

Thus if two packets overlap by even the smallest amount in the vulnerable period both packets will be corrupted and need to be retransmitted.

Through Put: the through put is defined as average successful traffic transmitted between stations per unit time.

The unit of time is slot – time, which is the time required to transmit a frame

The average no. Successful transmission time for pure ALOHA is

$$S = G \times e^{-2G}$$

The maximum throughput S_{max} is 0.184, for G = ½

$$i.e., S = \frac{1}{2e} = 0.184$$

This is the best channel utilization that can be achieved is around 18% for pure ALOHA method.

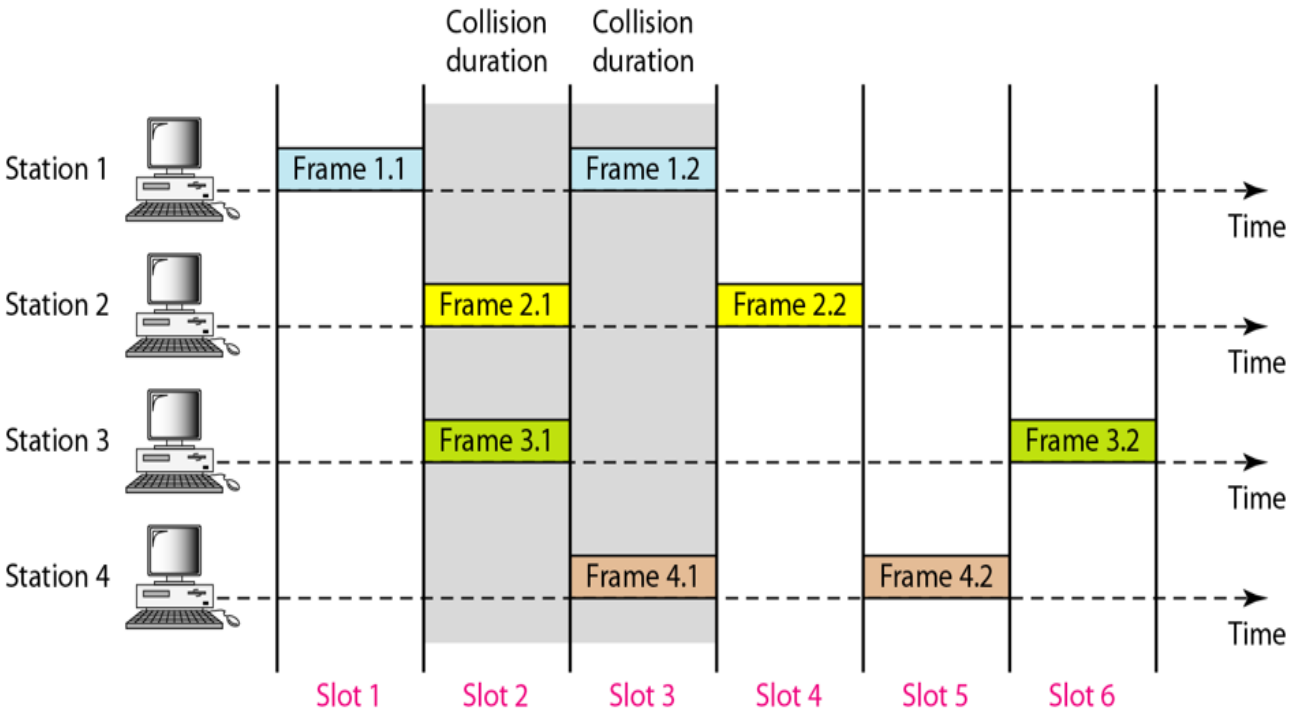
Advantages: it is a simple protocol which can result in low-cost stations since no synchronization is required between stations in the each system.

Slotted ALOHA: It was invented for improve the efficiency of pure ALOHA.

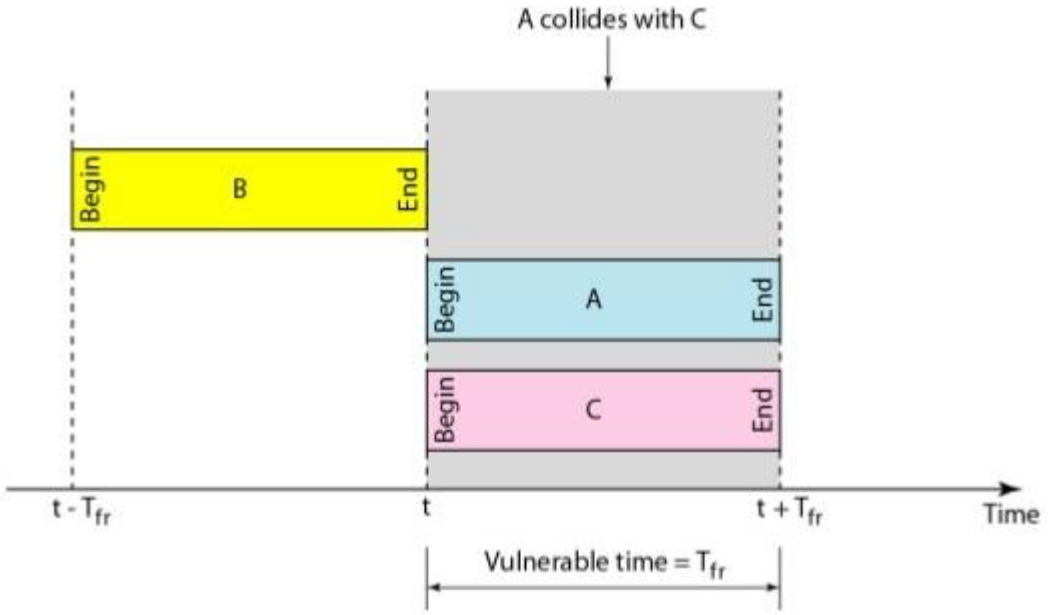
In slotted ALOHA the synchronized to these time slots and the stations are allowed to transmit at specific instance of time.

All users are then synchronized to these time slots, so that whenever a user generates a packet it must synchronize exactly with the next possible channel slot.

Consequently, the wasted time due to collision can be reduced to one packet time or vulnerable period is reduced to half.



Transmission attempts for four network user and random retransmission delays for colliding packets in slotted ALOHA.



Vulnerable time for Slotted ALOHA

Assumptions:

1. All frames are of same size.
2. Time is divided into equal size slots, a slot equals the time to transmit one frame.
3. Nodes start to transmit frames only at beginning of slots.
4. Nodes are synchronized
5. If two or more nodes transmit in a slot, all nodes detects collision before the slot ends.

Through of Slotted ALOHA:

In slotted ALOHA, the packets arrive in a synchronized fashion. The probability of single transmission during a slot time is

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

$$S = \frac{1}{e} = 0.368$$

The maximum through put S_{max} is = 0.368 $\therefore G = 1$

Pros & Cons:

1. Single active node can continuously transmit at full rate of channel.
2. Highly decentralized, each node independently divides when to retransmit.
3. Simple to implement

Cons:

1. Collisions waste slots
2. Idle slots.

Carriers Sense Multiple Accesses (CSMA): In order to minimize the chance of collision and therefore increase the performance, CSMA was developed.

The low maximum through of the ALOHA schemes is due to the wastage of transmission band width because of frame collisions.

This wastage can be reduced by avoiding transmissions that are certain to cause collisions. By sending the medium for the presence of a carrier signal from other stations, a station can determine whether there is an ongoing transmission.

CSMA requires that each station first listen to the medium before sending.

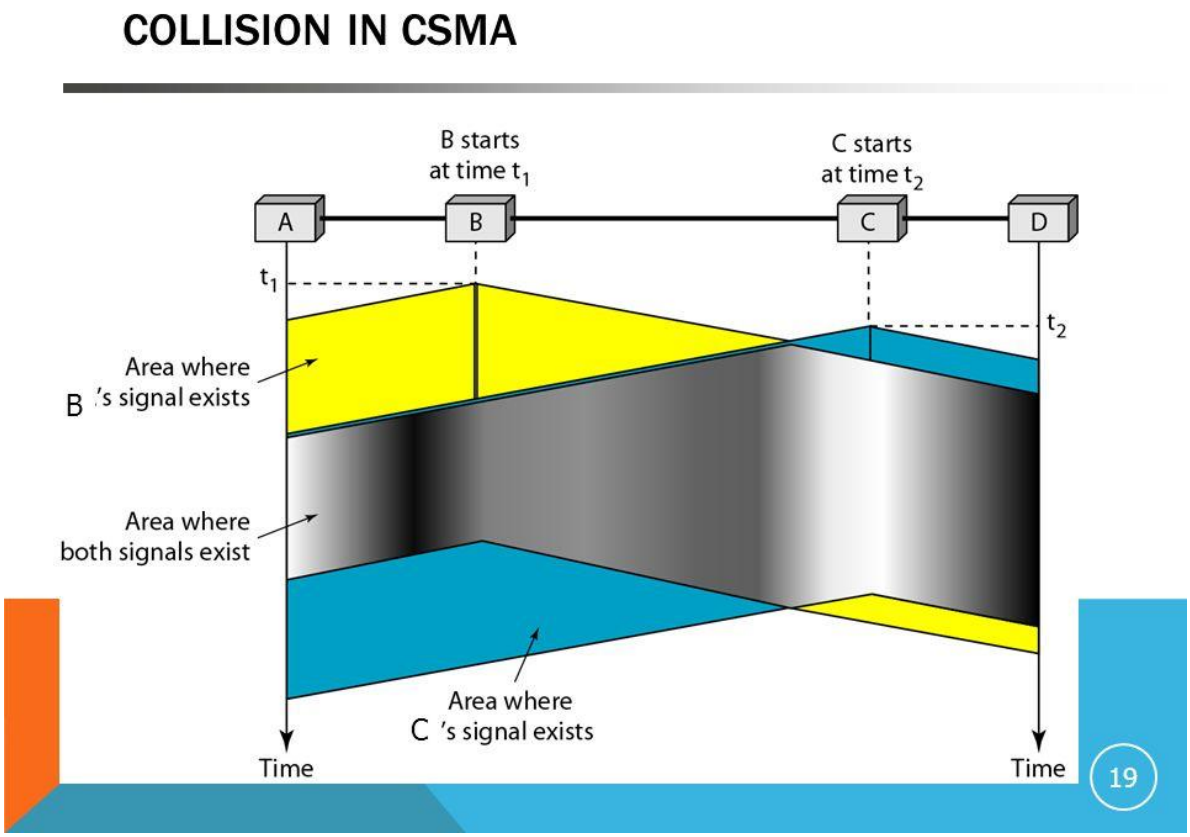
In other words, CSMA is based on the principle “sense before transmit” or “ listen before talk”

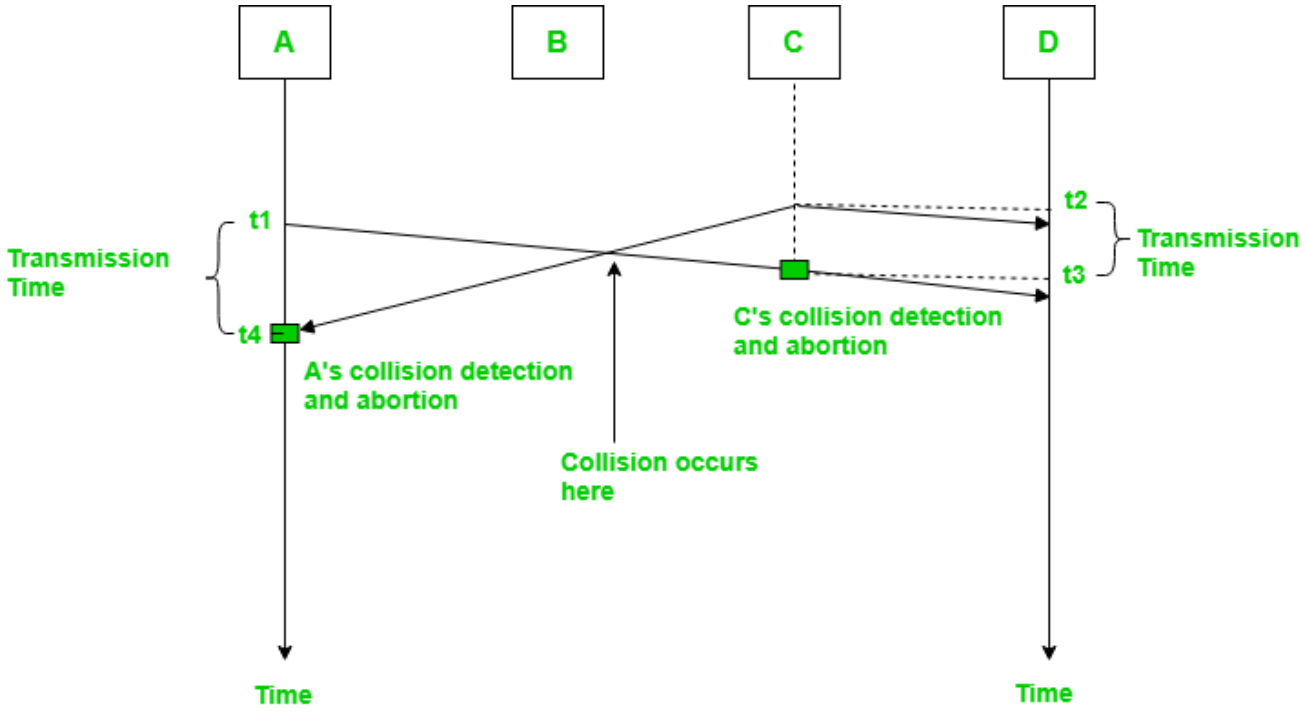
CSMA can reduce the possibility of collision but it cannot eliminate it.

The possibility of collision still exists because of propagation delay. A station may sense the medium and find it idle, only because the first bit sent by another station has not yet between received.

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 A sends a frame at time t_1 which reaches the right most station D at time $t_1 + T_p$





Persistence Methods: There are the three protocols

1. Non persistent CSMA
2. 1- persistent CSMA
3. P – persistent CSMA

1. Non – Persistent Method:

In non-persistent CSMA, when a station having a packet (frame) to transmit and finds that the channel is busy, it backs off a fixed interval of time

It then checks the channel again and if the channel is free then it transmits.

The back – off delay is determined by the transmission time of a frame, propagation time and other system parameters.

If the channel already in use, the stations does not continuously sense it for the purpose of seizing it immediately upon detecting the end of the previous transmission

But it waits a random period of time and again checks for activity.

2. 1- Persistent:

Any station wishing to transmits, monitor the channel continuously until the channel is ideal and then transmits immediately with probability one, hence the name called 1 – persistent.

When two or more stations are waiting to transmit a collision is guaranteed. Since each station will transmit immediately at the end of busy period. In this case each will wait a random amount of time and will then reattempt to transmit.

3. P – Persistent CSMA:

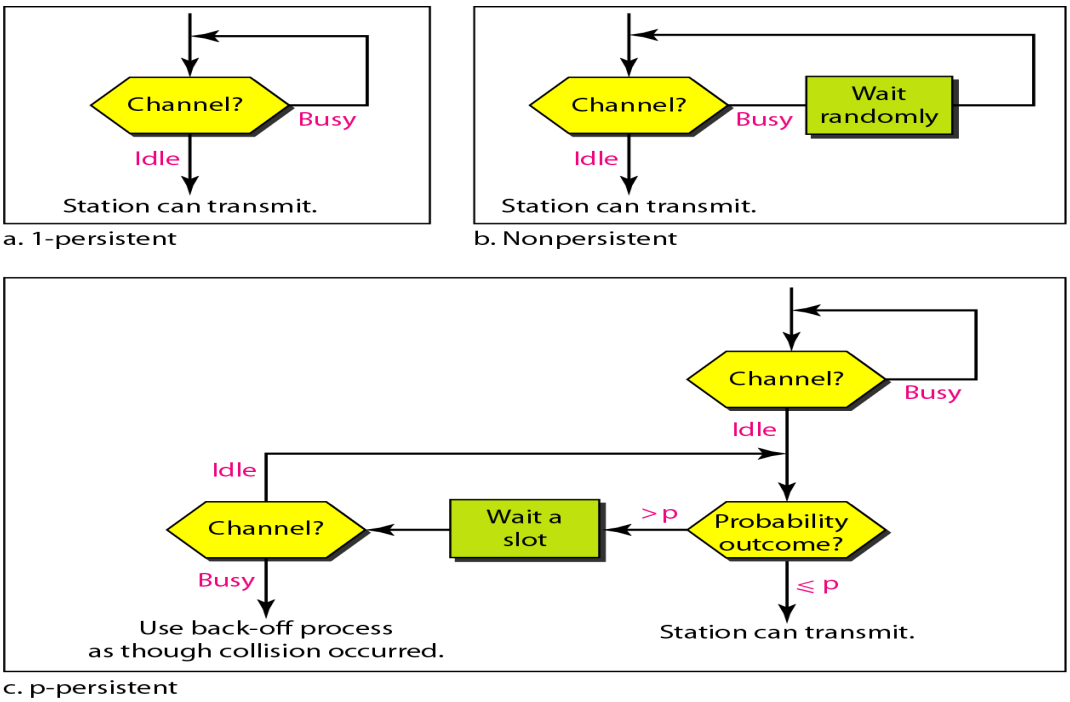
To reduce the probability of collisions in 1- persistent CSMA, not all the waiting stations are allowed to transmit immediately, after the channel is idle.

When a station becomes ready to send and it sense the channel to be idle, it either transmits with a probability p or it defers transmission by one time slot with a probability

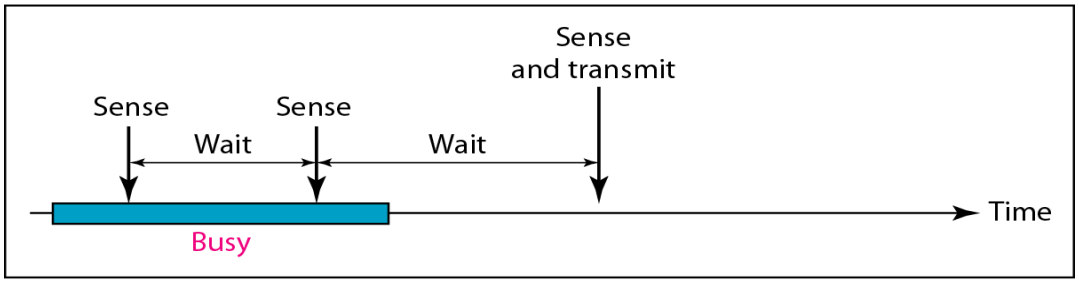
$$q = 1 - p$$

If deferred slot is idle, the station either transmits with probability p or defers again with a probability q.

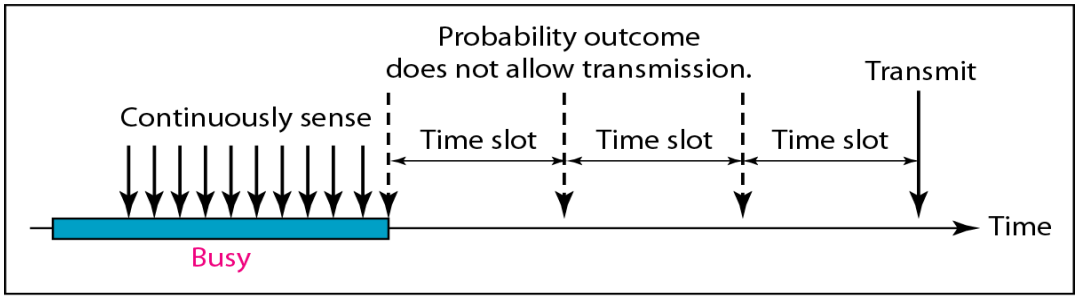
This process is repeated until either packets are transmitted or the channel becomes busy.



a. 1-persistent



b. Nonpersistent



c. p-persistent

CSMA/CD (Collision Detection):

In both CSMA and ALOHA schemes, collisions involve entire frame transmission. If a station can determine whether a collision is taking place, then the amount of wasted bandwidth can be reduce by aborting the transmission when a collision is detected.

CSMA/CD is the most commonly used protocol for LANS.

It was developed jointly by digital equipment corporation (DEC), Intel and Xerox

This network is called as Ethernet. The IEEE 802.3 CSMA /CD standard for LAN based on the Ethernet specification.

A basic protocol is that, a station with a message to send must monitor the channel to see if any other station is sending.

If another is sending the second station must wait or defer, until the sending station has finished. Then it may send its messages.

If no station was sending at the time that it first listened, the station may sends its message immediately.

The term “carried sense” indicates “listening before transmitting” behavior

If two or more stations have messages to send at the same time and they are separated by significant distance on the bus / channel, each may begin transmitting at roughly the same time without being a wave of the other station.

The signals from each station will super impose on the channel and is garbled beyond the decoding ability of the receiving station. This is called “collision”

A protocol is required for transmitting station to monitor the channel while sending each of bits message and to detect such “collisions”

When a collision has been detected, each of sending stations must cease transmitting wait for a random length of time, and then try again.

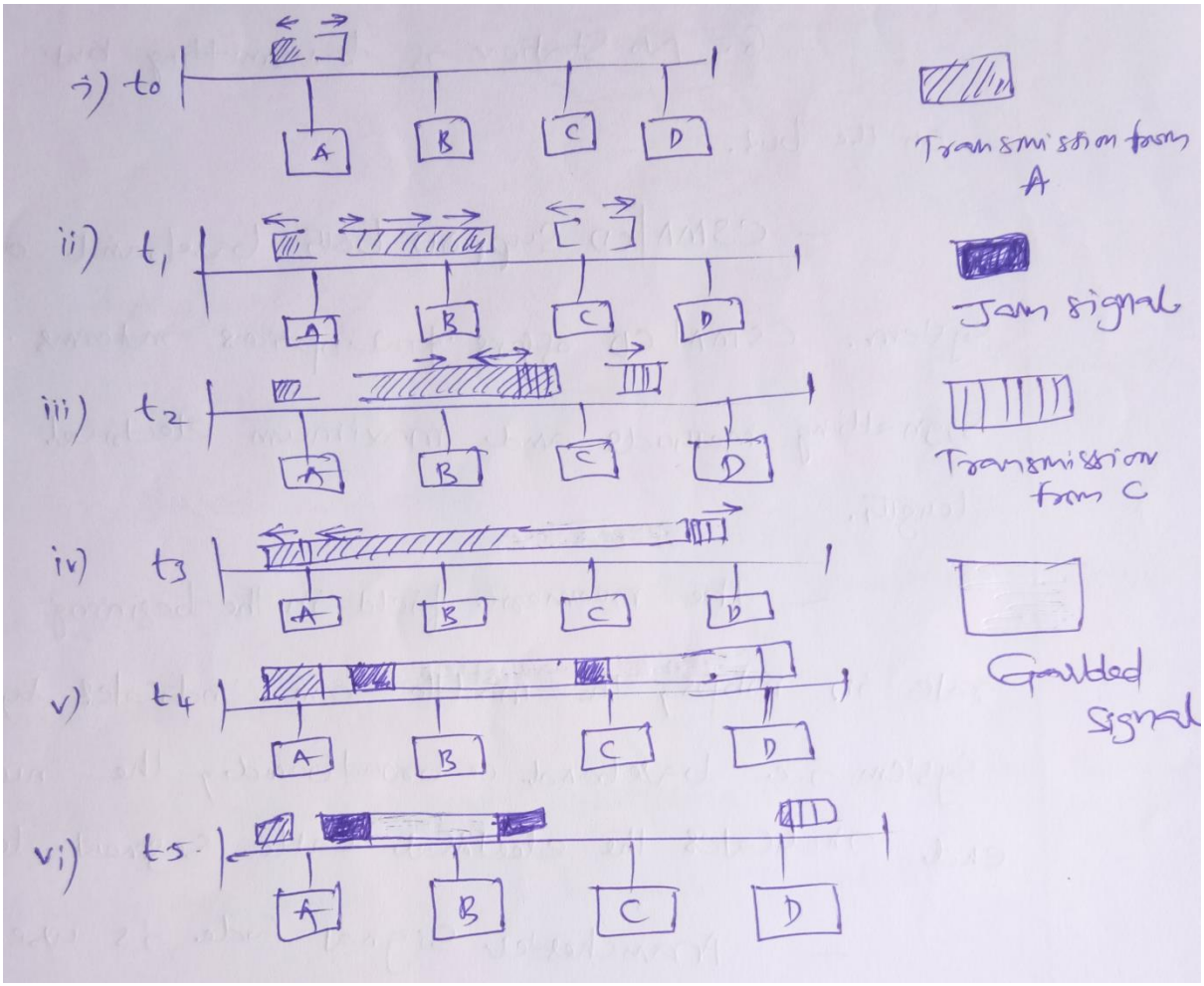
Because of quick termination of transmission time and bandwidth is saved. Therefore CSMA/CD is more efficient then ALOHA, slotted ALOHA and CSMA.

CSMA/ CD network best on a bus, multipoint topology with bursty asynchronous transmission. All stations are attached to one path and monitor the signal on the channel through trans receiver attached to the cable.

CSMA/CD has totally decentralized control and is based on contention access.

From an above figure, station A and station D are the extreme ends of a bus structure

1. Station A listens channel starts transmitting a packet addressing D.
2. Station B and C all ready for transmission. B sense a transmission on channel so defers. C is unaware of transmission and begins its own transmission.
3. Station A transmission reaches C. C detects collision and crass transmission sends jam signal.
4. Effect of collision propagates back to A, A stops its transmission.
5. A sends jam signal.
6. No station is transmitting but there still signals on the bus.



CSMA/CD supports both base band and broad band system CSMA/CD offers four options in terms of bit rate signaling methods and maximum electrical cable segment length.

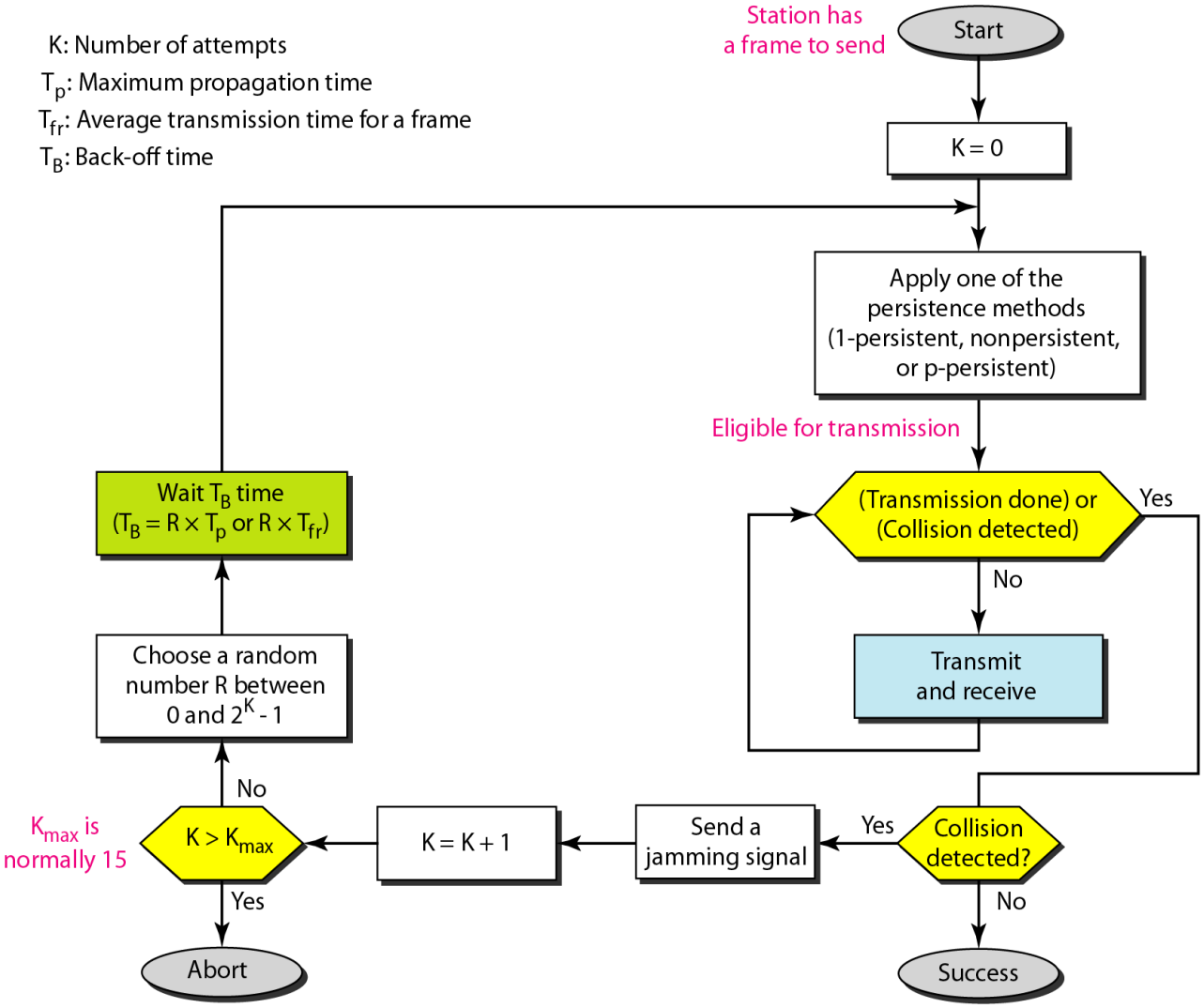
The numeric field in the beginning indicates the bit rate in Mbps, the middle terms indicates type of signaling system i.e. back band or broad band, the numeric field in the end indicates the electrical cable segment length in × 100 meters

Manchester signal code is used at the baseband level of transmission.

CSMA/CD Throughput:

1. The throughput of CSMA/CD is greater than that of pure or slotted ALOHA
2. For 1- persistent method, the maximum throughput is around 50% when G = 1
3. For non-persistent method, the maximum throughput can go up to 90% when G is between 3 and 8.

CSMA/CD Flow Graph:



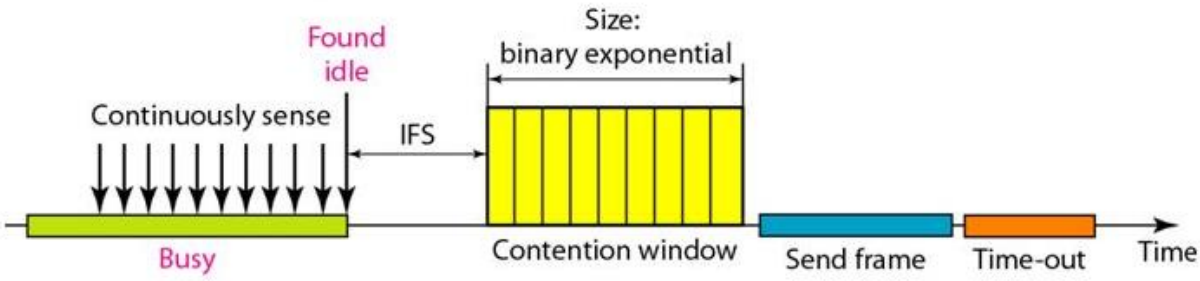
CSMA/CA: (collision Avoidance): In wireless n/w's cannot use CSMA/Cd in the MAC sub layer, this requires the ability to receive and transmit at the same time.

In a wireless n/w's, much of the sent energy is lost transmission. The received signal has very little energy. Therefore, a collision may add only 5 to 10 percent additional energy.

This is not useful for effective collision detection; we need to avoid collision on wireless networks because they cannot be detected. Since, CSMA/CA was invented for the networks.

Collision avoidance using in three models

1. Inter frame space
2. Contention window
3. Acknowledgement



Inter Frames Space:

1. Collisions are avoided by deterring transmission even if the channel is found idle.
2. When an idle channel is found, the station does not send immediately. It waits for a period of time called the inter frame space. (IFS)
3. In CSMA/CA, the IFS can also be used to define the priority of a station of a frame. A station that is assigned shorter it's has a higher priority.

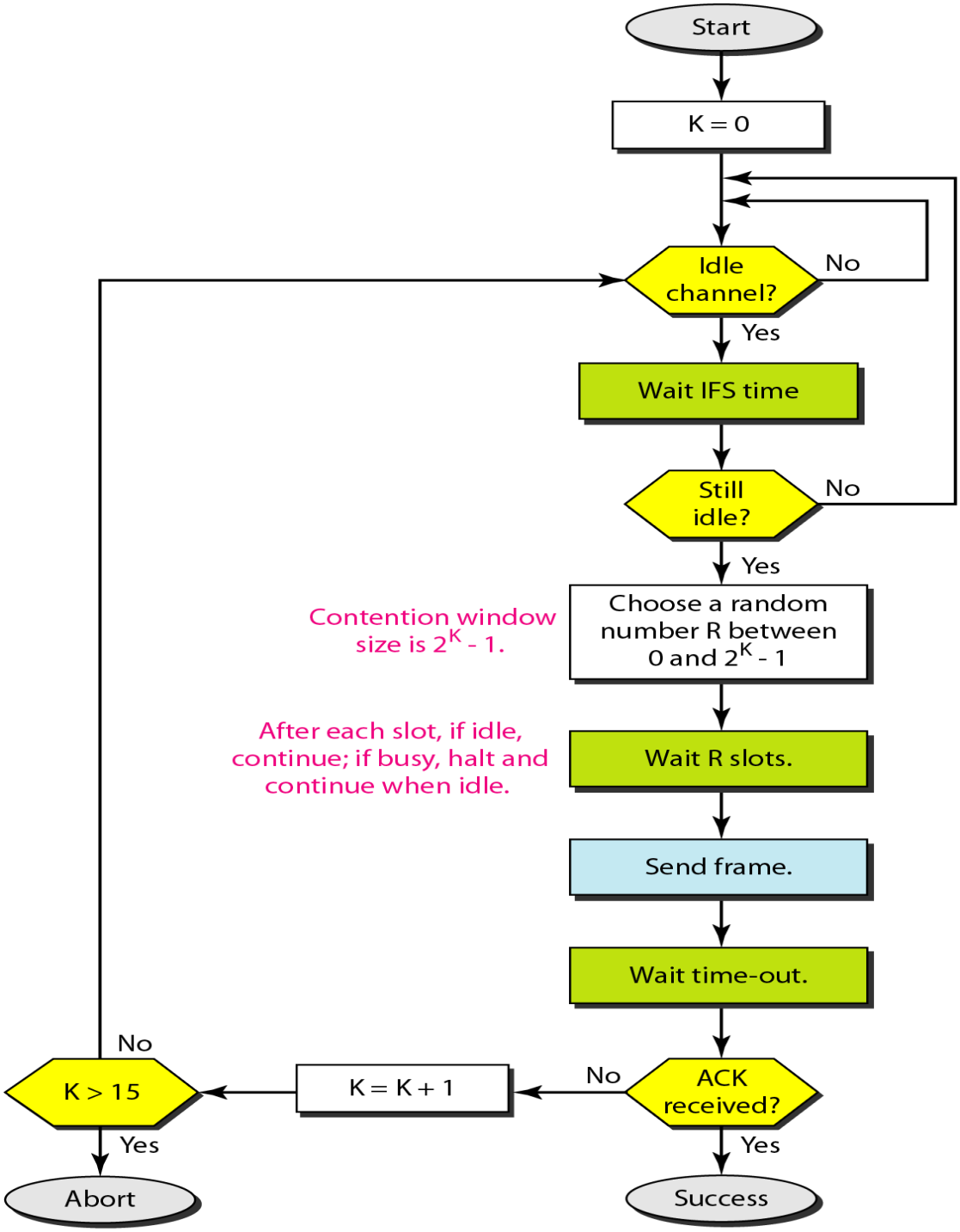
Contention Window:

1. Contention windows are an amount of time divided into slots. A station that is ready to send chooses a random no. of slots as its wait time.
2. Station set one slot for the first time and then double each time station cannot detect an idle channel after the IFS time.
3. In this method, the station needs to sense the channel after each time slot
4. If the station finds channel busy, it does not restart the process, it just stops the timer and restarts it when the channel is sensed as idle.
5. This method gives the priority to the station with the longest waiting time.

Acknowledgement:

The data may be corrupted during the transmission. The positive acknowledgement and the time out can help guarantee that the receiver has received the frame.

Flow Chart for CSMA/CA:



Controlled Access or Collision Free Protocols:

Collision free protocols in order to medium so that every station has chance to transfer.

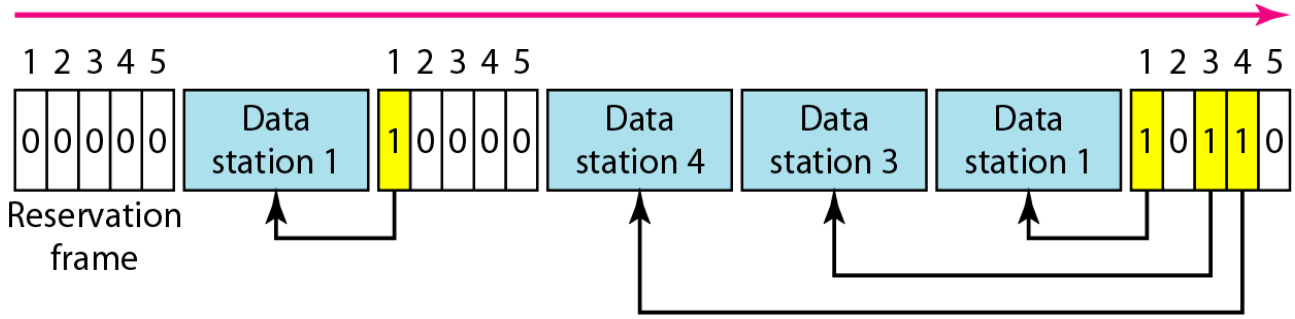
Eliminates the collision completely

The collision free protocols are used for controlled access methods like.

1. Reservation
2. Polling
3. Token passing

1. Reservation:

Before sending data, station needs to make reservation time is divided into intervals. In each interval, a reservation frame precedes the data frames sent in that interval.



The no. of reservations are equal to the no. of stations. Each station has their own minislot in the reservation frames. When station needs to send a data frame it makes a reservation in its own minislot. The stations that have made reservations can send their data. In the first slot, only station 1, 3 and 4 have made reservation.

Polling:

Polling work with topologies, one device is designed as primary stations and other devices are secondary stations.

All the data exchange take place through the primary device. The primary device controls the links. The secondary devices follow its instructions.

The primary device decides, which device is allowed to use the channel at a given time.

The primary device wants to receiver data; it asks the secondary if they have anything to send, this is called polling

In polling there are two modes

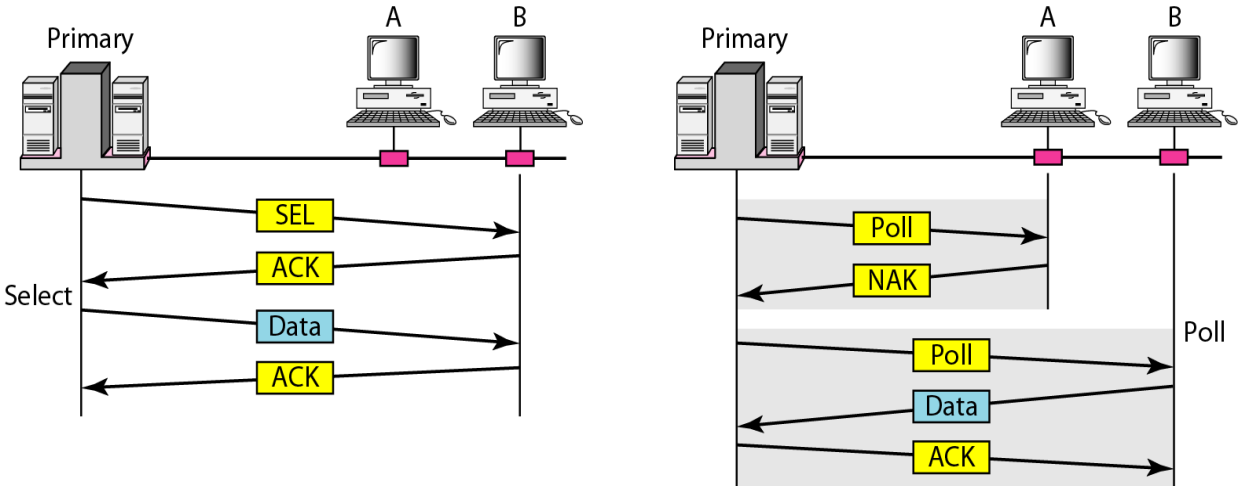
1. Select mode
2. Poll mode

Select Mode:

In polling primary device receives the data, the primary device sends data to secondary remembers that the primary controls the link.

Before sending data, the primary created and transmit a select (SEL) frame

SEL frame includes address of the intended secondary device.



Poll mode: the poll function is used by the primary devices to solicit transmission from the secondary devices.

When the primary is ready to receive data, 4 must be 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 has nothing to send or with data if it does.

If the response is negative (a NAK frame) then the primary polls the next secondary in the same manner until it finds one with data to send.

If the response is positive, the primary reads the frame and returns an acknowledgement (ACK), verifying its receipt.

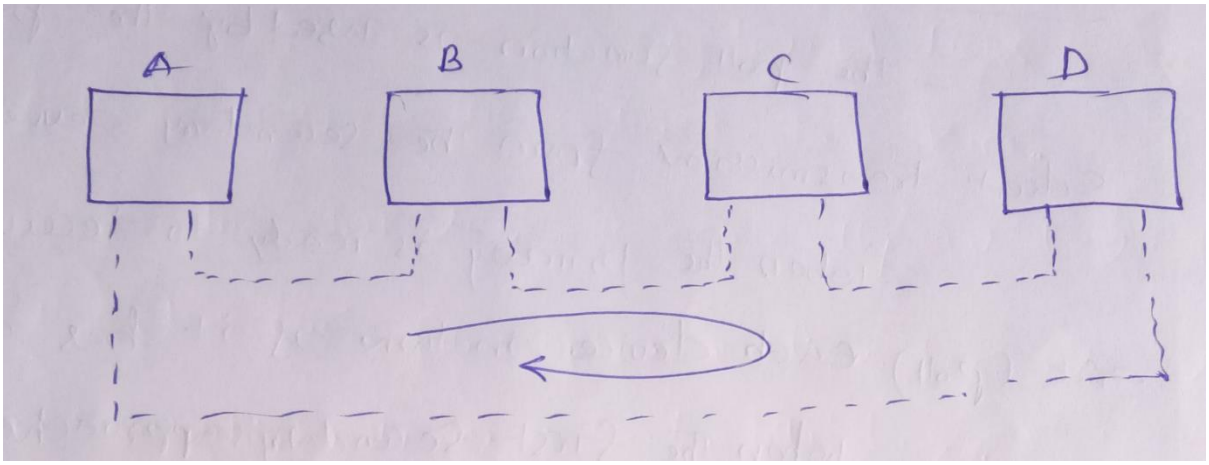
Token Passing:

A station is allowed to send data when it receiver a token (special frame).

Ring topology is used for connecting devices. Each has a predecessor and a successor.

Frames are coming from predecessor and going to the successor.

Token circulates around the ring the station capture the token if they want to send data.

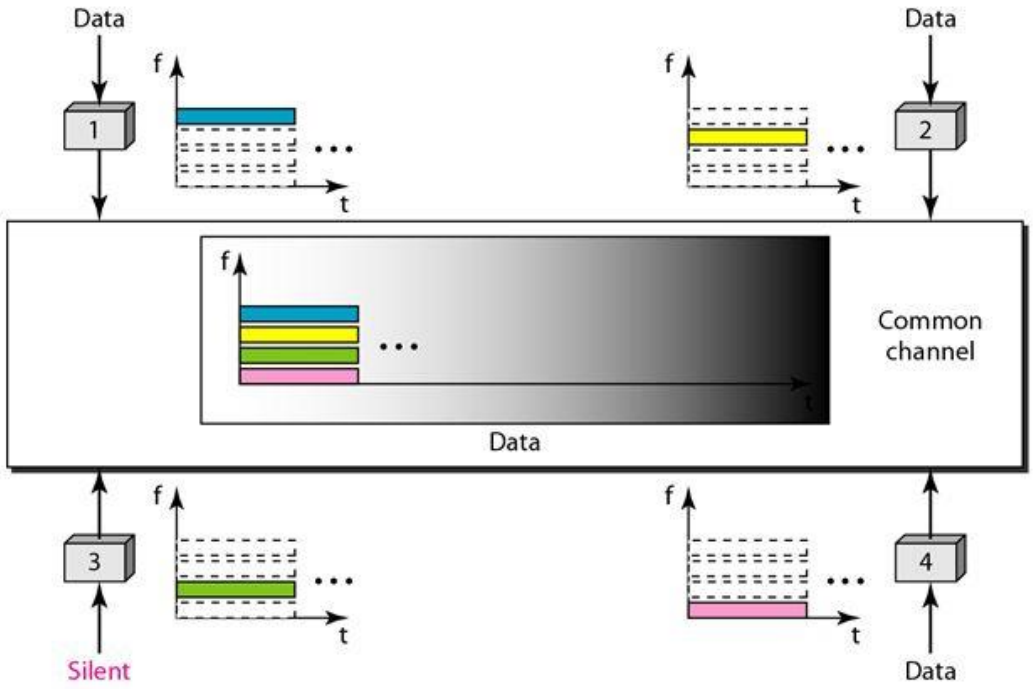


Channelization:

Channelization is a multiple-access method in which the available bandwidth of a link is shared in time, frequency, or through code, between different stations. The three channelization protocols are FDMA, TDMA, and CDMA.

The Frequency-Division Multiple Access (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. In other words, 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. The following figure shows the idea of FDMA.



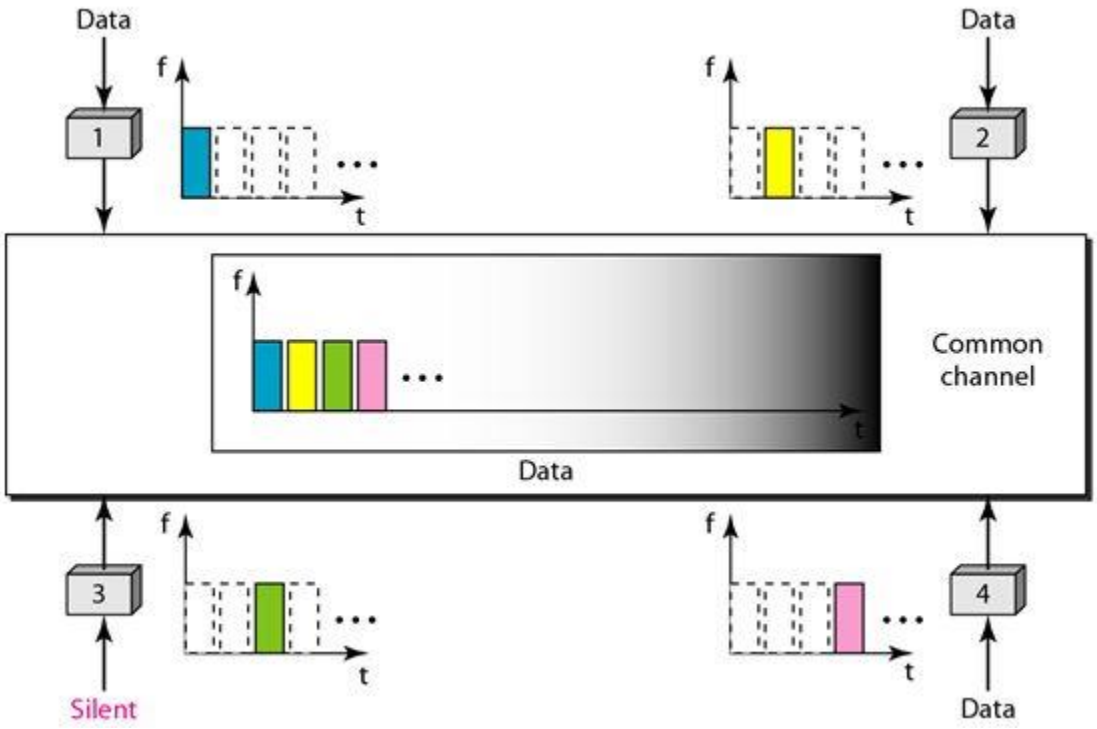
The differences between FDM and FDMA are as follows:

FDM, is a physical layer technique that combines the loads from low-bandwidth channels and transmits them by using a high-bandwidth channel. The channels that are combined are low-pass. The multiplexer modulates the signals, combines them, and creates a bandpass signal. The bandwidth of each channel is shifted by the multiplexer.

FDMA, on the other hand, is an access method in the data link layer. The data link layer in each station tells its physical layer to make a bandpass signal from the data passed to it. The signal must be created in the allocated band. There is no physical multiplexer at the physical layer. The signals created at each station are automatically bandpass-filtered. They are mixed when they are sent to the common channel.

Time-Division Multiple Access (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 is assigned time slot. The following 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, we can insert guard times.

Synchronization is normally accomplished by having some synchronization bits at the beginning of each slot.

The differences between TDMA and TDM are:

• TDM is a physical layer technique that combines the data from slower channels and transmits them by using a faster channel. The process uses a physical multiplexer that interleaves data units from each channel.

• TDMA, on the other hand, is an access method in the data link layer. The data link layer in each station tells its physical layer to use the allocated time slot. There is no physical multiplexer at the physical layer.

Code-Division Multiple Access (CDMA):

CDMA simply means communication with different codes. CDMA differs from FDMA because only one channel occupies the entire bandwidth of the link. It differs from TDMA because all stations can send data simultaneously; there is no timesharing.

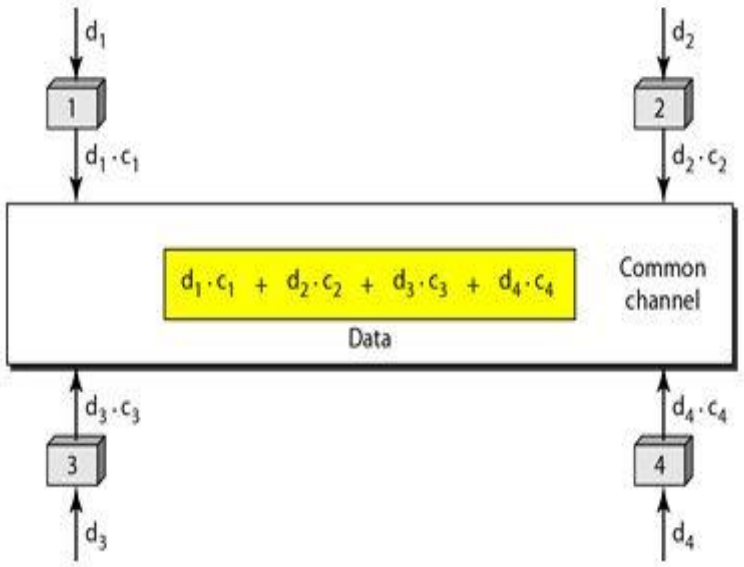
Implementation:

Let us 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).

With these two properties in mind, how the above four stations can send data using the same common channel, as shown in the following figure.



Station 1 multiplies (a special kind of multiplication, as we will see) its data by its code to get d1.c1. Station 2 multiplies its data by its code to get d2.c2. And so on. The data that go on the channel are the sum of all these terms, as shown in the box.

Any station that wants to receive data from one of the other three multiplies the data on the channel by the code of the sender. For example, suppose stations 1 and 2 are talking to each other. Station 2 wants to hear what station 1 is saying. It multiplies the data on the channel by c1 the code of station1.

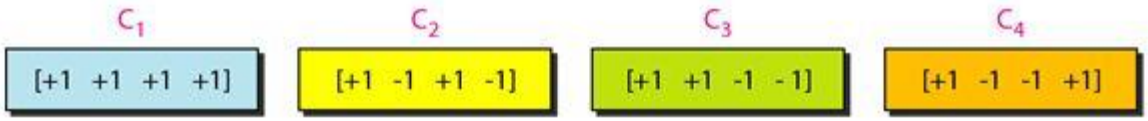
Because (c1.c1) is 4, but (c2 . c1), (c3. c1), and (c4 .c1) are all 0s, station 2 divides the result by 4 to get the data from station1.

data =(d1.c1+d2.c2+d3.c3+d4.c4).c1

= c1. d1. c1+ c1. d2. c2+ c1. d3. c3+ c1. d4. c4= 4d1

Chips:

CDMA is based on coding theory. Each station is assigned a code, which is a sequence of numbers called chips, as shown in the following figure. The codes are for the previous example.



We need to know that we did not choose the sequences randomly; they were carefully selected. They are called orthogonal sequences and have the following properties:

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. [+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 the each sequence. This is called the inner product of two equal sequences. For example,

[+1 +1-1 -1]. [+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 inner product of two different sequences. For example,

[+1 +1 -1-1] • [+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]

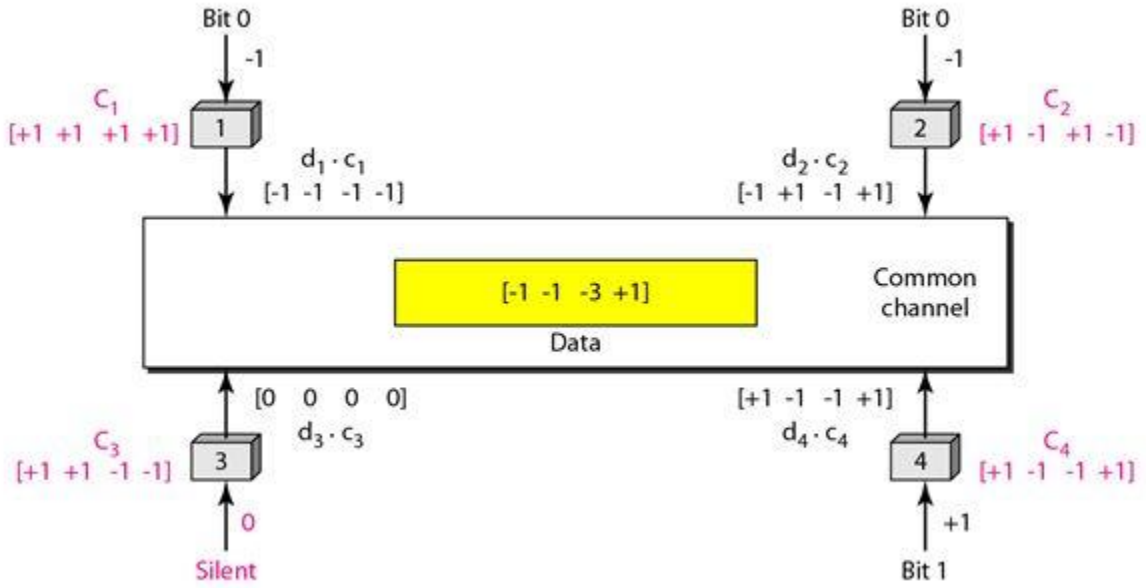
Data Representation:

We follow the following 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.

Encoding and Decoding:

As a simple example, we show how four stations share the link during a 1-bit interval. The procedure can easily be repeated for additional intervals. We 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. For simplicity, we assume that all stations send the resulting sequences at the same time. The sequence on the channel is the sum of all four sequences as defined before. The following figure shows the situation.



Now imagine 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 1..

IEEE 802.3 MAC Frame:

MAC sub layer: it governs the operation of the access method. Is also frames data receiver from the upper layer and passes them to physical layer.

Frames format:

The Ethernet frame contains several fields; preamble, SFD, DA, SA, length or type of protocol data unit (PDU) upper layer data and CRC.

Ethernet does not provide any mechanism for acknowledge received frame, making it what is known as an unreliable medium.

Preamble	SFD	Desfination address	source addness	length or type	Data & padding	CRC
7 bytes	1 byte	6 Bytes	6 bytes	2bytes		4 bytes
physical layer header						

Preamble: A 7 – byte pattern of alternating 0's and 1's used by the receiver to establish bit synchronization.

Each frame contains the bit pattern 1 0 1 0 1 01 0 the pattern provides only an alert and a timing pulse.

The preamble is actually added at the physical layer and is not part of the frame.

Start frame delimiter (SFD):

Sequence 10101011 which indicates the actual start of the frame and enables the receiver the actual start of the frame and enables the receiver to locate the first bit of the rest of the frame.

Destination address (DA): The destination address field is 6 bytes and specifies the station for which is the frame is intended.

It may be a unique physical address, a group address or global address.

Source address (SA):

The source address field is also 6 bytes and contains the physical address of the sender of the packet

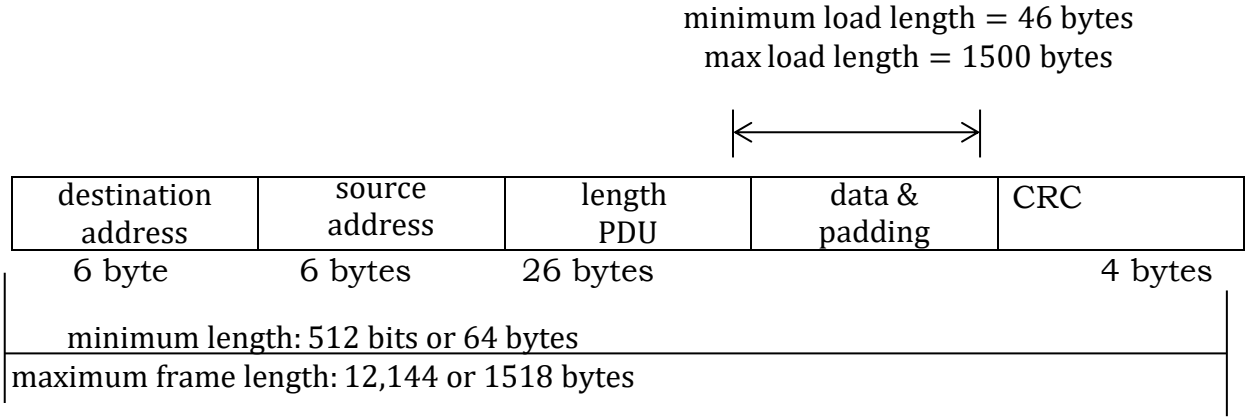
Length of type: length of LLC data field and Ethernet type field depending upon whether the frame confirms to the IEEE 802.3 standard (or) earlier Ethernet specification.

Data: this field carries data encapsulated from the upper – layer protocols. It is a minimum of 46 and maximum of 1500 bytes

CRC: this field contains error detection information

Frame length:

An Ethernet frame needs to have a minimum length of 512 bits or 64 bytes.



An Ethernet frames need to have a minimum length of 512 bits or 64 bytes. Part of this length is the header and the trailer.

If we count 18 bytes of header and trailer i.e. 6 bytes of source address + 6 bytes of destination address + 2 bytes of length + 4 bytes of CRC then the minimum length of the data from the upper layer is 64 – 18 = 46 bytes.

If upper layer packet is less than 46 bytes padding is added to make up the difference.

The standard defines the maximum length of a frame (without preamble and SFD) is 1518 bytes. If we subtract the 18 bytes of header and trailer, the maximum length of the payload is 1500 bytes.