UNIT -3
DATA LINK LAYER

3.1. FUNCTION OF DATA LINK LAYER

Data Link Layer is the second layer of OSI Layered Model. This layer is one of the most complicated layers and has complex functionalities and liabilities. Data link layer hides the details of underlying hardware and represents itself to the upper layer as the medium to communicate.

Data link layer is responsible for converting data stream to frame and to send that over the underlying hardware. At the receiving end, Data link layer picks up data from hardware which are in the form of electrical signals, assembles them in a recognizable frame format, and hands over to the upper layer.

Data link layer has two sub-layers:

- **Logical Link Control:** It deals with protocols, flow-control, and error control
- **Media Access Control:** It deals with actual control of media

FUNCTIONALITY OF DATA-LINK LAYER

Data link layer does many tasks on behalf of the upper layer. These are:

- **Framing** Data-link layer takes packets from Network Layer and encapsulates them into Frames. Then, it sends each frame bit-by-bit on the hardware. At receiver' end, the data link layer picks up signals from hardware and assembles them into frames.

- **Addressing** Data-link layer provides layer-2 hardware addressing mechanism. Hardware address is assumed to be unique on the link. It is encoded into hardware at the time of manufacturing.

- **Synchronization** When data frames are sent on the link, both machines must be synchronized in order to transfer to take place.

- **Error Control** Sometimes signals may have encountered problem in transition and the bits are flipped. These errors are detected and attempted to recover actual data bits. It also provides an error reporting mechanism to the sender.

- **Flow Control** Stations on the same link may have different speed or capacity. Data-link layer ensures flow control that enables both machines to exchange data at the same speed.

- **Multi-Access** When the host on the shared link tries to transfer the data, it has a high probability of collision. Data-link layer provides mechanisms such as CSMA/CD to equip capability of accessing a shared media among multiple Systems.

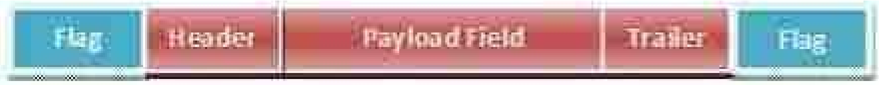
3.2. DATA LINK CONTROL:FRAMING, FLOW AND ERROR CONTROL

3.2.1 Framing

Framing is a function of the data link layer. It provides a way for a sender to transmit a set of bits that are meaningful to the receiver. Ethernet, token ring, frame relay, and other data link layer technologies have their own frame structures. Frames have headers that contain information such as error-checking codes.

The data link layer needs to pack bits into frames, so that each frame is distinguishable from another. The Data Link layer prepares a packet for transport across the local media by encapsulating it with a header and a trailer to create a frame.

Parts of a Frame



- **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 – There are two types of framing:

1. **Fixed size –** The frame is of fixed size and there is no need to provide boundaries to the frame, length of the frame itself acts as delimiter.
2. **Variable size –** In this there is a need to define the end of the frame as well as beginning of the next frame to distinguish.

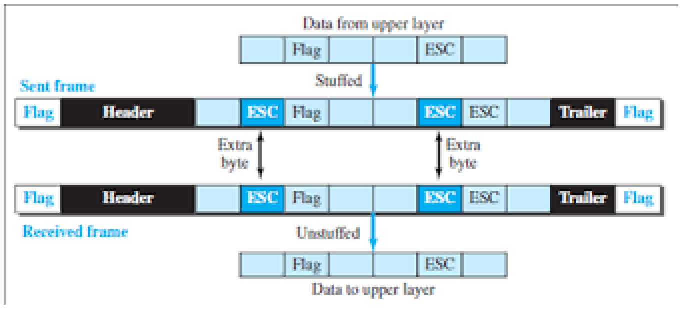
This can be done in two ways:

Length field – We can introduce a length field in the frame to indicate the length of the frame. Used in Ethernet(802.3). The problem with this is that sometimes the length field might get corrupted.

End Delimiter (ED) – We can introduce an ED(pattern) to indicate the end of the frame. Used in Token Ring. The problem with this is that ED can occur in the data. **This can be solved by:**

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

   The process of adding 1 extra byte whenever there is a flag or escape character in the text.



2. **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.**

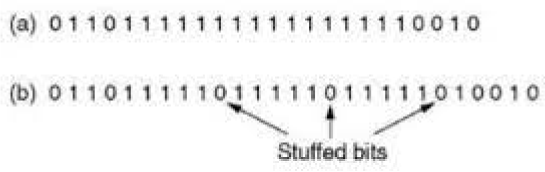

(a) 011011111111111111110010
(b) 011011111011111011111010010
Stuffed bits
(c) 011011111111111111110010

Fig1: Bit stuffing

The process of adding one extra 0 whenever five consecutive 1s follow a 0 in the data.

3.2.2. Flow and Error Control

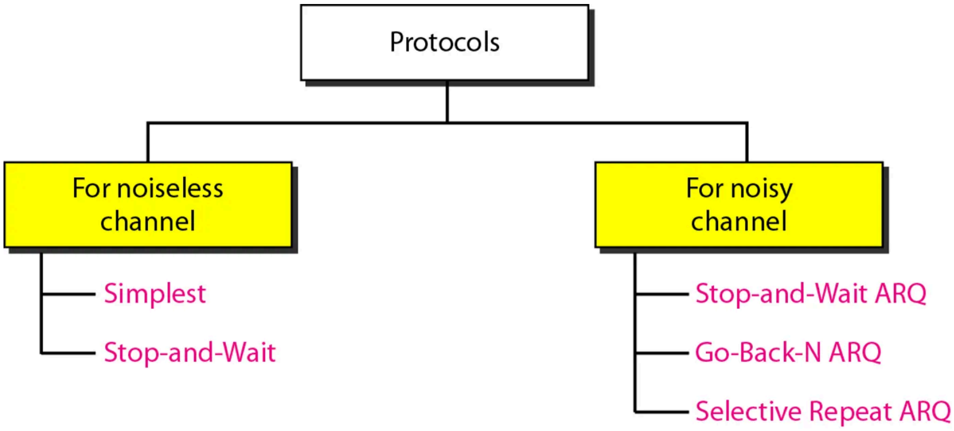
Data link control = flow control + error control

Flow control refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgement

Error control in the data link layer is based on automatic repeat request (ARQ), which is the retransmission of data

ACK, NAK(Negative ACK), Piggybacking (ACKs and NAKs in data frames)

***NAK** is used to indicate that a packet has been corrupted and to resend it, but there is no need to change the transmission rate.*



<u>Flow Control</u>

An idealistic channel in which no frames are lost, corrupted or duplicated. The protocol does not implement error control in this category. There are two protocols for the noiseless channel as follows.

Simplest:

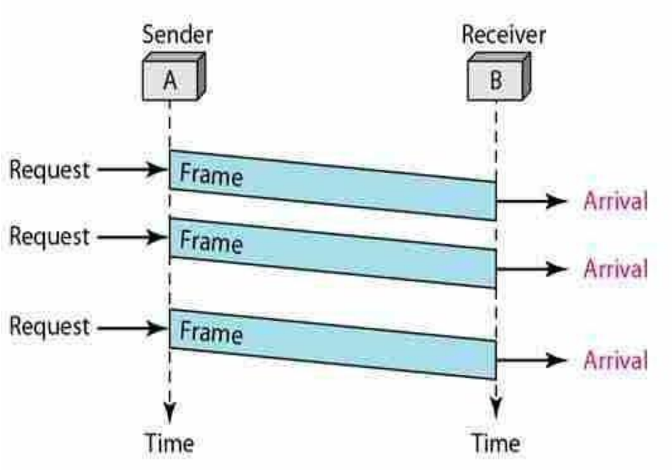
Simplest Protocol is one that has no flow or error control and it is a unidirectional protocol in which data frames are traveling in only one direction-from the sender to receiver.

We assume that the receiver can immediately handle any frame it receives with a processing time that is small enough to be negligible.

The data link layer of the receiver immediately removes the header from the frame and hands the data packet to its network layer, which can also accept the packet immediately.

The following figure shows an example of communication using this protocol.

It is very simple. The sender sends a sequence of frames without even thinking about the receiver.



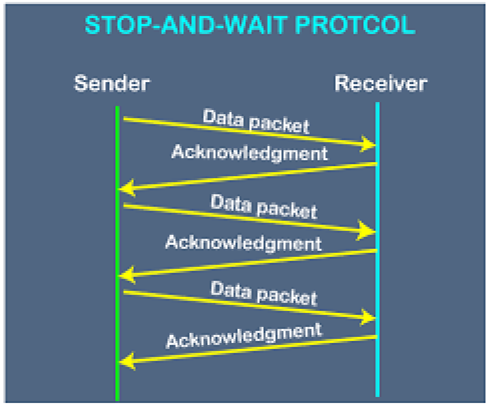
Stop & wait Protocol:

The sender sends a frame and waits for a response from the receiver. When ACK(acknowledged) will arrive from the receiver side then send the next frame and so on.

1. The sender node sends a data packet to the receiver node.
2. Then, waits for the feedback of the transmitted packet.
3. As soon as the receiver node receives a data packet it starts processing it.
4. Then, the receiver node sends the feedback to the sender node (about the received data packet).
5. After receiving the feedback, if the feedback is positive then the sender node sends the next data packet otherwise resends the damaged packet.

Stop & Wait Protocol performs better for LANs than WANs because the efficiency of the protocol is inversely proportional to the distance between the sender & receiver node and as we all know that the distance is less in LANs as compared to WANs.

But this protocol has some issues, one such issue is the probability of occurrence of deadlock, which is very high due to loss in a data packet or loss in feedback which can lead to infinite waiting, where the sender will keep on sending the data packet if the feedback is lost.



Note:

(ARQ) Automatic repeat request, also known as automatic repeat query, is an error-control method for data transmission that uses acknowledgements and timeouts to achieve reliable data transmission over an unreliable communication channel.

FOR NOISY CHANNEL

Stop & Wait ARQ

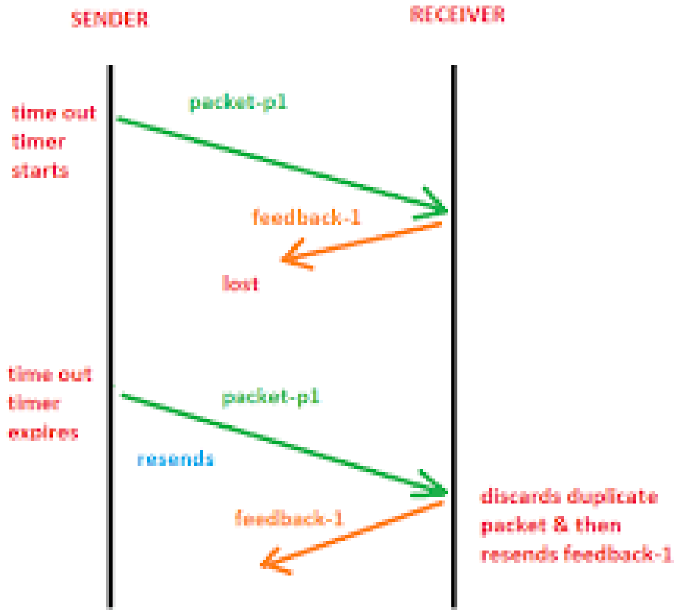
Stop & Wait ARQ is a sliding window protocol for flow control and it overcomes the limitations of Stop & Wait,

We can say that it is the improved or modified version of Stop & Wait protocol.

A sliding window protocol is a feature of packet-based data transmission protocols. Sliding window protocols are used where reliable in-order delivery of packets is required, such as in the data link layer as well as in the Transmission Control Protocol.

Stop & Wait ARQ assumes that the communication channel is noisy (previously Stop & Wait assumed that the communication channel is not noisy). Stop & Wait ARQ also assumes that errors may occur in the data while transmission.

Working of Stop & Wait ARQ

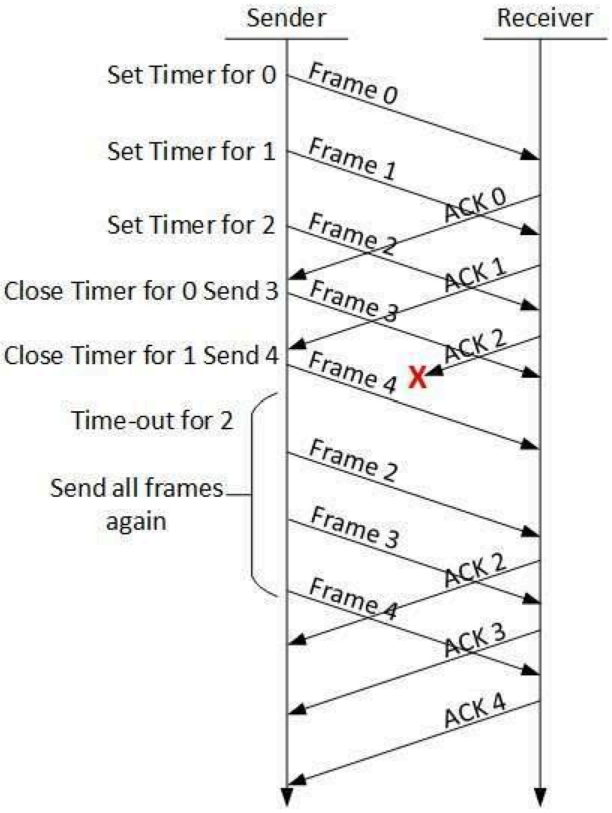


The following transition may occur in Stop-and-Wait ARQ:

- The sender maintains a timeout counter.
- When a frame is sent, the sender starts the timeout counter.
- If acknowledgement of a frame comes in time, the sender transmits the next frame in the 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.



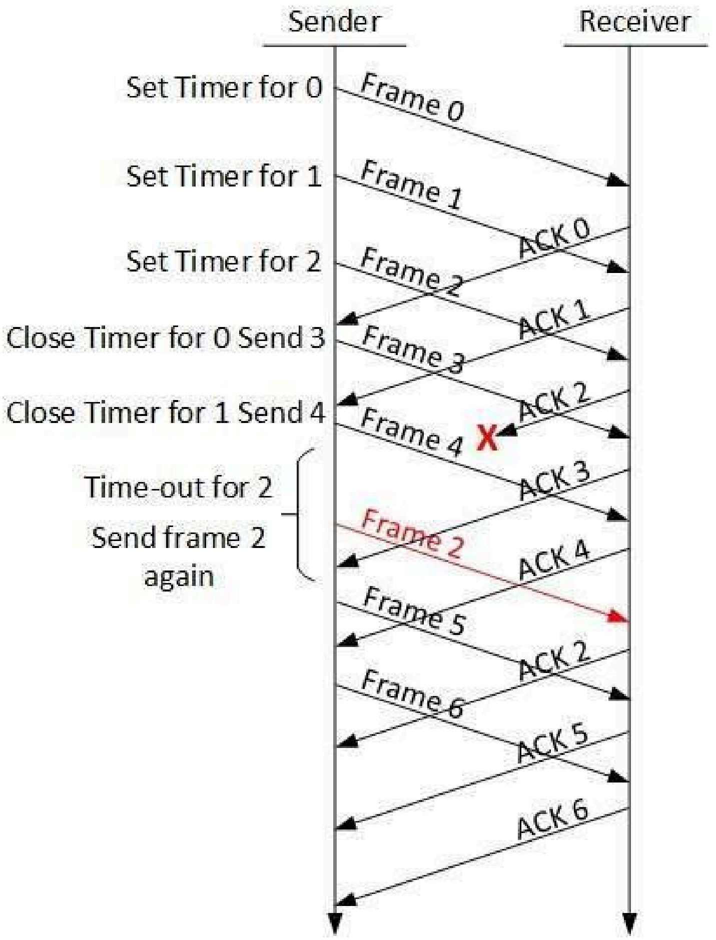
The sending-window size enables the sender to send multiple frames without receiving the acknowledgement of the previous ones.

The receiving-window enables the receiver to receive multiple frames and acknowledge them. The receiver keeps track of the incoming frame's sequence number.
When the sender sends all the frames in the window, it checks up to what sequence number it has received positive acknowledgement.

If all frames are positively acknowledged, the sender sends the next set of frames. If the sender finds that it has received NACK or has not received 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.



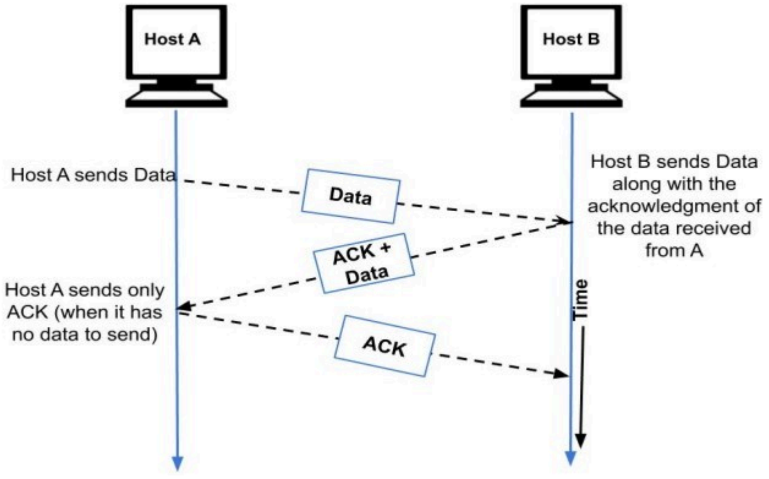
In Selective-Repeat ARQ, the receiver, while keeping track of sequence numbers, buffers the frames in memory and sends NACK for only frames which are missing or damaged.
The sender in this case, sends only the packet for which NACK is received.

Piggybacking Protocol

To improve the efficiency of the bidirectional protocols.

Piggybacking in Go-Back-N ARQ

Piggybacking is a method of **attaching acknowledgement to the outgoing data packet**.



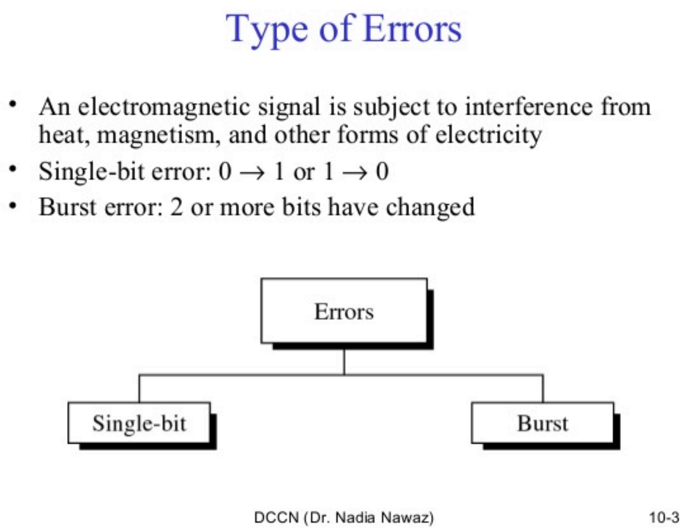
Host A
Host B
Host A sends Data
Data
Host B sends Data along with the acknowledgment of the data received from A
ACK + Data
Host A sends only ACK (when it has no data to send)
ACK
Time

Error Control-

When bits are transmitted over the computer network, they are subject to corruption due to interference and network problems. The corrupted bits lead to spurious data being received by the receiver and are **called errors.**

Error detection techniques are responsible for checking whether any error has occurred or not in the frame that has been transmitted via network.

When the sender transmits data to the receiver, the data might get scrambled by noise or data might get corrupted during the transmission.



Type of Errors

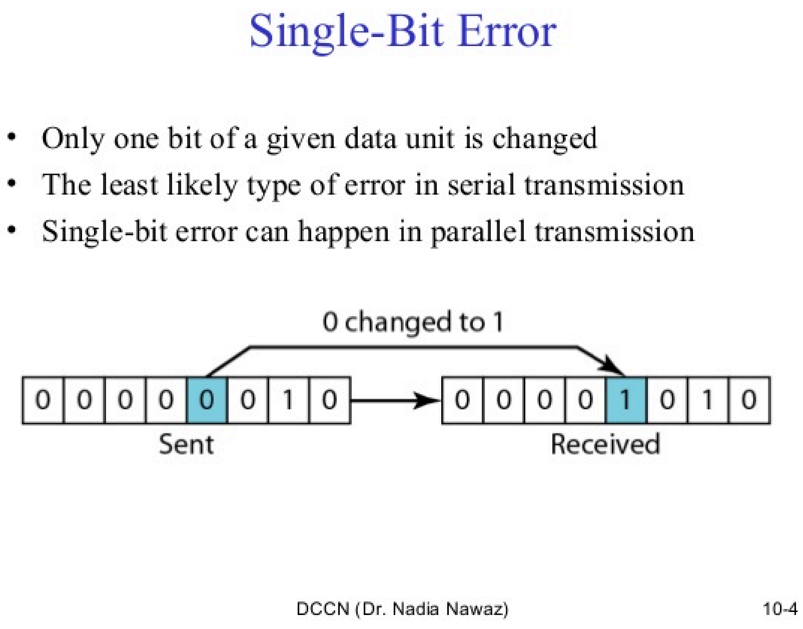
- An electromagnetic signal is subject to interference from heat, magnetism, and other forms of electricity
- Single-bit error: 0 → 1 or 1 → 0
- Burst error: 2 or more bits have changed

Errors

Single-bit

Burst

DCCN (Dr. Nadia Nawaz) 10-3



Single-Bit Error

- Only one bit of a given data unit is changed
- The least likely type of error in serial transmission
- Single-bit error can happen in parallel transmission

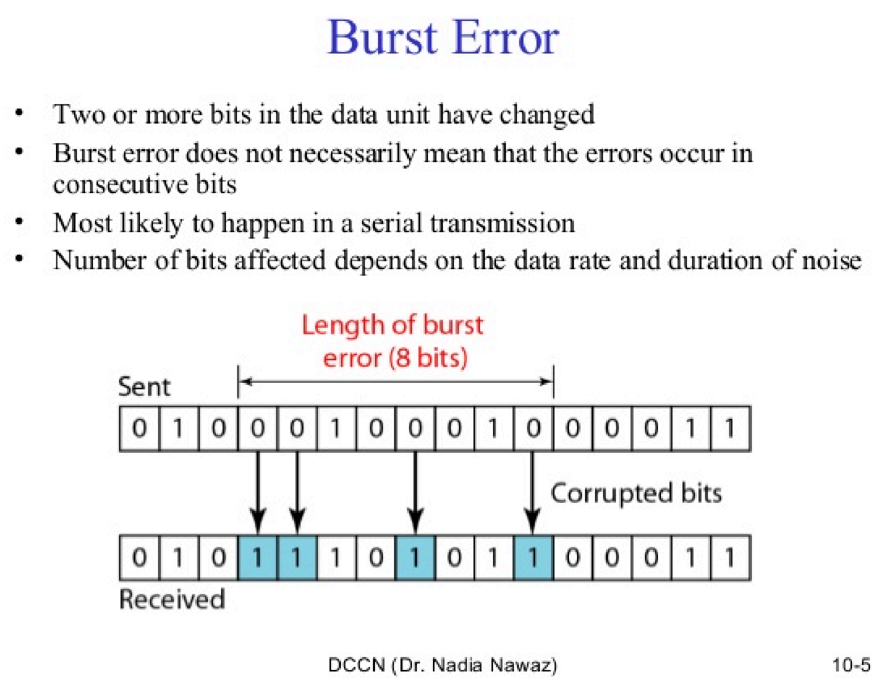
0 changed to 1

0 0 0 0 0 0 1 0 → 0 0 0 0 1 0 1 0

Sent

Received

DCCN (Dr. Nadia Nawaz) 10-4



Burst Error

- Two or more bits in the data unit have changed
- Burst error does not necessarily mean that the errors occur in consecutive bits
- Most likely to happen in a serial transmission
- Number of bits affected depends on the data rate and duration of noise

Length of burst error (8 bits)

Sent

0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 1

Corrupted bits

0 1 0 1 1 1 0 1 0 1 1 0 0 0 1 1

Received

DCCN (Dr. Nadia Nawaz) 10-5

Some popular techniques for error detection are:

1. Simple Parity check

2. Two-dimensional Parity check

3. Checksum

4. Cyclic redundancy check

1. <u>Single Parity Check-</u>

In this technique,

- One extra bit called a parity bit is sent along with the original data bits.
- Parity bit helps to check if any error occurred in the data during the transmission.

Steps Involved-

Error detection using single parity check involves the following steps-

Step-01:

At sender side,

- Total number of 1’s in the data unit to be transmitted is counted.
- The total number of 1’s in the data unit is made even in case of even parity.
- The total number of 1’s in the data unit is made odd in case of odd parity.
- This is done by adding an extra bit called the parity bit.

Step-02:

- The newly formed **code word** (Original data + parity bit) is transmitted to the receiver.

Step-03:

At receiver side,

- Receiver receives the transmitted **code word.**
- The total number of 1's in the received codeword is counted.

Then, following cases are possible-

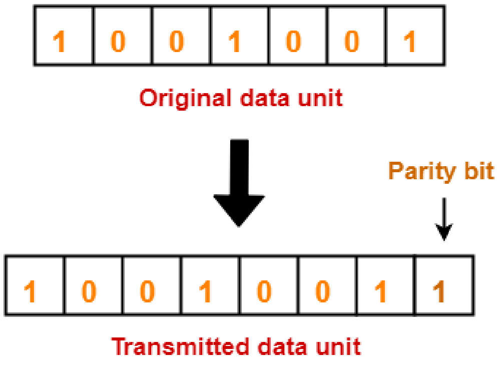
- If the total number of 1's is even and even parity is used, then the receiver assumes that no error occurred.
- If the total number of 1's is even and odd parity is used, then the receiver assumes that error occurred.
- If the total number of 1's is odd and odd parity is used, then the receiver assumes that no error occurred.
- If the total number of 1's is odd and even parity is used, then the receiver assumes that error occurred.

Parity Check Example-

Consider the data unit to be transmitted is 1001001 and even parity is used.

At Sender Side-

- Total number of 1's in the data unit is counted.
- Total number of 1's in the data unit = 3.
- Clearly, even parity is used and total number of 1's is odd.
- So, parity bit = 1 is added to the data unit to make total number of 1's even.
- Then, the code word 10010011 is transmitted to the receiver.

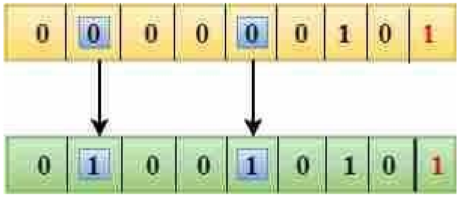


At Receiver Side-

- After receiving the code word, total number of 1's in the code word is counted.
- Consider receiver receives the correct code word = 10010011.
- Even parity is used and total number of 1's is even.
- So, receiver assumes that no error occurred in the data during the transmission.

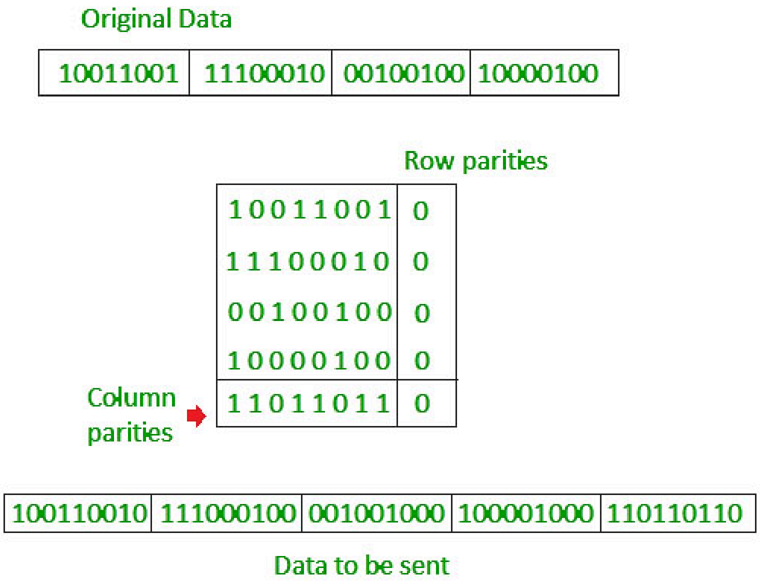
Drawbacks Of Single Parity Checking

- It can only detect single-bit errors which are very rare.
- If two bits are interchanged, then it cannot detect the errors.



2. Two-dimensional Parity check

Parity check bits are calculated for each row, which is equivalent to a simple parity check bit. Parity check bits are also calculated for all columns, then both are sent along with the data. At the receiving end these are compared with the parity bits calculated on the received data.



Original Data

10011001	11100010	00100100	10000100

Row parities

10011001	0
11100010	0
00100100	0
10000100	0
11011011	0

Column parities

100110010	111000100	001001000	100001000	110110110

Data to be sent

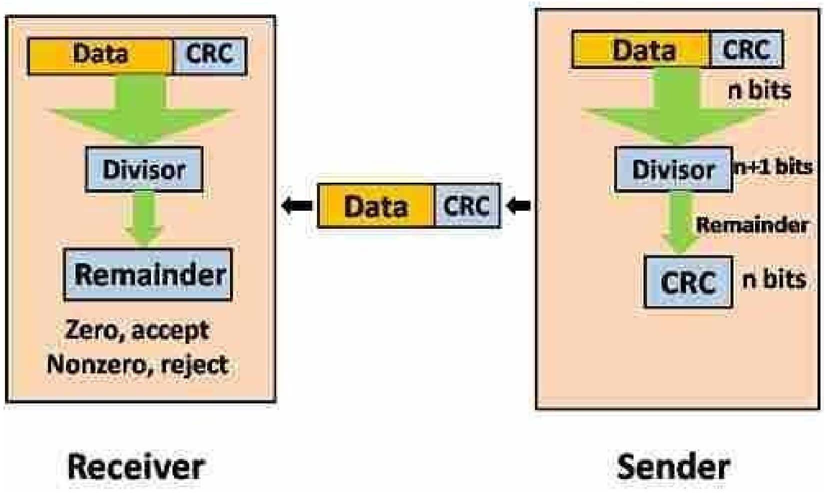
3. Cyclic Redundancy Check-

Cyclic redundancy check (CRC)

- CRC is based on binary division.
- In CRC, a sequence of bits, called cyclic redundancy check bits, are appended to the end of data unit so that the resulting data unit becomes exactly divisible by a second, predetermined binary number.
- At the destination, the incoming data unit is divided by the same number. If at this step there is no

remainder, the data unit is assumed to be correct and is therefore accepted.

- A remainder indicates that the data unit has been damaged in transit and therefore must be rejected.



- Cyclic Redundancy Check (CRC) is an error detection method.
- It is based on binary division.

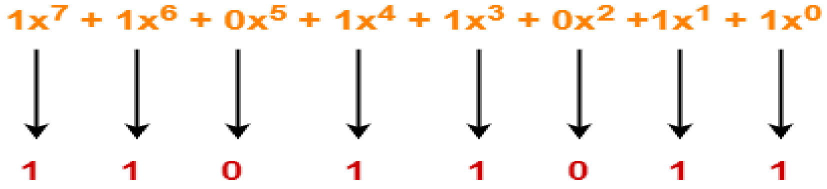
CRC Generator-

- CRC generator is an algebraic polynomial represented as a bit pattern.
- Bit pattern is obtained from the CRC generator using the following rule-

The power of each term gives the position of the bit and the coefficient gives the value of the bit.

Example-

Consider the CRC generator is $\mathbf{x^7 + x^6 + x^4 + x^3 + x + 1}$.
The corresponding binary pattern is obtained as-



Thus, for the given CRC generator, the corresponding binary pattern is 11011011.

Important Notes-

If the CRC generator is chosen according to the above rules, then-

- CRC can detect all single-bit errors

- CRC can detect all double-bit errors provided the divisor contains at least three logic 1’s.
- CRC can detect any odd number of errors provided the divisor is a factor of x+1.
- CRC can detect all burst error of length less than the degree of the polynomial.
- CRC can detect most of the larger burst errors with a high probability.

Steps Involved-

Error detection using CRC technique involves the following steps-

Step-01: *Calculation Of CRC At Sender Side-*

At sender side,

- A string of n 0’s is appended to the data unit to be transmitted.
- Here, n is one less than the number of bits in CRC generator.
- Binary division is performed of the resultant string with the CRC generator.
- After division, the remainder so obtained is called as CRC.
- It may be noted that CRC also consists of n bits.

Step-02: Appending CRC To Data Unit-

At sender side,

- The CRC is obtained after the binary division.
- The string of n 0’s appended to the data unit earlier is replaced by the CRC remainder.

Step-03: *Transmission To Receiver-*

- The newly formed code word (Original data + CRC) is transmitted to the receiver.

Step-04: Checking at Receiver Side-

At receiver side,

- The transmitted code word is received.
- The received code word is divided with the same CRC generator.
- On division, the remainder so obtained is checked.

The following two cases are possible-

Case-01: Remainder = 0

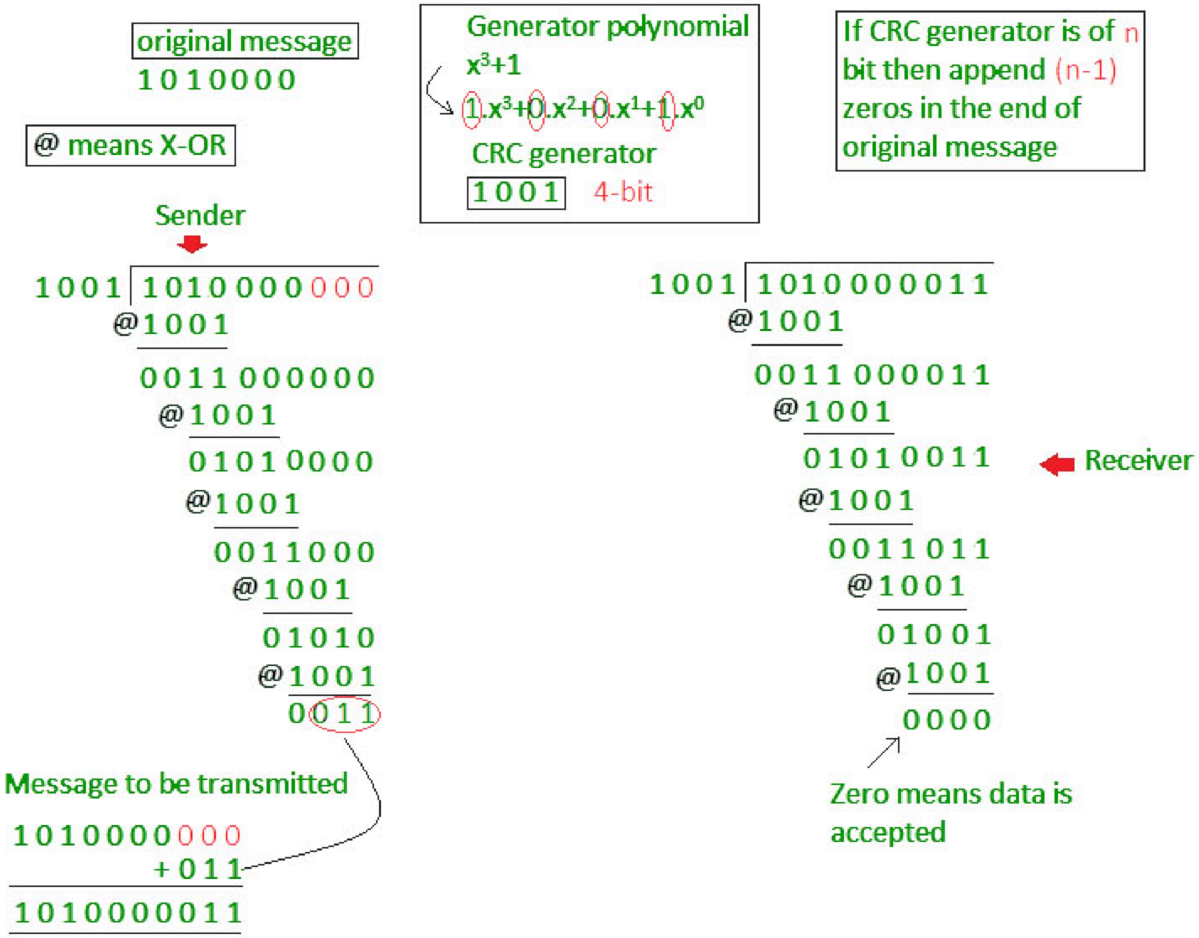
If the remainder is zero,

- Receiver assumes that no error occurred in the data during the transmission.
- Receiver accepts the data.

Case-02: Remainder ≠ 0

If the remainder is non-zero,

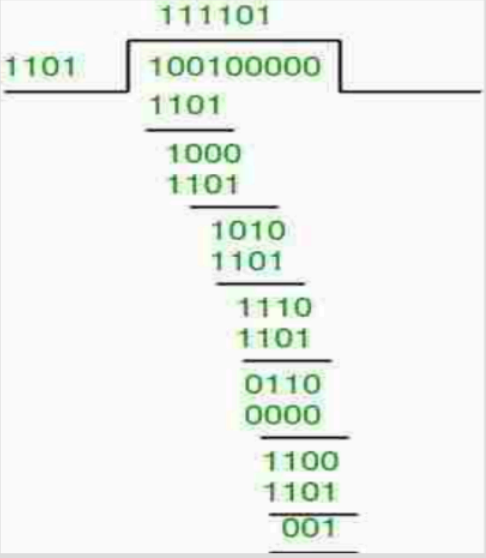
- Receiver assumes that some error occurred in the data during the transmission.
- Receiver rejects the data and asks the sender for retransmission.



Example 1 (No error in transmission):

Data word to be sent - 100100
Key - 1101 [Or generator polynomial $x^3 + x^2 + 1$]

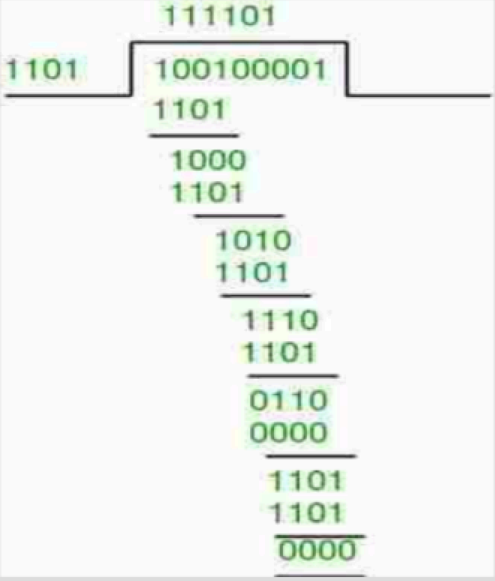
Sender Side:



Therefore, the remainder is 001 and hence the encoded data sent is 100100001.

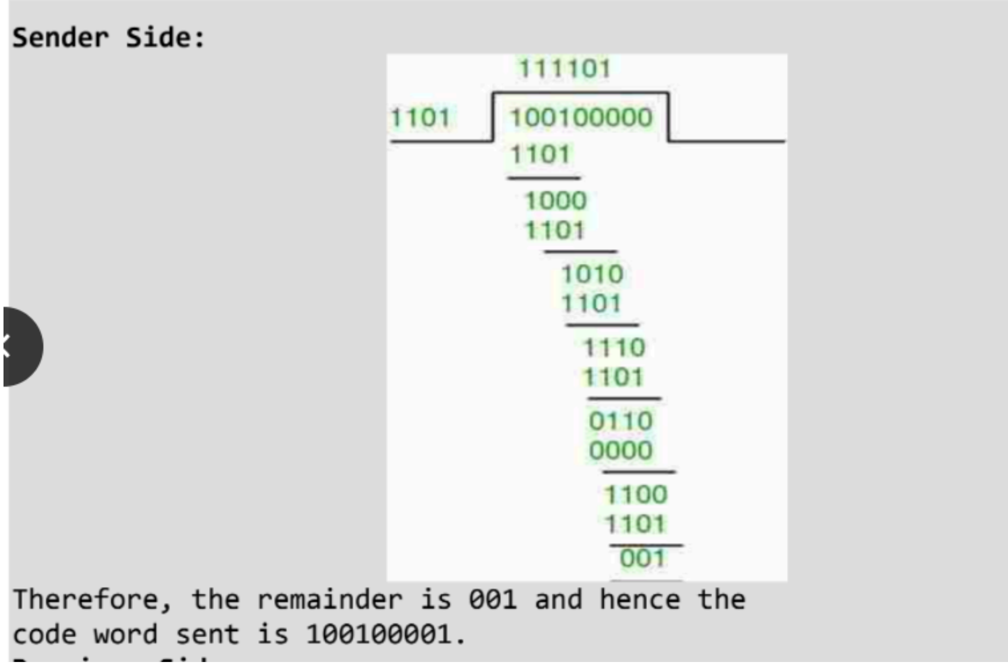
Receiver Side:

Code word received at the receiver side 100100001



Therefore, the remainder is all zeros. Hence, the data received has no error.

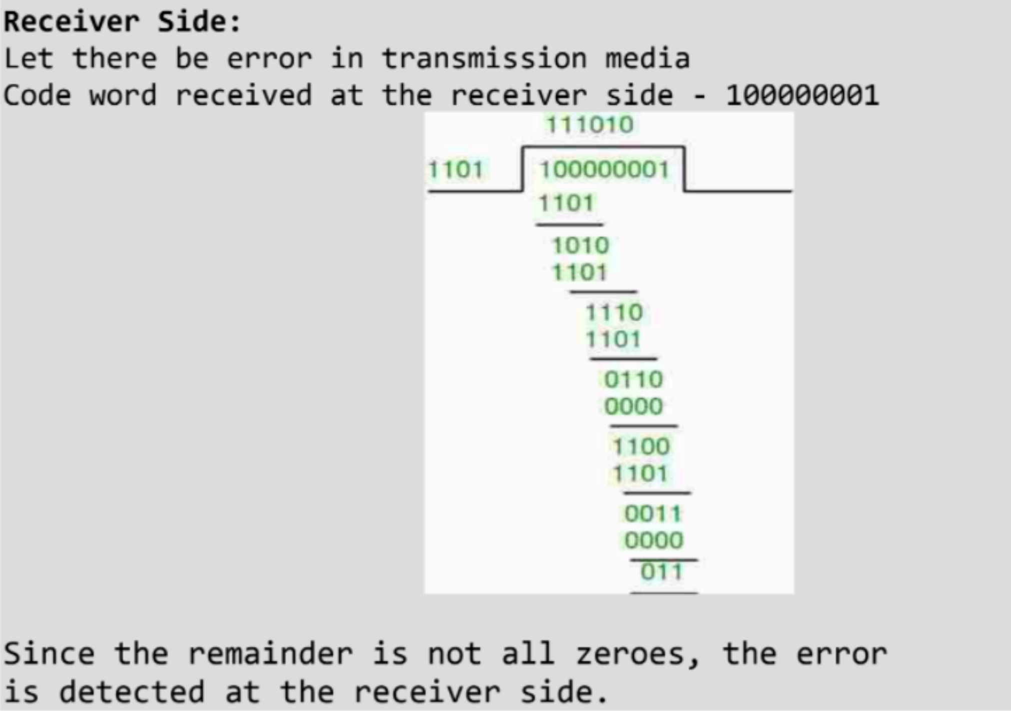
Example 2:
Data word to be sent - 100100
Key - 1101 [Or generator polynomial $x^3 + x^2 + 1$]



Sender Side:

```
            111101
1101  | 100100000
        1101
        ----
         1000
         1101
         ----
          1010
          1101
          ----
           1110
           1101
           ----
            0110
            0000
            ----
             1100
             1101
             ----
              001
```

Therefore, the remainder is 001 and hence the code word sent is 100100001.



Receiver Side:
Let there be error in transmission media
Code word received at the receiver side - 100000001

```
            111010
1101  | 100000001
        1101
        ----
         1010
         1101
         ----
          1110
          1101
          ----
           0110
           0000
           ----
            1100
            1101
            ----
             0011
             0000
             ----
              011
```

Since the remainder is not all zeroes, the error is detected at the receiver side.

Problem-01:

A bit stream 1101011011 is transmitted using the standard CRC method. The generator polynomial is x^4+x+1. What is the actual bit string transmitted?

Solution-

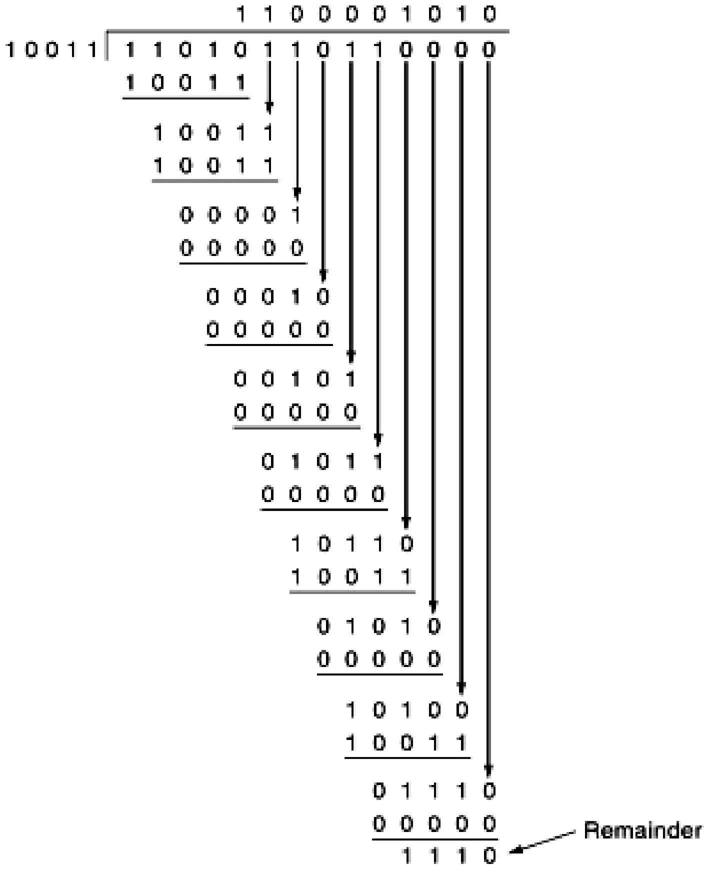
The generator polynomial G(x) = $x^4 + x + 1$ is encoded as 10011. (i.e.x^4x^3 x^2 x^1x^0)

Clearly, the generator polynomial consists of 5 bits.

So, a string of 4 zeroes is appended to the bit stream to be transmitted.

The resulting bit stream is 1101011011**0000**.

Now, the binary division is performed as-



From here, CRC = 1110.

Now,

The code word to be transmitted is obtained by replacing the last 4 zeroes of 1101011011**0000** with the CRC.

Thus, the code word transmitted to the receiver = 1101011011**1110**.

Problem-02:

A bit stream 10011101 is transmitted using the standard CRC method. The generator polynomial is x^3+1.l

1. What is the actual bit string transmitted?
2. Suppose the third bit from the left is inverted during transmission. How will receiver detect this error?

Solution-

Part-01:

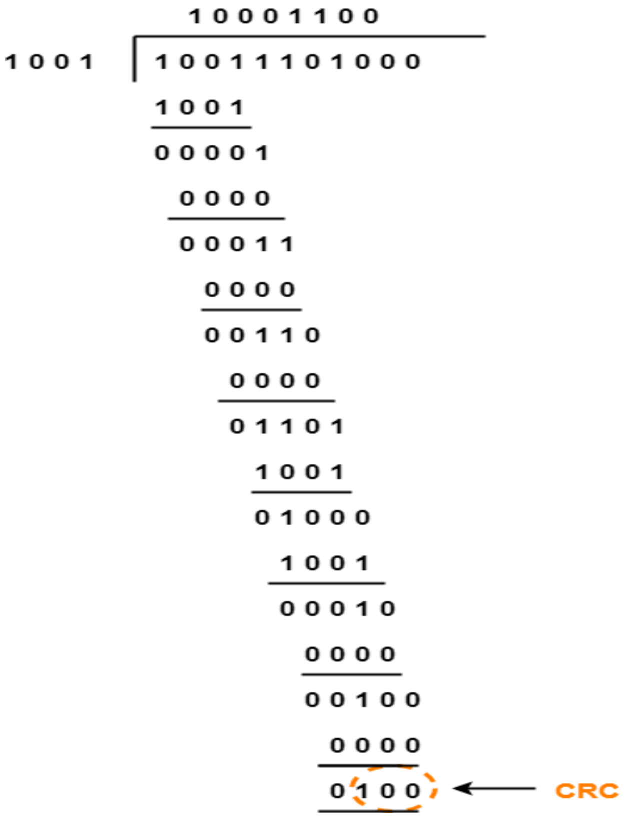
The generator polynomial $G(x) = x^3 + 1$ is encoded as 1001.

Clearly, the generator polynomial consists of 4 bits.

So, a string of 3 zeroes is appended to the bit stream to be transmitted.

The resulting bit stream is 10011101**000**.

Now, the binary division is performed as-



```
             1 0 0 0 1 1 0 0
        ________________________
1 0 0 1 | 1 0 0 1 1 1 0 1 0 0 0
          1 0 0 1
          -------
          0 0 0 0 1
            0 0 0 0
            -------
            0 0 0 1 1
              0 0 0 0
              -------
              0 0 1 1 0
                0 0 0 0
                -------
                0 1 1 0 1
                  1 0 0 1
                  -------
                  0 1 0 0 0
                    1 0 0 1
                    -------
                    0 0 0 1 0
                      0 0 0 0
                      -------
                      0 0 1 0 0
                        0 0 0 0
                        -------
                        0 1 0 0  ←  CRC
                        -------
```

From here, CRC = 100.

Now,

The code word to be transmitted is obtained by replacing the last 3 zeroes of 10011101**000** with the CRC.

Thus, the code word transmitted to the receiver = 10011101**100**.

Part-02:

According to the question,

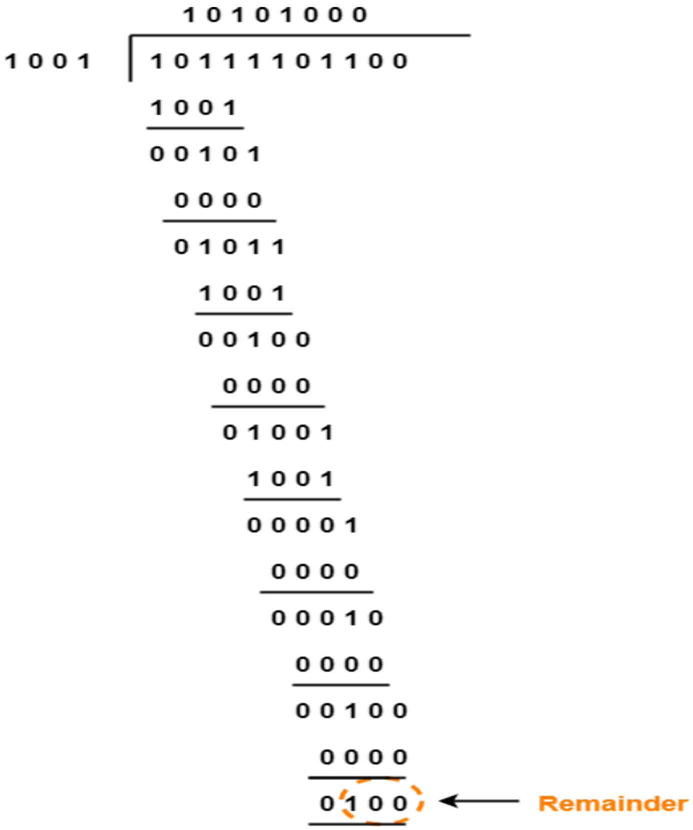
Third bit from the left gets inverted during transmission.

So, the bit stream received by the receiver = 10111101100.

Now,

Receiver receives the bit stream = 10111101100.

Receiver performs the binary division with the same generator polynomial as-



From here,

The remainder obtained on division is a non-zero value.

This indicates to the receiver that an error occurred in the data during the transmission.

Therefore, receiver rejects the data and asks the sender for **retransmission**.

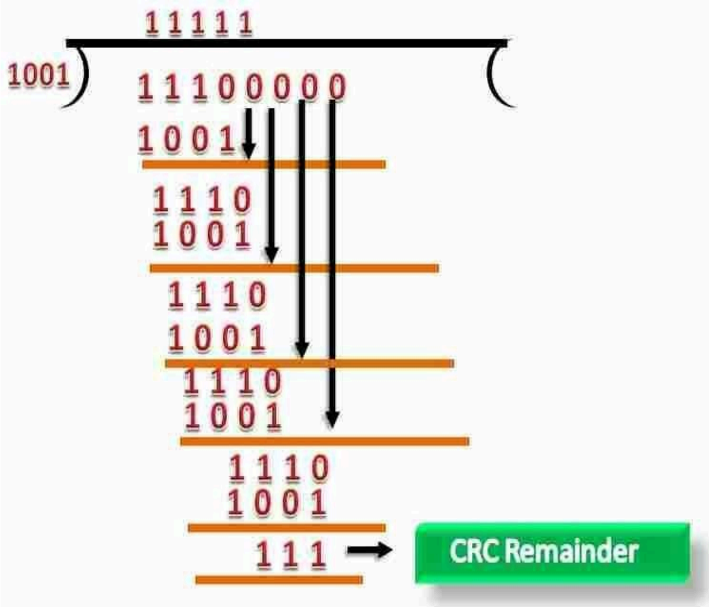
Example: **Suppose the original data is 11100 and divisor is 1001 CRC.**

Solution

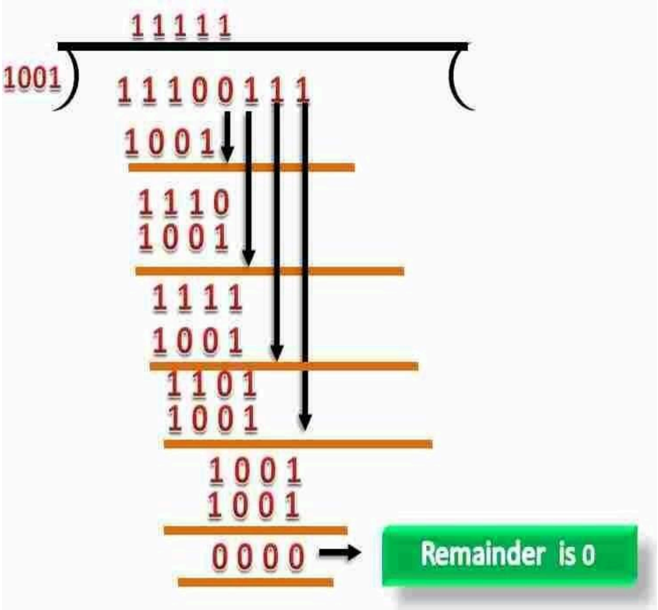
- A CRC generator uses a modulo-2 division. Firstly, three zeroes are appended at the end of the data as the length of the divisor is 4 and we know that the length of the string 0s to be appended is always one less than the length of the divisor.

- Now, the string becomes 11100000, and the resultant string is divided by the divisor 1001.
- The remainder generated from the binary division is known as CRC remainder. The generated value of the CRC remainder is 111.
- CRC remainder replaces the appended string of 0s at the end of the data unit, and the final string would be 11100111 which is sent across the network.

Sender

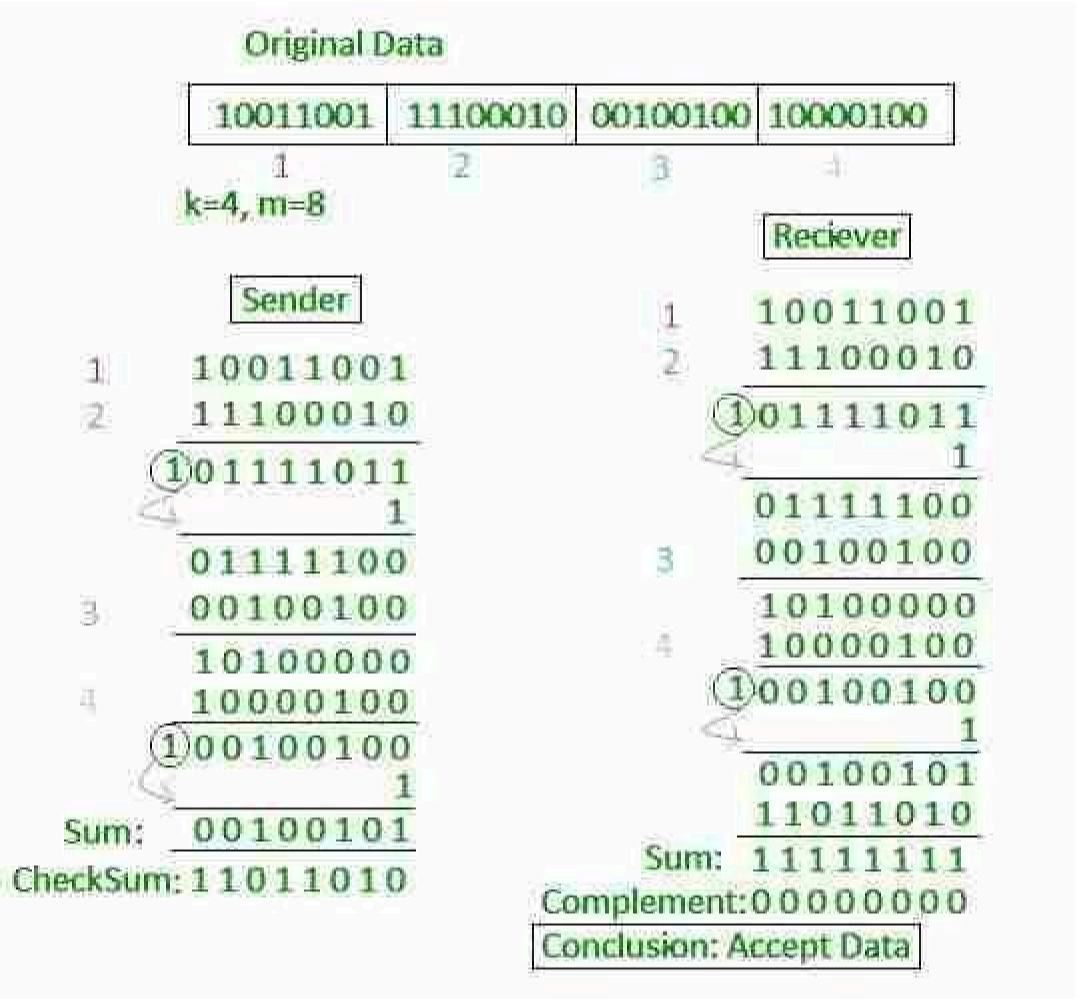


Receiver



Checksum

- In checksum error detection scheme, the data is divided into k segments each of m bits.
- In the sender's end the segments are added using 1's complement arithmetic to get the sum. The sum is complemented to get the checksum.
- The checksum segment is sent along with the data segments.
- At the receiver's end, all received segments are added using 1's complement arithmetic to get the sum. The sum is complemented.
- If the result is zero, the received data is accepted; otherwise discarded.



A Checksum is an error detection technique based on the concept of redundancy. Error detection using checksum method involves the following steps-

Step-01:

At sender side,

If m bit checksum is used, the data unit to be transmitted is divided into segments of m bits.
All the m bit segments are added.
The result of the sum is then complemented using 1's complement arithmetic.

The value so obtained is called as **checksum**.

Step-02:

- The data along with the checksum value is transmitted to the receiver.

Step-03:

At receiver side,

If m bit checksum is being used, the received data unit is divided into segments of m bits.
All the m bit segments are added along with the checksum value.
The value so obtained is complemented and the result is checked.

Then, following two cases are possible-

Case-01: Result = 0

If the result is zero,

Receiver assumes that no error occurred in the data during the transmission.
Receiver accepts the data.

Case-02: Result ≠ 0

If the result is non-zero,

- Receiver assumes that error occurred in the data during the transmission.
- Receiver discards the data and asks the sender for retransmission.

Checksum Example-

Consider the data unit to be transmitted is-

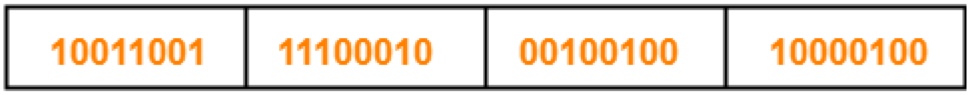
10011001111000100010010010000100

Consider 8 bit checksum is used.

Step-01:

At sender side,

The given data unit is divided into segments of 8 bits as-



10011001	11100010	00100100	10000100

Now, all the segments are added and the result is obtained as-

- 10011001 + 11100010 + 00100100 + 10000100 = 1000100011
- Since the result consists of 10 bits, so extra 2 bits are wrapped around.
- 00100011 + 10 = 00100101 (8 bits)
- Now, 1's complement is taken which is 11011010.
- Thus, checksum value = 11011010

Step-02:

The data along with the checksum value is transmitted to the receiver.

Step-03:

At receiver side,

- The received data unit is divided into segments of 8 bits.
- All the segments along with the checksum value are added.
- Sum of all segments + Checksum value = 00100101 + 11011010 = 11111111
- Complemented value = 00000000
- Since the result is 0, receiver assumes no error occurred in the data and therefore accepts it.

Example 2:

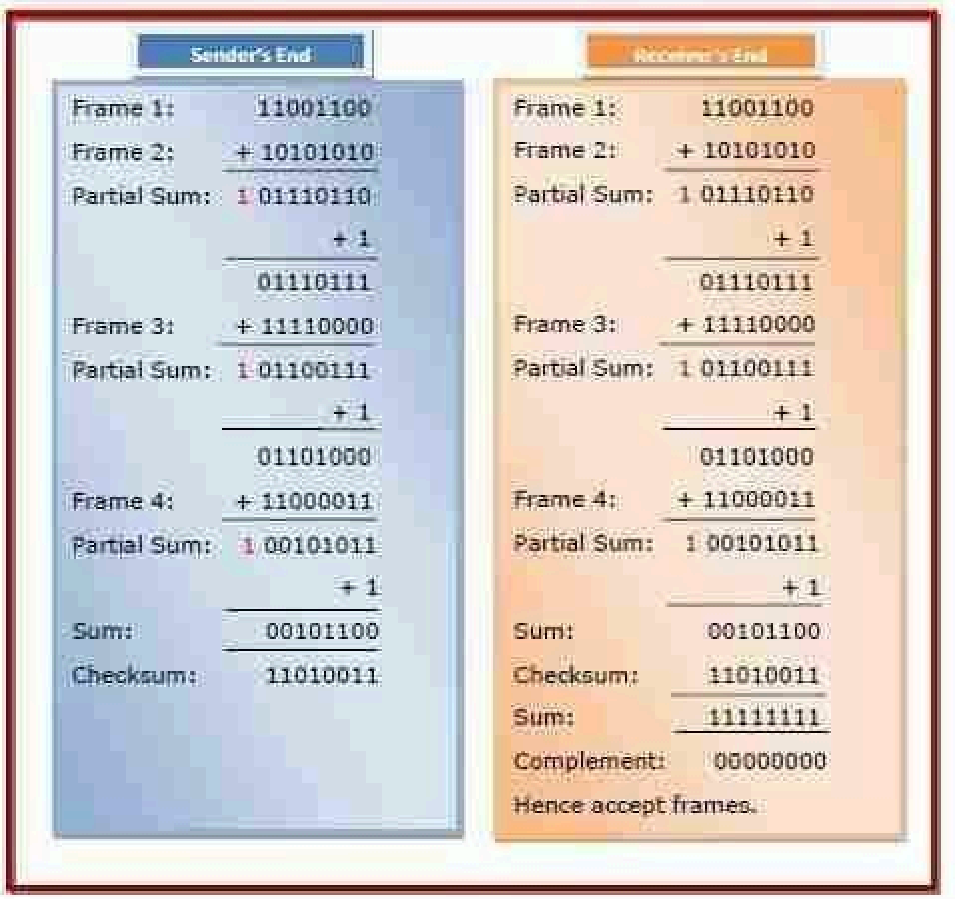
Suppose that the sender wants to send 4 frames each of 8 bits, where the frames are 11001100, 10101010, 11110000 and 11000011.

Solution:

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.



Sender's End

```
Frame 1:          11001100
Frame 2:        + 10101010
Partial Sum:    1 01110110
                       + 1
                  01110111
Frame 3:        + 11110000
Partial Sum:    1 01100111
                       + 1
                  01101000
Frame 4:        + 11000011
Partial Sum:    1 00101011
                       + 1
Sum:              00101100
Checksum:         11010011
```

Receiver's End

```
Frame 1:          11001100
Frame 2:        + 10101010
Partial Sum:    1 01110110
                       + 1
                  01110111
Frame 3:        + 11110000
Partial Sum:    1 01100111
                       + 1
                  01101000
Frame 4:        + 11000011
Partial Sum:    1 00101011
                       + 1
Sum:              00101100
Checksum:         11010011
Sum:              11111111
Complement:       00000000
Hence accept frames.
```

(Please see short details of Checksum example in above)

Example 3:

3.3. ERROR DETECTION AND CORRECTION

Error-Detecting codes

Whenever a message is transmitted, it may get scrambled by noise or data may get corrupted. To avoid this, we use error-detecting codes which are additional data added to a given digital message to help us detect if an error occurred during transmission of the message. A simple example of error-detecting code is **parity check**.

Error-Correcting codes

Along with error-detecting code, we can also pass some data to figure out the original message from the corrupt message that we received. This type of code is called an error-correcting code. Error-correcting codes also deploy the same strategy as error-detecting codes but additionally, such codes also detect the exact location of the corrupt bit.

In error-correcting codes, parity check has a simple way to detect errors along with a sophisticated mechanism to determine the corrupt bit location. Once the corrupt bit is located, its value is reverted (from 0 to 1 or 1 to 0) to get the original message.

How to Detect and Correct Errors?

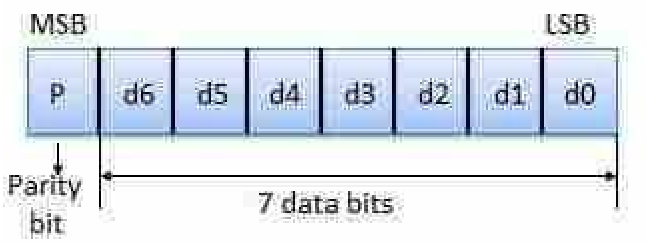
To detect and correct the errors, additional bits are added to the data bits at the time of transmission.

The additional bits are called **parity bits**. They allow detection or correction of the errors.

The data bits along with the parity bits form a **code word**.

Parity Checking of Error Detection

It is the simplest technique for detecting and correcting errors. The MSB of an 8-bits word is used as the parity bit and the remaining 7 bits are used as data or message bits. The parity of 8-bits transmitted word can be either even parity or odd parity.



Even parity -- Even parity means the number of 1's in the given word including the parity bit should be even (2,4,6,.).

Odd parity -- Odd parity means the number of 1's in the given word including the parity bit should be odd (1,3,5,.).

Use of Parity Bit

The parity bit can be set to 0 and 1 depending on the type of the parity required.

For even parity, this bit is set to 1 or 0 such that the no. of "1 bits" in the entire word is even. Shown in fig. (a).

For odd parity, this bit is set to 1 or 0 such that the no. of "1 bits" in the entire word is odd. Shown in fig. (b).

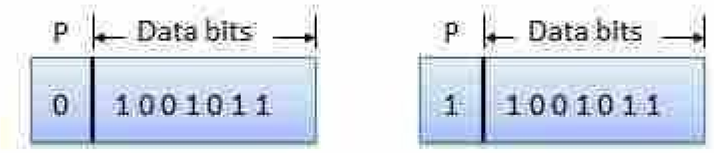


Fig. (a)

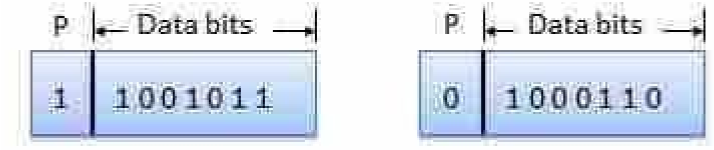
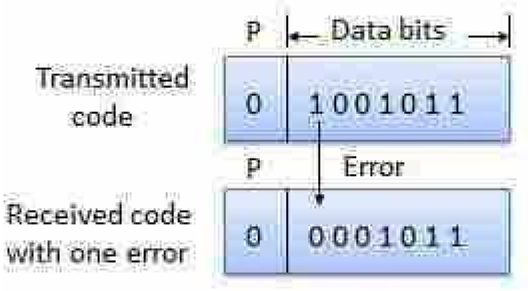


Fig. (b)

How Does Error Detection Take Place?

Parity checking at the receiver can detect the presence of an error if the parity of the receiver signal is different from the expected parity. That means, if it is known that the parity of the transmitted signal is always going to be "even" and if the received signal has an odd parity, then the receiver can conclude that the received signal is not correct. If an error is detected, then the receiver will ignore the received byte and request for retransmission of the same byte to the transmitter.



More research

https://computernetwork-mmc.blogspot.com/2020/04/error-correction-hamming-code-in_15.html

Hamming Code

ASSIGNMENT

3.4. HDLC AND PPP

High-Level Data Link Control(HDLC)

High-level Data Link Control (HDLC) 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.

High-level Data Link Control (HDLC) is a **bit-oriented protocol for communication** over

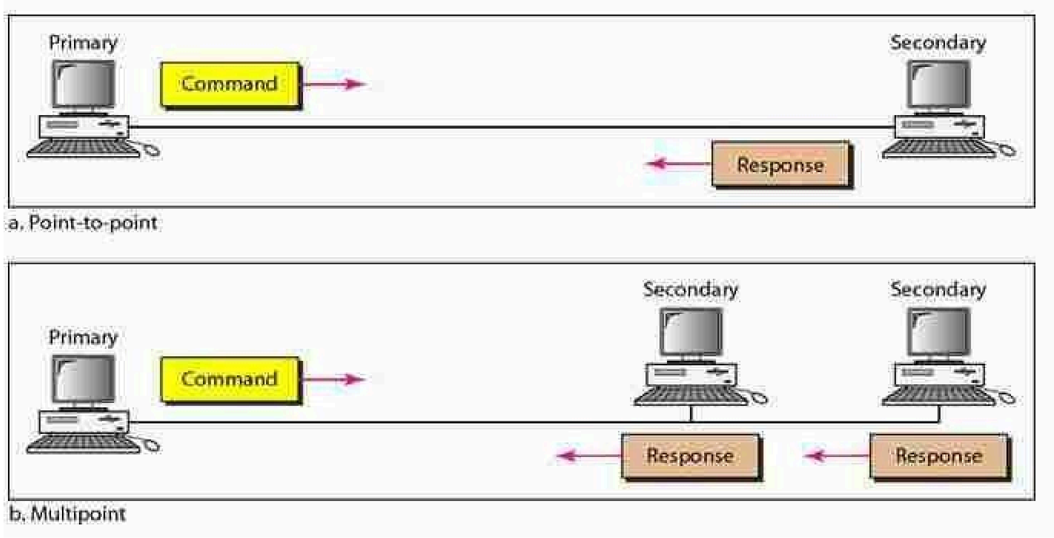
point-to- point and multipoint links. It implements the ARQ mechanisms.

Configurations and Transfer Modes:

HDLC provides two common transfer modes that can be used in different configurations: normal response mode (NRM) and asynchronous balanced mode (ABM).

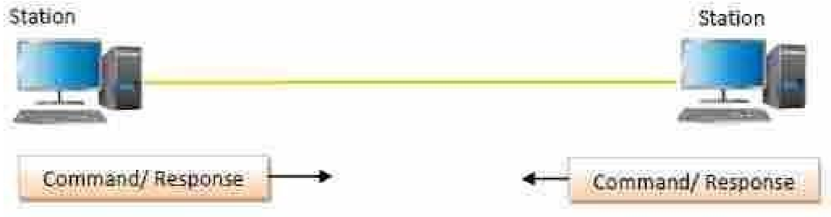
Normal Response Mode:

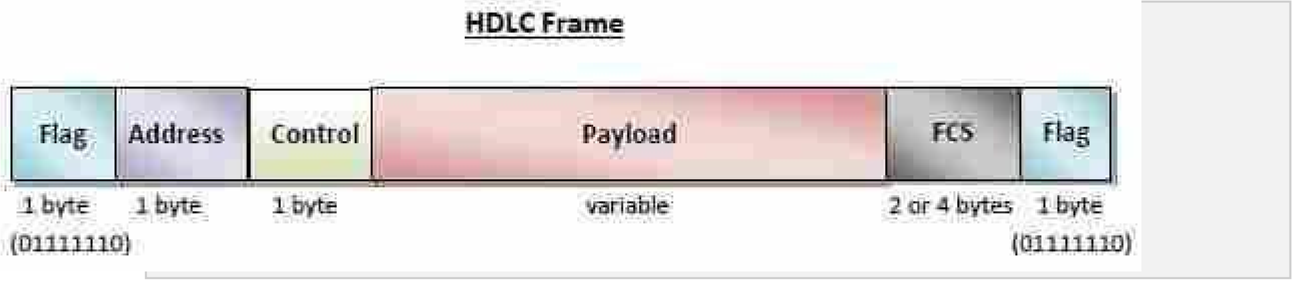
Here, two types of stations are there, a primary station that sends commands and a secondary station that can respond to received commands. It is used for both point - to - point and multipoint communications.



Asynchronous Balanced Mode:

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.





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 bi t pattern of the flag is 01111110.

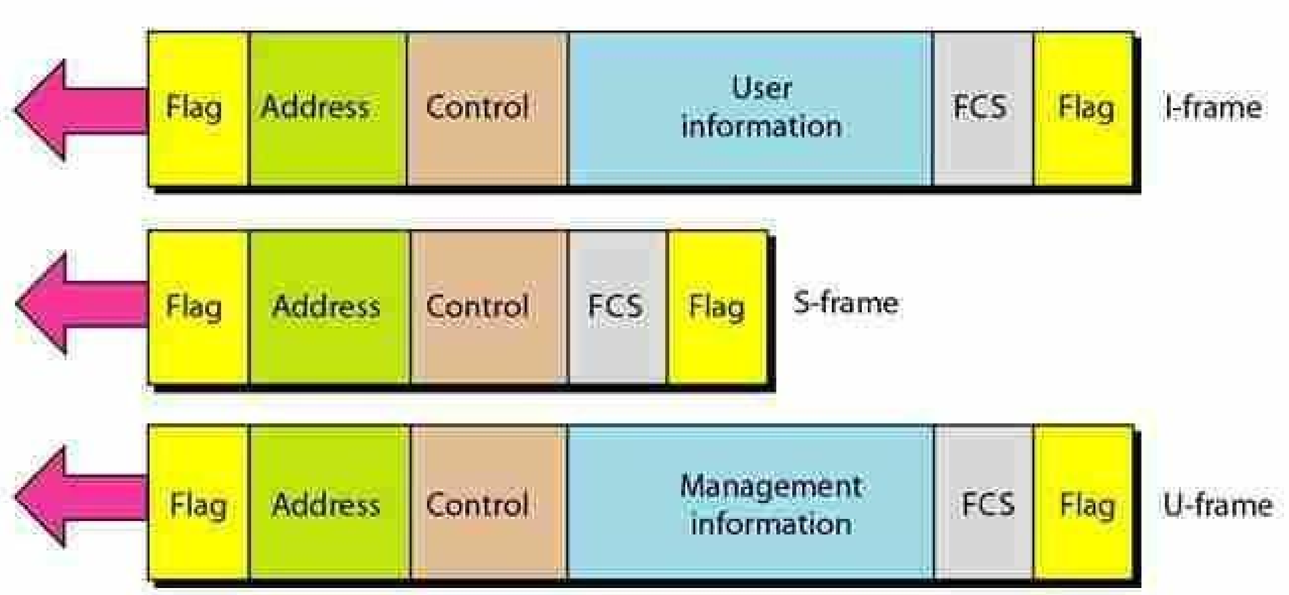
Address − It contains the address of the receiver. If the frame is sent by the primary station, it contains the address(es) of the secondary station(s). If it is sent by the secondary station, i t contains the address of the primary station. The address field may be from 1 byte to several bytes.

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



Point - to - Point Protocol (PPP)

Point - to - Point Protocol (PPP) is a communication protocol of the data link layer that is used to transmit multiprotocol data between two directly connected (point-to-point) computers.

It is a **byte - oriented protocol** that i s widely used in broadband communications having heavy loads and high speeds.

Since it is a data link layer protocol, data a i s t transmitted i n frames. It is also known as RFC 1661.

One of the most common protocols for point-to-point access

Many Internet users who need to connect their home computer to the server o f an Internet service provider use PPP

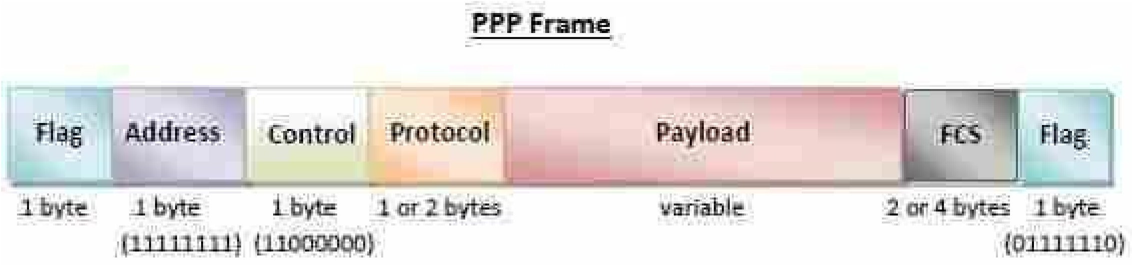
A point-to-point link protocol is required to control and manage the transfer of data

PPP defines/provides

- ✔ the format of the frame to be exchanged between devices
- ✔ how network layer data are encapsulated in the data link frame
- ✔ how two devices can authenticate each other
- ✔ multiple network layer services
- ✔ connection over multiple links
- ✔ Network address configuration

But, several services are missing for simplicity

- ✔ no flow control, simple error control (detection and discard), no sophisticate addressing for multipoint configuration



Flag − marks the beginning and the end of the frame. The bit pattern of the flag is 01111110.

Address – it is set to 11111111 in case of broadcast.

Control − set to a constant value of 11000000.(No need because PPP has no flow control and limited error control)

Protocol − 1 or 2 bytes that define the type of data contained in the payload field.

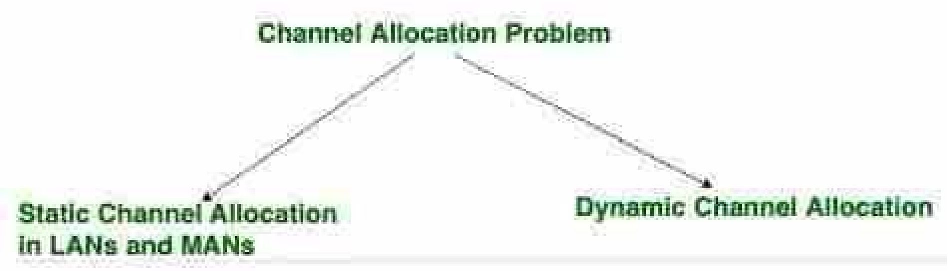
Payload − This carries the data from the network layer. The maximum length of the payload field is 1500 bytes. However, this may be negotiated between the endpoints of communication.

FCS − It is a 2 byte or 4 bytes frame check sequence for error detection. The standard code used is CRC (cyclic redundancy code)

- *PPP is a byte-oriented protocol using byte stuffing with the escape byte 01111101*

3.5. CHANNEL ALLOCATION PROBLEM

- When there are more than one user who desires to access a shared network channel, an algorithm is deployed for channel allocation among the competing users.
- The network channel may be a single cable or optical fiber connecting multiple nodes, or a portion of the wireless spectrum.
- Channel allocation algorithms allocate the wired channels and bandwidths to the users, who may be base stations, access points or terminal equipment.
- ❖ Channel allocation is a process in which a single channel is divided and allotted to multiple users in order to carry user specific tasks. The user’s quantity may vary every time the process takes place.
- ❖ Channel allocation problem can be solved by two schemes: Static Channel Allocation in LANs and MANs, and Dynamic Channel Allocation.


Channel Allocation Problem
Static Channel Allocation in LANs and MANs
Dynamic Channel Allocation

Static Channel Allocation

In the static channel allocation scheme, a fixed portion of the frequency channel is allotted to each user. For N competing users, the bandwidth is divided into N channels using frequency division multiplexing (FDM), and each portion is assigned to one user.

This scheme is also referred to as fixed channel allocation or fixed channel assignment.

It is not efficient to divide into a fixed number of chunks.

Dynamic Channel Allocation

In the dynamic channel allocation scheme, frequency bands are not permanently assigned to the users. Instead channels are allotted to users dynamically as needed, from a central pool. The allocation is done considering a number of parameters so that transmission interference is minimized.
This allocation scheme optimizes bandwidth usage and results in faster transmissions.

Dynamic channel allocation is further divided into centralized and distributed allocation.

Possible assumptions include:

1. **Single Channel Assumption:**
   In this allocation all stations are equivalent and can send and receive on that channel.

2. **Collision Assumption:**
   If two frames overlap in time-wise, then that's a collision. Any collision is an error, and both frames must be re-transmitted. Collisions are only possible errors.

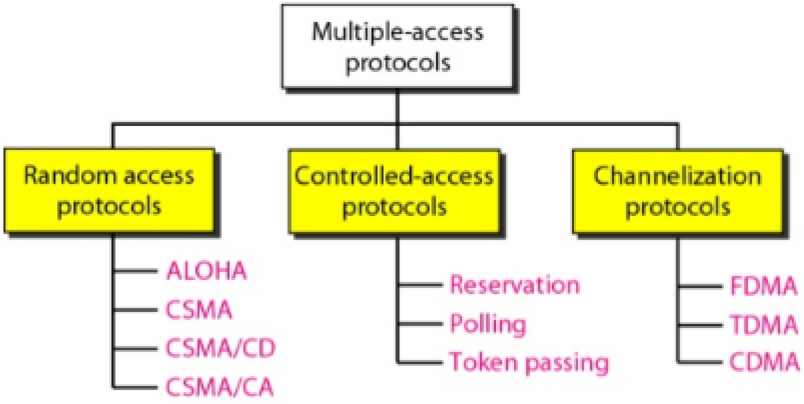
3. **Time** can be divided into Slotted or Continuous.

4. **Stations** can sense a channel is busy before they try it.

3.6. MULTIPLE ACCESS PROTOCOL:

Multiple access protocols are a set of protocols operating in the Medium Access Control sublayer (MAC sublayer) of the Open Systems Interconnection (OSI) model. These protocols allow a number of nodes or users to access a shared network channel.

1. **Random Access(ALOHA, CSMA, CSMA CD, CSMA/CA),**
2. **Controlled Access(Reservation, Polling, Token Passing),**
3. **Channelization(FDMA, TDMA, CDMA)**


Multiple-access protocols
Random access protocols
ALOHA
CSMA
CSMA/CD
CSMA/CA
Controlled-access protocols
Reservation
Polling
Token passing
Channelization protocols
FDMA
TDMA
CDMA

1. Random Access Protocol

• In random access or contention methods, no station is superior to another station and none is assigned the control over another
• No station permits, or does not permit, another station to send(Randomly send if medium is free)

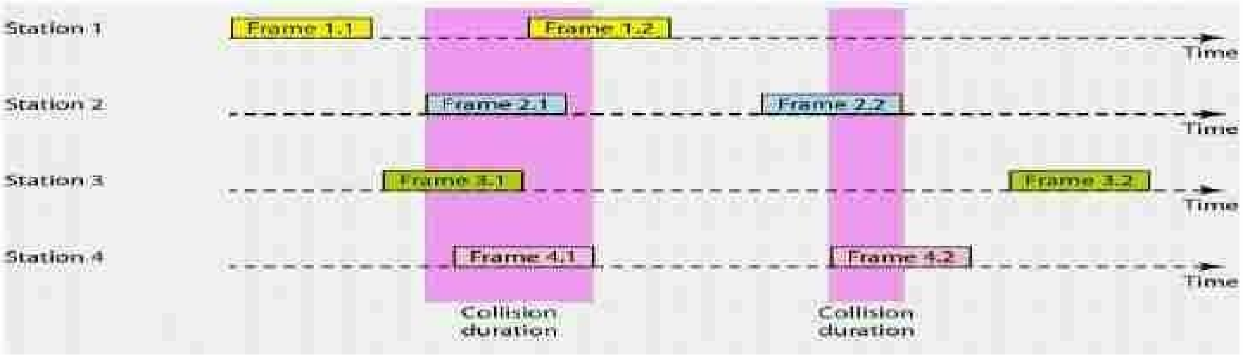
I) ALOHA

The Aloha protocol was designed as part of a project at the University of Hawaii. It provided data transmission between computers on several of the Hawaiian Islands involving packet radio networks. Aloha is a multiple access protocol at the data link layer and proposes how multiple terminals access the medium without interference or collision.

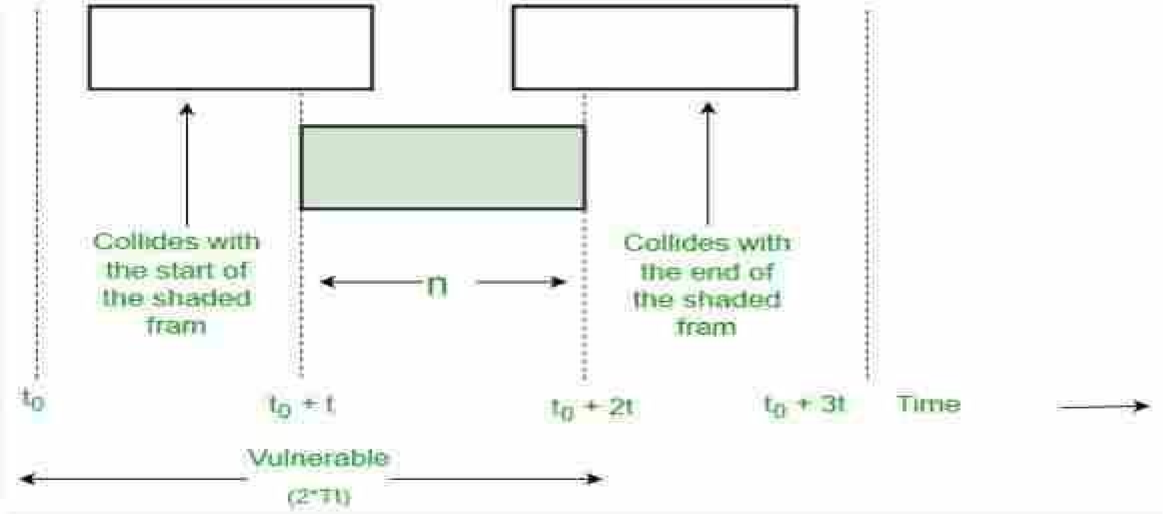
There are two different versions of ALOHA:

Pure Aloha

- Pure Aloha is a simple to implement a protocol.
- In pure ALOHA, the stations simply transmit frames whenever they want data to send.
- It does not check whether the channel is busy or not before transmitting.
- In case, two or more stations transmit simultaneously, the collision occurs and frames are destroyed.
- Whenever any station transmits a frame, it expects the acknowledgment from the receiver. If it is not received within a specified time, the station assumes that the frame or acknowledgment has been destroyed.
- Then, the station waits for a random amount of time and sends the frame again.
- This randomness helps in avoiding more collisions.
- This scheme works well in small networks where the load is not much.
- But in largely loaded networks, this scheme fails poorly.
- This led to the development of Slotted Aloha.
- To assure pure aloha: Its throughput and rate of transmission of the frame to be predicted.


Station 1
Frame 1.1
Frame 1.2
Time
Station 2
Frame 2.1
Frame 2.2
Time
Station 3
Frame 3.1
Frame 3.2
Time
Station 4
Frame 4.1
Frame 4.2
Time
Collision duration
Collision duration

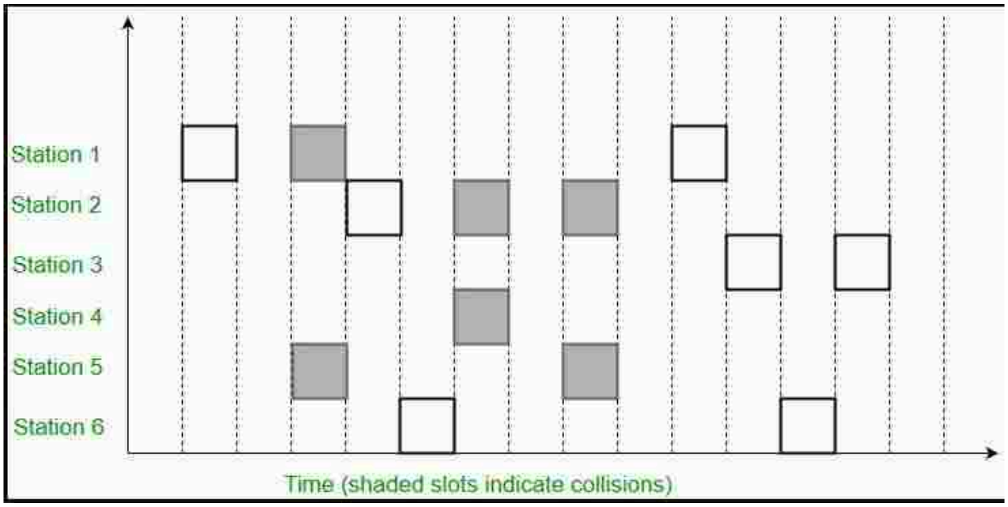
Overlapping frames in the pure ALOHA protocol. Frame-time is equal to 1 for all frames.

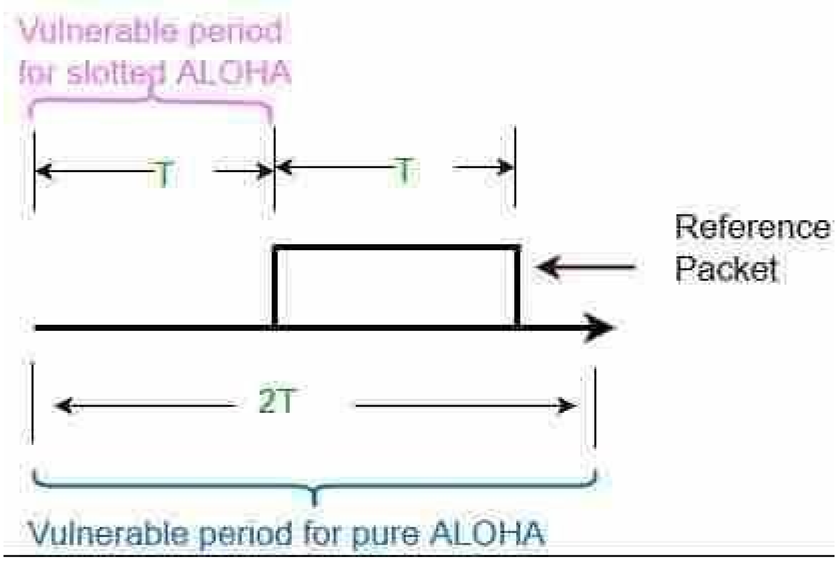

Collides with the start of the shaded fram
Collides with the end of the shaded fram
Tt
t0
t0 + t
t0 + 2t
t0 + 3t
Time
Vulnerable
(2*Tt)

Vulnerable time in which collision may occur = 2 x T_t

Slotted Aloha

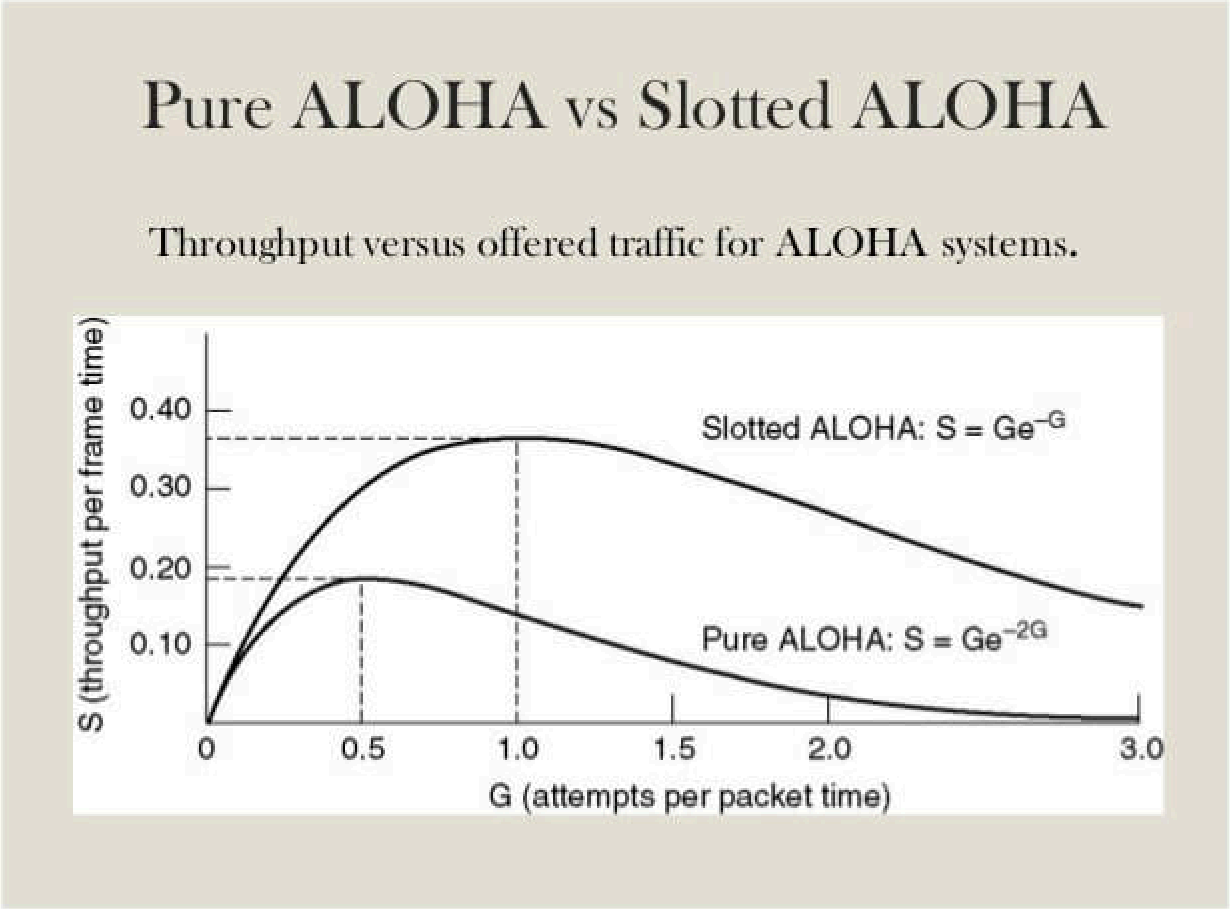
- This is quite similar to Pure Aloha, differing only in the way transmissions take place.
- Instead of transmitting right at demand time, the sender waits for some time.
- In slotted ALOHA, the time of the shared channel is divided into discrete intervals called *Slots*.
- The stations are eligible to send a frame only at the beginning of the slot and only one frame per slot is sent.
- If any station is not able to place the frame onto the channel at the beginning of the slot, it has to wait until the beginning of the next time slot.
- There is still a possibility of collision if two stations try to send at the beginning of the same time slot.
- But still the number of collisions that can possibly take place is reduced by a large margin and the performance becomes much well compared to Pure Aloha.





Collision is possible for only the current slot. Therefore, Vulnerable Time is Tt.

Efficiency of Pure and Slotted ALOHA:



Pure ALOHA vs Slotted ALOHA

Throughput versus offered traffic for ALOHA systems.

S=Maximum efficiency = 18.4%.(PURE ALOHA)	S=Maximum efficiency = 36.8%.(SLOTTED ALOHA)

Following are the important differences between Pure Aloha and Slotted Aloha.

Sr. No.	Key	Pure Aloha	Slotted Aloha

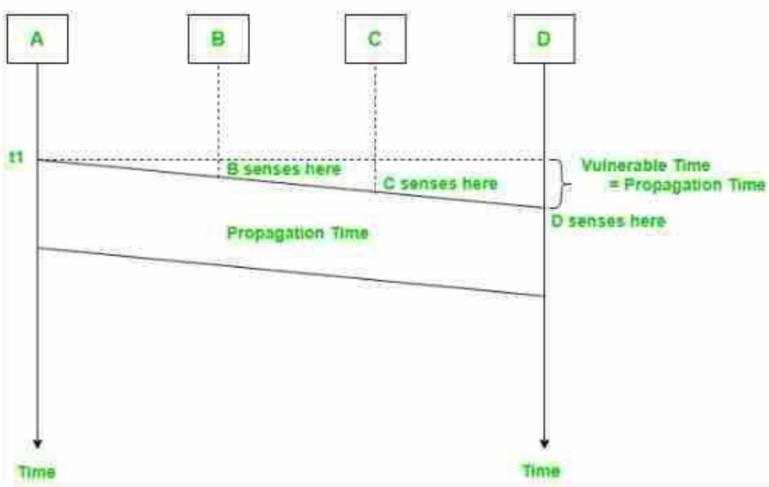
1	Time Slot	In Pure Aloha, any station can transmit data at any time.	In Slotted Aloha, any station can transmit data only at beginning of any time slot.
2	Time	In Pure Aloha, time is continous and is not globally syncronized.	In Slotted Aloha, time is discrete and is globally syncronized.
3	Vulnerable time	Vulnerable time = 2 x Tt.	Vulnerable time = Tt.
4	Probability	Probability of successful transmission of data packet = G x e^{-2G}	Probability of successful transmission of data packet = G x e^{-G}
5	Maximum efficiency	Maximum efficiency = 18.4%.	Maximum efficiency = 36.8%.
6	Number of collisions	Does to reduces the number of collisions.	Slotted Aloha reduces the number of collisions to half thus doubles the efficiency.

II) Carrier Sense Multiple Access (CSMA):

- Invented to minimize collisions and increase the performance
- A station now “follows” the activity of other stations
- Simple rules for a polite human conversation
    1. Listen before talking
    2. If someone else begins talking at the same time as you, stop talking
- A node should not send if another node is already sending(Carrier Sensing)
- Vulnerable time is the propagation time which is the time needed for a signal to propagate from one end of the medium to the other
- **Carrier Sense Multiple Access** (**CSMA**) is a probabilistic Media Access Control (MAC) protocol in

which a node verifies the absence of other traffic before transmitting on a shared transmission medium, such as an electrical bus, or a band of the electromagnetic spectrum.

-
- "**Carrier Sense**" describes the fact that a transmitter uses feedback from a receiver that detects a carrier wave before trying to send. That is, it tries to detect the presence of an encoded signal from another station before attempting to transmit. If a carrier is sensed, the station waits for the transmission in progress to finish before initiating its own transmission.
-
- "**Multiple Access**" describes the fact that multiple stations send and receive on the medium. Transmissions by one node are generally received by all other stations using the medium.



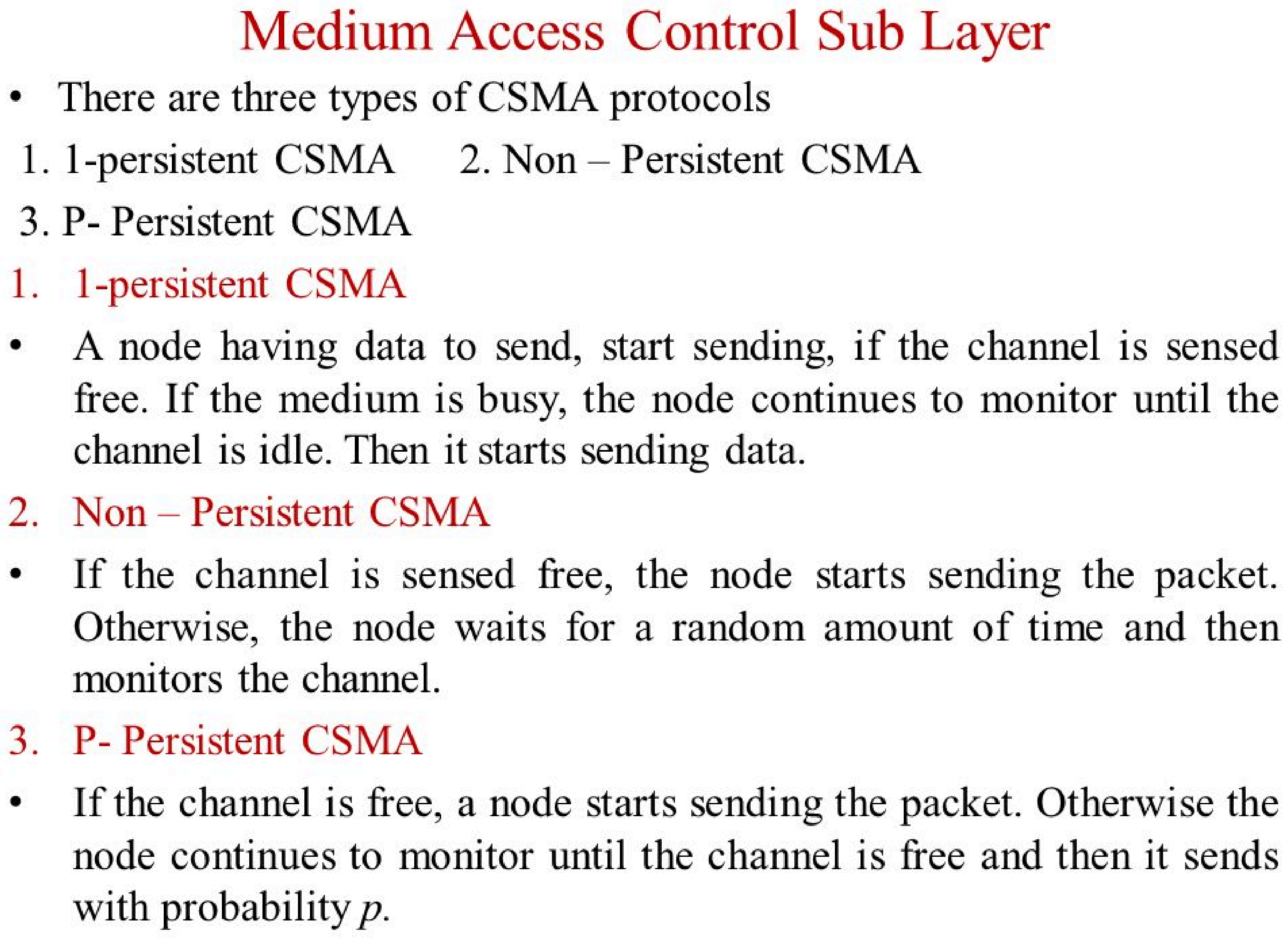
This method was developed to decrease the chances of collisions when two or more stations start sending their signals over the datalink layer. Carrier Sense multiple access requires that each station first check the state of the medium before sending.

Vulnerable Time –

Vulnerable time = Propagation time (Tp)

CSMA (Persistence Methods)

- Persistence methods :- Methods for Sensing the channel (busy/ idle)
- 3 Persistence methods are available:
  1. **I-persistence**
  2. **Non-persistence**
  3. **P-persistence**



Medium Access Control Sub Layer

- There are three types of CSMA protocols

1. 1-persistent CSMA 2. Non – Persistent CSMA

3. P- Persistent CSMA

1. 1-persistent CSMA

- 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.

2. 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.

3. 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*.

1. 1-persistent CSMA :

In 1-persistent CSMA, the station continuously senses the channel to check its state i.e. idle or busy so that it can transfer data or not.

The algorithm of 1-persistent CMSA is:

- When a frame is ready, the transmitting station checks whether the channel is idle or busy.

- If the channel is busy, the station waits and continually checks until the channel becomes idle.
- If the channel is idle then it transmits the frame immediately, with a probability 1.
- A collision may occur if two or more channels transmit simultaneously. If collision occurs, the station waits for a random period of time and restarts the algorithm all over again.

The problem with this method is that there are a large number of chances for the collision it is because there is a chance when two or more stations found channel in idle state and the transmit frames at the same time. On the time when collision occurs the station has to wait for the random time for the channel to be idle and to start all again.

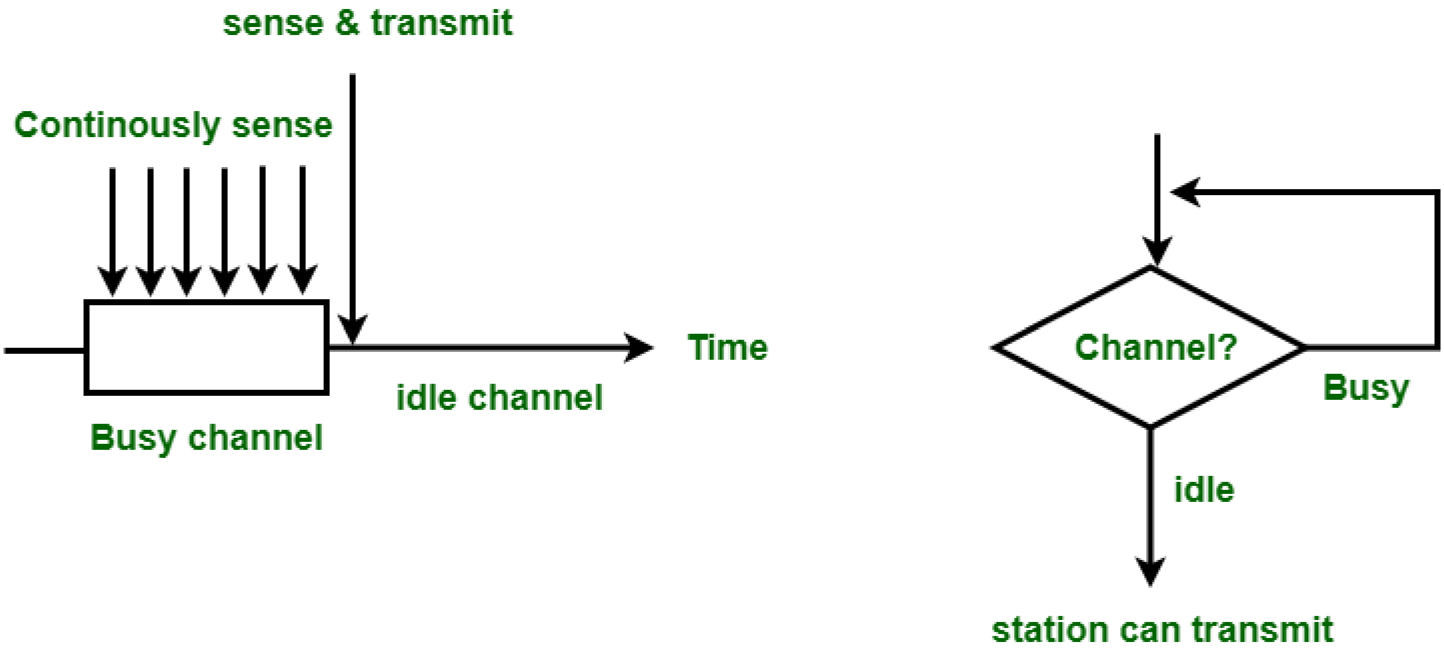


Figure – 1-persistent CSMA

2. p-persistent CSMA :

This is the method that is used when channel has time-slots
Algorithm

The algorithm of p-persistent CMSA is:

- *When a frame is ready, the transmitting station checks whether the channel is idle or busy.*
- *If the channel is idle then it transmits the frame immediately.*
- *If the channel is busy, the station waits and continually checks until the channel becomes idle.*
- *When the channel becomes idle, the station transmits the frame with a probability p.*
- *With a probability (1 – p), the channel waits for next time slot. If the next time slot is idle, it again transmits with a probability p and waits with a probability (1 – p).*
- *The station repeats this process until either frame has been transmitted or another station has begun transmitting.*

- *If another station begins transmitting, the station waits for a random amount of time and restarts the algorithm.*

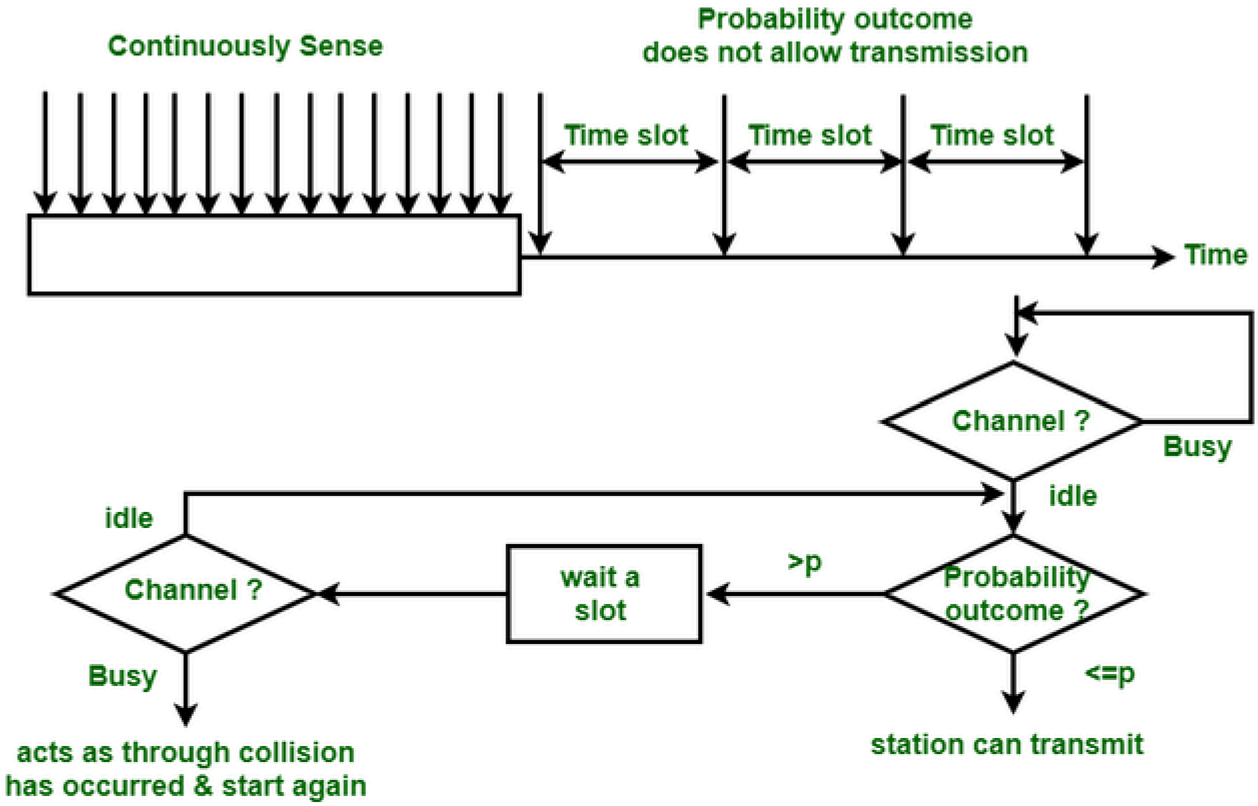


Figure – p-persistent CSMA

3. Non-persistent CSMA :
In this method, the station that has frames to send, only that station senses for the channel. In case of an idle channel, it will send frame immediately to that channel. In case when the channel is found busy, it will wait for the random time and again sense for the state of the station whether idle or busy. In this method, the station does not immediately sense for the channel for only the purpose of capturing it when it detects the end of the previous transmission. The main advantage of using this method is that it reduces the chances of collision. The problem with this is that it reduces the efficiency of the network.

Algorithm

The algorithm of non-persistent CMSA is

- *When a frame is ready, the transmitting station checks whether the channel is idle or busy.*
- *If the channel is idle then it transmits the frame immediately.*
- *If the channel is busy, the station waits for a random time period during which it does not check whether the channel is idle or busy.*
- *At the end of the waiting time period, it again checks the status of the channel and restarts the algorithm.*

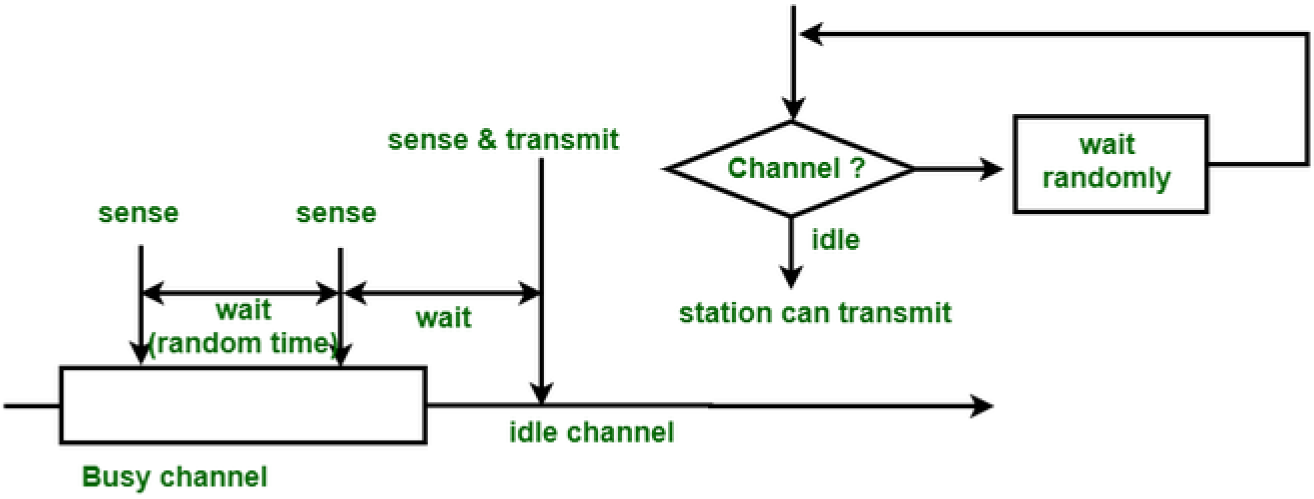


Figure – Non-persistent CSMA

Difference between 1-persistent, p-persistent and Non-persistent CSMA :

Parameter	1-persistent CSMA	p-persistent CSMA	Non-persistent CSMA
Carrier Sense	It sends with the probability of 1 when channel is idle.	It sends with the probability of p when channel is idle.	It send when channel is idle.
Waiting	It continuously senses the channel or carrier.	It waits for the next time slot.	It will wait for the random amount of time to check the carrier.
Chances of Collision	There is highest chances of collision in this.	Less chances as compared to 1-persistence and p-persistence.	Less chances as compared to 1-persistence but more than the p-persistence.

Utilization	It’s utilization is above ALOHA as frames are only sent when the channel is idle.	It’s utilization is depend upon the probability p.	It’s utilization is above 1-persistent as not all the stations constantly check the channel at the same time.
Delay Low Load	It is low as frames are send when the channel become idle.	It is large when p is small as station will not always send when channel is idle.	It is small as station will send whenever channel is found idle but longer than 1-persistent since it checks for the random time when busy.
Delay High Load	It is high due to collision.	It is large when the probability p of sending is small when channel is idle and channel is rarely idle.	It is longer than 1-persistent as channel is checked randomly when busy.

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

In this method, a station monitors the medium after it sends a frame to see if the transmission was successful.If succcessful, the station is finished, if not, the frame is sent again.

Contention Window: If participants determine that the channel is free, they wait a random amount of time before they start sending.

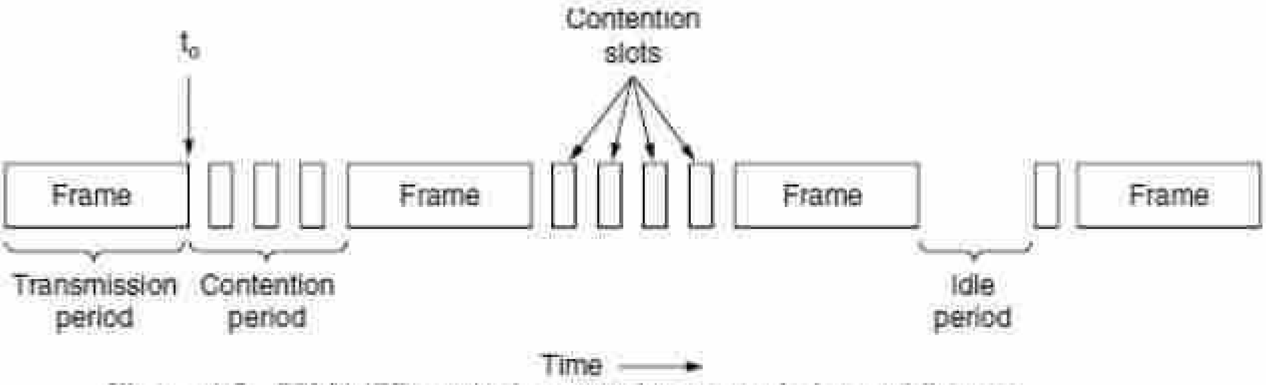


Figure 4-5. CSMA/CD can be in contention, transmission, or idle state.

In CSMA/CD Channel can be in one of the three states: contention, transmission, and idle.

Throughput and Efficiency – The throughput of CSMA/CD is much greater than pure or slotted ALOHA.

JAM SIGNAL

The **jam signal** is a signal that carries a 32-bit binary pattern sent by a data station to inform the other stations that they must not transmit.

ADVANTAGES

More efficient than basic CSMA

DISADVANTAGES

Requires ability to detect collisions

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

The basic idea behind CSMA/CA is that the station should be able to receive while transmitting to detect a collision from different stations. In wired networks, if a collision has occurred then the energy of received signal almost doubles and the station can sense the possibility of collision. In case of wireless networks, most of the energy is used for transmission and the energy of received signal increases by only 5-10% if collision occurs. It can't be used by station to sense collision. Therefore **CSMA/CA has been specially designed for wireless networks**.
These are three type of strategies:

1. **InterFrame Space (IFS) –** When a station finds the channel busy, it waits for a period of time called IFS time. IFS can also be used to define the priority of a station or a frame. Higher the IFS lower is the priority.
2. **Contention Window –** It is the amount of time divided into slots.A station which is ready to send frames chooses random number of slots as **wait time**.
3. **Acknowledgements –** The positive acknowledgements and time-out timer can help guarantee a successful transmission of the frame.

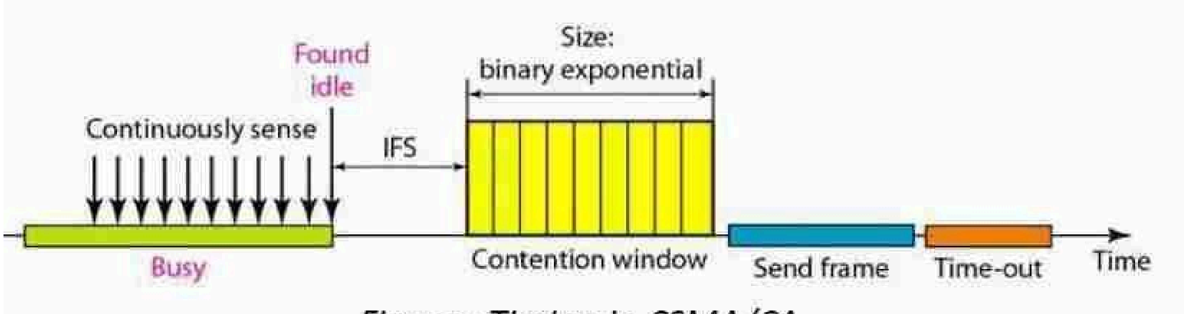


Figure : Timing in CSMA/CA

Let’s see the difference between CSMA/CA and CSMA/CD:-

S.NO	CSMA/CD	CSMA/CA
1.	CSMA / CD is effective after a collision.	Whereas CSMA / CA is effective before a collision.
2.	CSMA / CD is used in wired networks.	Whereas CSMA / CA is commonly used in wireless networks.
3.	It only reduces the recovery time.	Whereas CSMA/ CA minimizes the possibility of collision.
5.	CSMA / CD is used in 802.3 standard.	While CSMA / CA is used in 802.11 standard.
6.	It is more efficient than simple CSMA(Carrier Sense Multiple Access).	While it is similar to simple CSMA(Carrier Sense Multiple Access).

CONTROLLED ACCESS

- In controlled access, the stations consults each other to find which station has right to send.
- Controlled access protocols grants permission to send only one node at a time, to avoid collision of messages on the shared medium.
- A station cannot send data unless it is authorized by the other stations.
- **Now we will discuss three named controlled access methods.**

1. Reservation. Ex: cable modem

2. Polling. Ex: HDLC(normal response mode)

3. Token Passing. Ex: Token Ring, Token Bus.

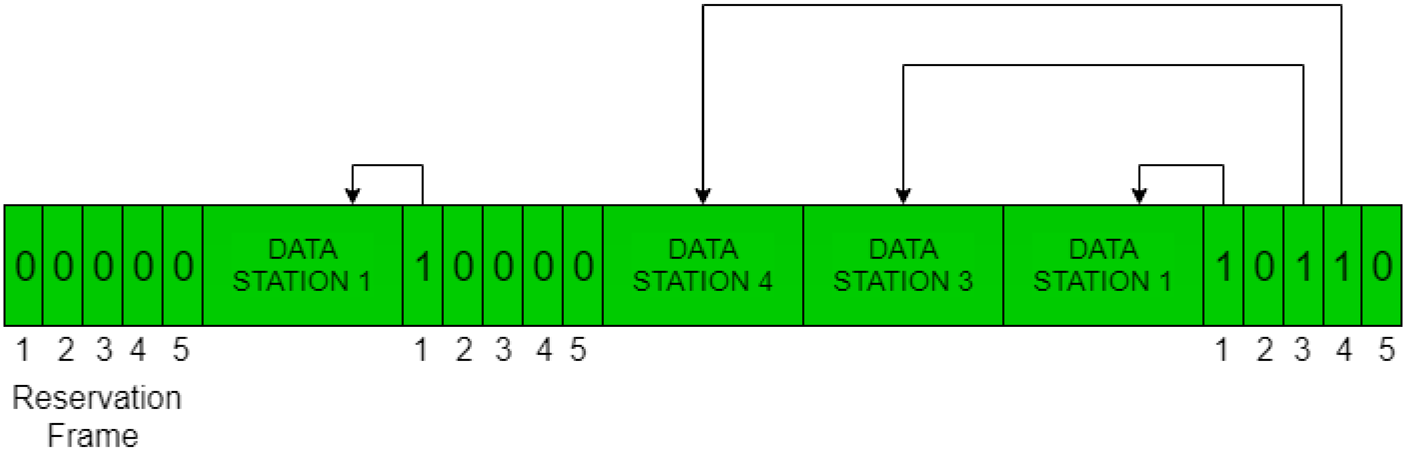
1.RESERVATION

- In the reservation method, a station needs to make a reservation before sending data.
- The time line has two kinds of periods:
  1. Reservation interval of fixed time length
  2. Data transmission period of variable frames.
- If there are M stations, the reservation interval is divided into M slots, and each station has one slot.
- Suppose if station 1 has a frame to send, it transmits 1 bit during the slot 1. No other station is

allowed to transmit during this slot.

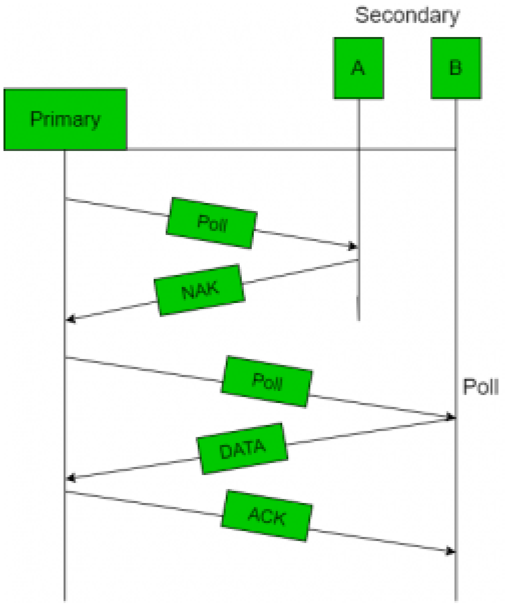
- In general, i th station may announce that it has a frame to send by inserting a 1 bit into i th slot. After all N slots have been checked, each station knows which stations wish to transmit.
- The stations which have reserved their slots transfer their frames in that order.
- After data transmission period, next reservation interval begins.
- Since everyone agrees on who goes next, there will never be any collisions.

The following figure shows a situation with five stations and a five slot reservation frame. In the first interval, only stations 1, 3, and 4 have made reservations. In the second interval, only station 1 has made a reservation.



POLLING

Polling process is similar to the roll-call performed in class. Just like the teacher, a controller sends a message to each node in turn.In this, one acts as a primary station(controller) and the others are secondary stations. All data exchanges must be made through the controller.The message sent by the controller contains the address of the node being selected for granting access.Although all nodes receive the message but the addressed one responds to it and sends data, if any. If there is no data, usually a “poll reject”(NAK) message is sent back.Problems include high overhead of the polling messages and high dependence on the reliability of the controller.



Efficiency

Let T_{poll} be the time for polling and T_t be the time required for transmission of data. Then,

Efficiency = $T_t/(T_t + T_{poll})$

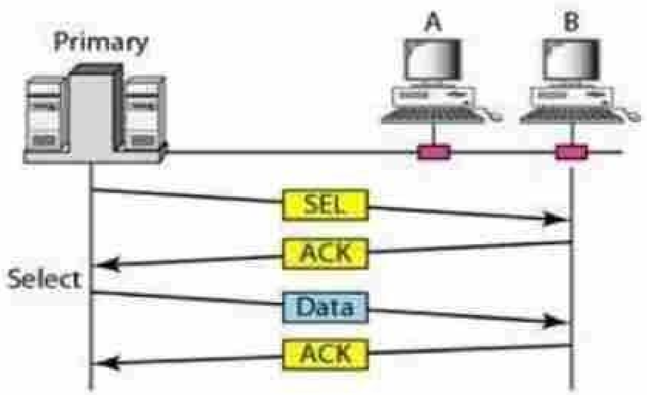
SELECT FUNCTION:

Whenever primary has something to send, it sends the message to each node.

Before Sending the data, it creates and transmits a Select(***SEL***) frame, one field of it includes the address of the intended secondary.

While sending, the primary should know whether the target device is ready to receive or not.

Hence, it alerts the secondary for the upcoming transmission and wait for an acknowledgement (ACK) of secondary's status.



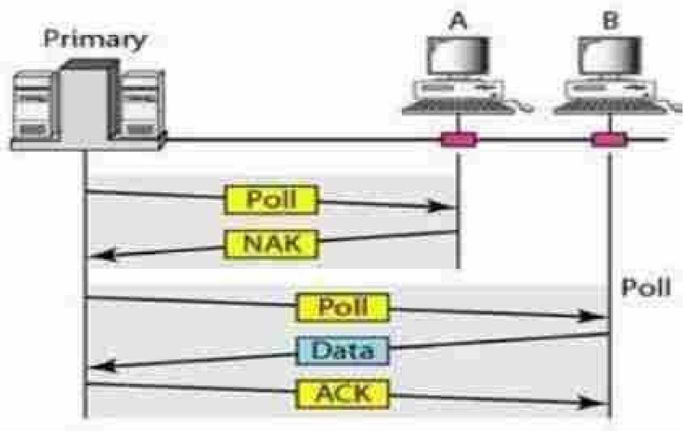
POLL FUNCTION:

When the primary is ready to receive data, it must ask (*poll*) each device if it has anything to send.

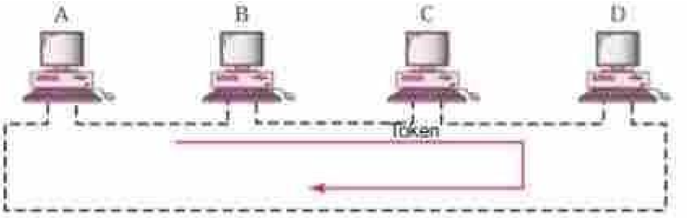
If the secondary has data to transmit, it sends the data frame. Otherwise, it sends a negative acknowledgement(*NAK*) .

The primary then polls the next secondary. When the response is positive (a data frame), the primary reads the frame and returns an acknowledgment (*ACK*).

There are two possibilities to terminate the transmission: either the secondary sends all data, finishing with an *EOT* frame, or the primary says timer is up.

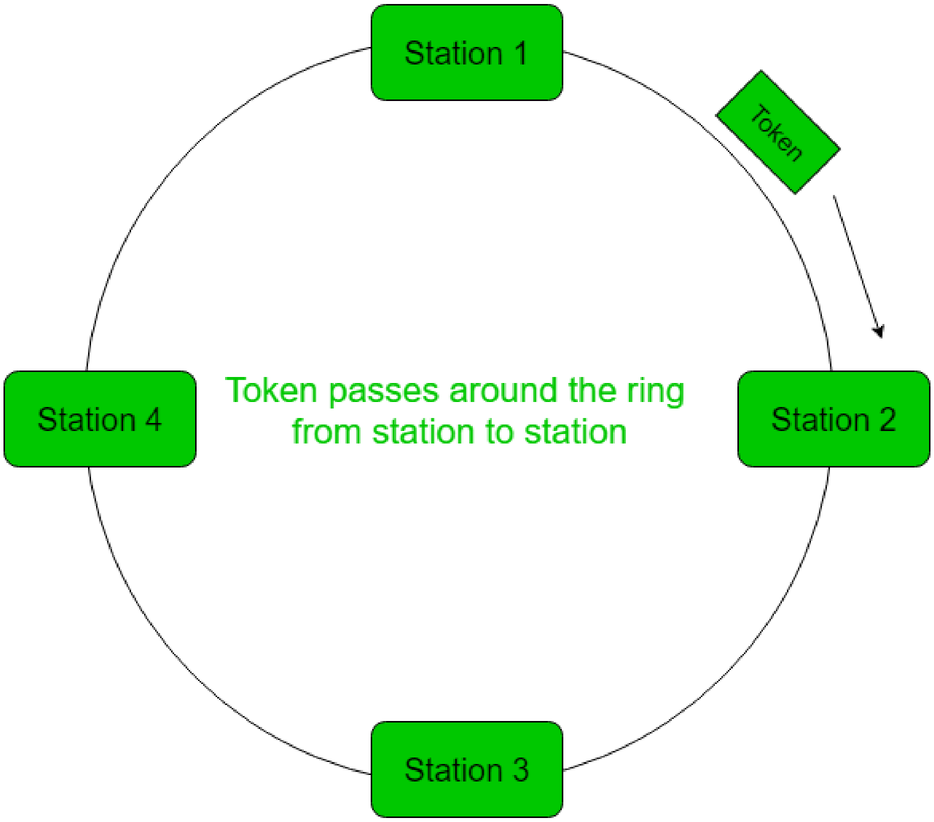


TOKEN PASSING



In token passing scheme, the stations are connected logically to each other in form of ring and access of stations is governed by tokens.A token is a special bit pattern or a small message, which circulate from one station to the next in the some predefined order.In Token ring, token is passed from one station to another adjacent station in the ring whereas incase of Token bus, each station

uses the bus to send the token to the next station in some predefined order.In both cases, token represents permission to send. If a station has a frame queued for transmission when it receives the token, it can send that frame before it passes the token to the next station. If it has no queued frame, it passes the token simply.After sending a frame, each station must wait for all N stations (including itself) to send the token to their neighbors and the other N – 1 stations to send a frame, if they have one.There exists problems like duplication of token or token is lost or insertion of new station, removal of a station, which need be tackled for correct and reliable operation of this scheme.

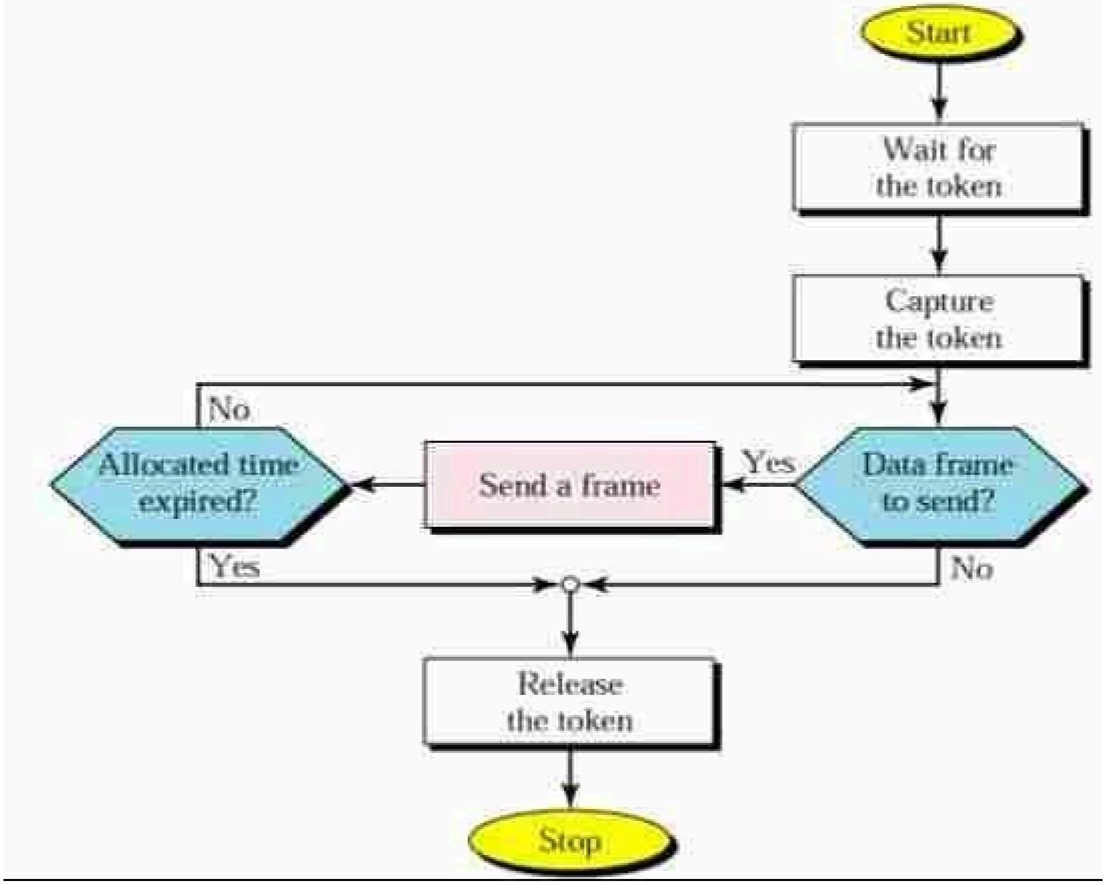


Performance

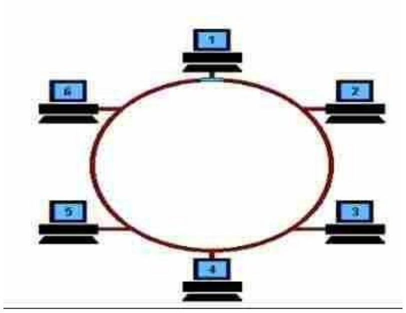
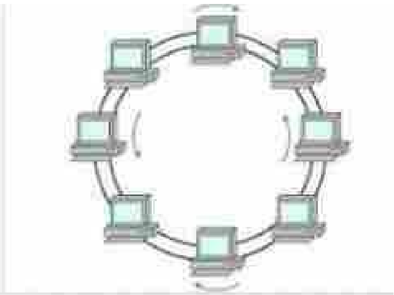
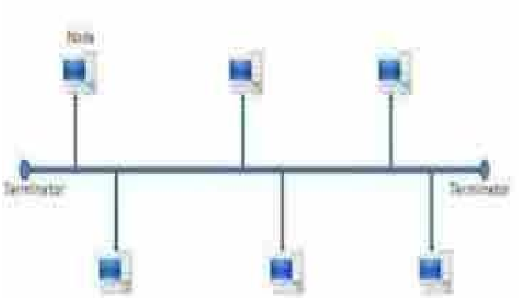
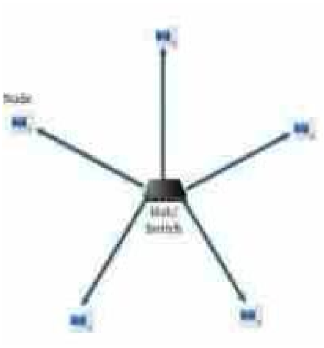
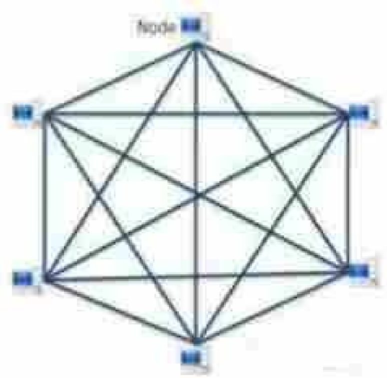
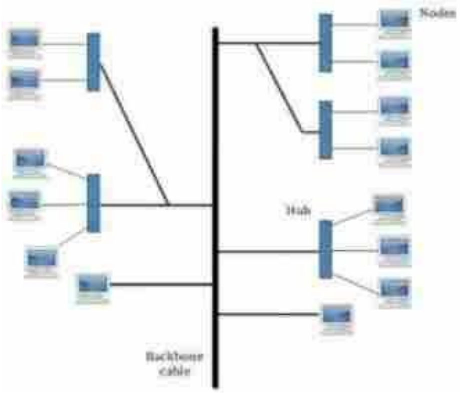
Performance of token ring can be concluded by 2 parameters:-

1. **Delay**, which is a measure of time between when a packet is ready and when it is delivered.So, the average time (delay) required to send a token to the next station.
2. **Throughput**, which is a measure of the successful traffic.

TOKEN PASSING FLOW CHART :



TYPES OF LOGICAL RINGS :

Physical Ring Topology  	**DUAL RING TOPOLOGY :** 
BUS RING TOPOLOGY :  	STAR TOPOLOGY : 
MESH TOPOLOGY  	TREE TOPOLOGY 

3. Channelization:

In this, the available bandwidth of the link is shared in time, frequency and code to multiple stations to access the channel simultaneously.

1. Frequency Division Multiple Access (FDMA) :

FDMA is a type of channelization protocol. In this bandwidth is divided into various frequency bands. Each station is allocated with band to send data and that band is reserved for particular station for all the time which is as follows :

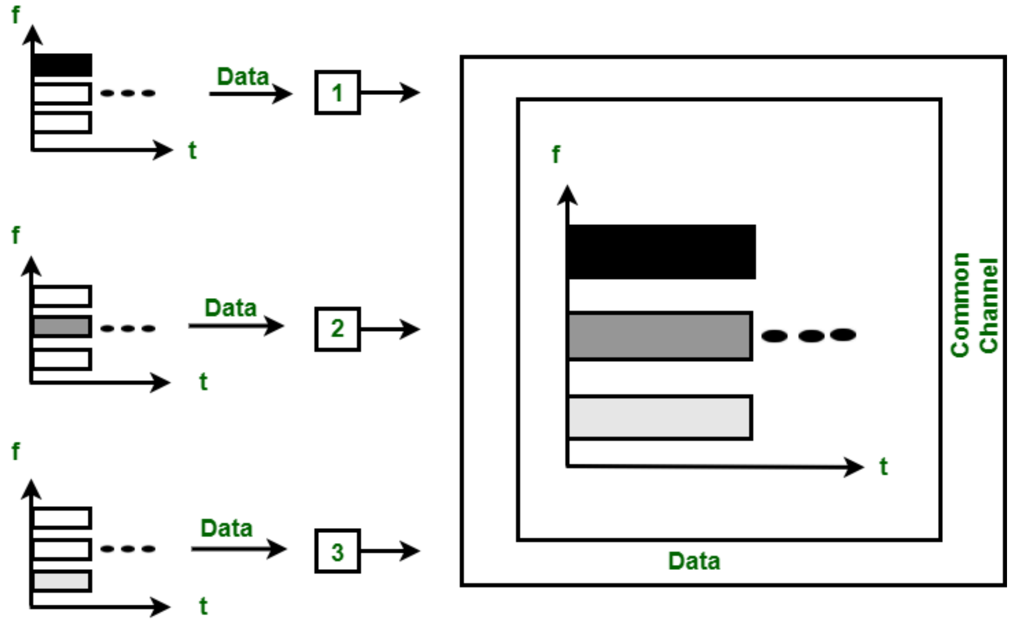


Figure – FDMA

The frequency bands of different stations are separated by small band of unused frequency and that unused frequency bands are called as guard bands that prevents the interference of stations. It is like access method in data link layer in which data link layer at each station tells its physical layer to make a band pass signal from the data passed to it. The signal is created in the allocated band and there is no physical multiplexer at the physical layer.

2. Time Division Multiple Access (TDMA) :
TDMA is the channelization protocol in which bandwidth of channel is divided into various stations on the time basis. There is a time slot given to each station, the station can transmit data during that time slot only which is as follows :

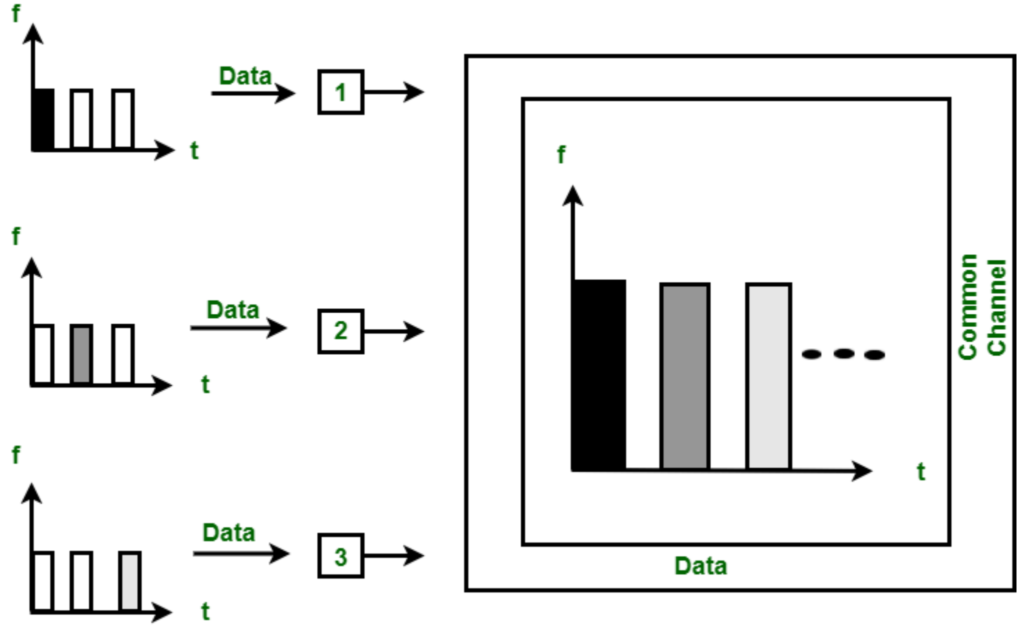
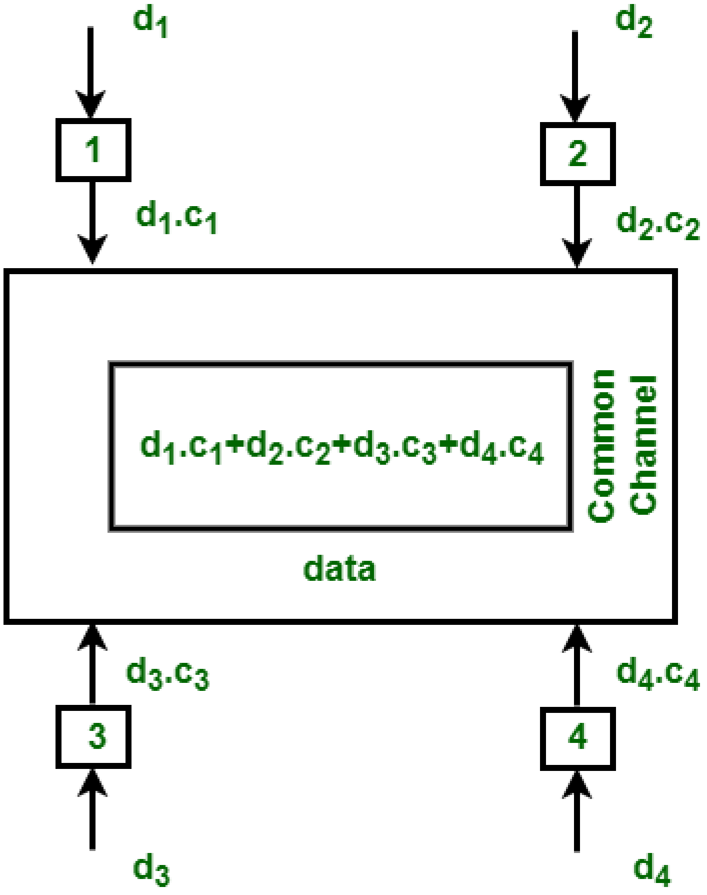


Figure – TDMA

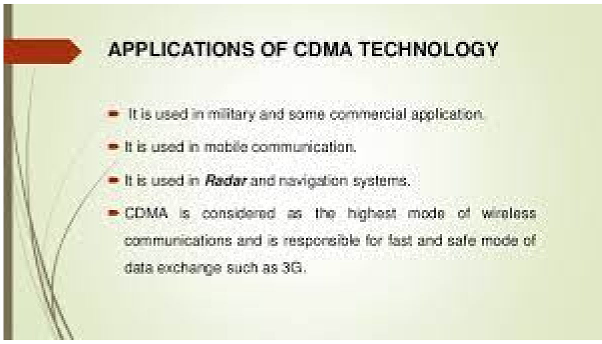
Each station must aware of its beginning of time slot and the location of the time slot. TDMA requires synchronization between different stations. It is type of access method in the data link layer. At each station data link layer tells the station to use the allocated time slot.

3. Code Division Multiple Access (CDMA) :

In CDMA, all the stations can transmit data simultaneously. It allows each station to transmit data over the entire frequency all the time. Multiple simultaneous transmissions are separated by unique code sequence. Each user is assigned with a unique code sequence. Eg: wash code



In the above figure, there are 4 stations marked as 1, 2, 3 and 4. Data assigned with respective stations as d_1, d_2, d_3 and d_4 and the code assigned with respective stations as c_1, c_2, c_3 and c_4.



APPLICATIONS OF CDMA TECHNOLOGY

- It is used in military and some commercial application.
- It is used in mobile communication.
- It is used in ***Radar*** and navigation systems.
- CDMA is considered as the highest mode of wireless communications and is responsible for fast and safe mode of data exchange such as 3G.

3.7. Wired LAN: Ethernet Standards and FDDI

Ethernet Standards

- The Ethernet standards come under the IEEE 802 section which deal with **local area networks** and **metropolitan area networks.** In particular, **IEEE 802.3 defines Ethernet.**
- The different IEEE 802.3 standards define different aspects of Ethernet covering the physical layer and data link layer's media access control (MAC) of ***wired Ethernet.***
- Some of the individual standards may introduce new versions or flavours of Ethernet to keep pace with the growing requirements for speed and performance, whereas other standards may define aspects like the data frames used.

The different standards with their numbers are outlined in the table below:

Standard Ethernet Code

In order to understand standard Ethernet code, one must understand what each digit means. Following is a guide:

Guide to Ethernet Coding

10	at the beginning means the network operates at 10Mbps.
BASE	means the type of signaling used is baseband.
2 or 5	at the end indicates the maximum cable length in meters.
T	the end stands for twisted-pair cable.
X	at the end stands for full duplex-capable cable.
FL	at the end stands for fiber optic cable.

For example: *100BASE-TX indicates a Fast Ethernet connection (100 Mbps) that uses a twisted pair cable capable of full-duplex transmissions.*

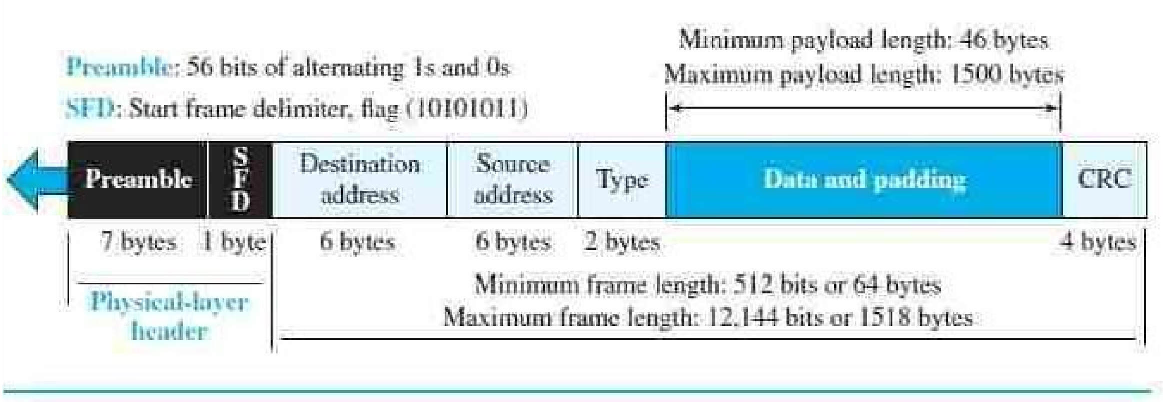
Some of Ethernet version numbering:

10BASE5: 10 Mb/s over coaxial cable (ThickWire)
10BROAD36: 10 Mb/s over broadband cable, 3600 m max segments
10BASE5: 1 Mb/s over 2 pairs of UTP
10BASE2: 10 Mb/s over thin RG58 coaxial cable (ThinWire), 185 m max segments
10BASE-T: 10 Mb/s over 2 pairs of UTP
10BASE-FL: 10 Mb/s fiber optic point-to-point link

10BASE-FB: 10 Mb/s fiber optic backbone (between repeaters).

Basic frame format which is required for all MAC implementation is defined in **IEEE 802.3 standard**. Though several optional formats are being used to extend the protocol’s basic capability.

Figure 13.3 *Ethernet frame*


Preamble: 56 bits of alternating 1s and 0s
SFD: Start frame delimiter, flag (10101011)
Minimum payload length: 46 bytes
Maximum payload length: 1500 bytes
Preamble
SFD
Destination address
Source address
Type
Data and padding
CRC
7 bytes
1 byte
6 bytes
6 bytes
2 bytes
4 bytes
Physical-layer header
Minimum frame length: 512 bits or 64 bytes
Maximum frame length: 12,144 bits or 1518 bytes

PREAMBLE – Ethernet frame starts (PRE).

Start frame delimiter (SFD). This field (1 byte: 10101011) signals the beginning

of the frame.

Type. This field defines the upper-layer protocol whose packet is encapsulated in

the frame.

Data. This field carries data encapsulated from the upper-layer protocols. For example, a datagram has a field that defines the
length(padding) of the data.

Cyclic Redundancy Check (CRC): The last field contains error detection information

Fiber Distributed Data Interface (FDDI) is a set of ANSI and ISO standards for transmission of data in local area network (LAN) over fiber optic cables. It is applicable in large LANs that can extend up to 200 kilometers in diameter.

Features

FDDI uses optical fiber as its physical medium.

It operates in the physical and medium access control (MAC layer) of the Open Systems Interconnection (OSI) network model.

It provides high data rate of 100 Mbps and can support thousands of users.

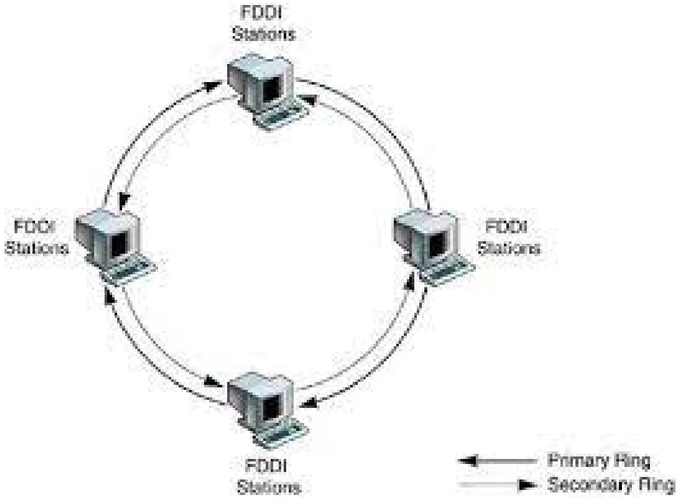
It is used in LANs up to 200 kilometers for long distance voice and multimedia communication.

It uses ring based token passing mechanism and is derived from IEEE 802.4 token bus standard.

It contains two token rings, a primary ring for data and token transmission and a secondary ring that provides backup if the primary ring fails.

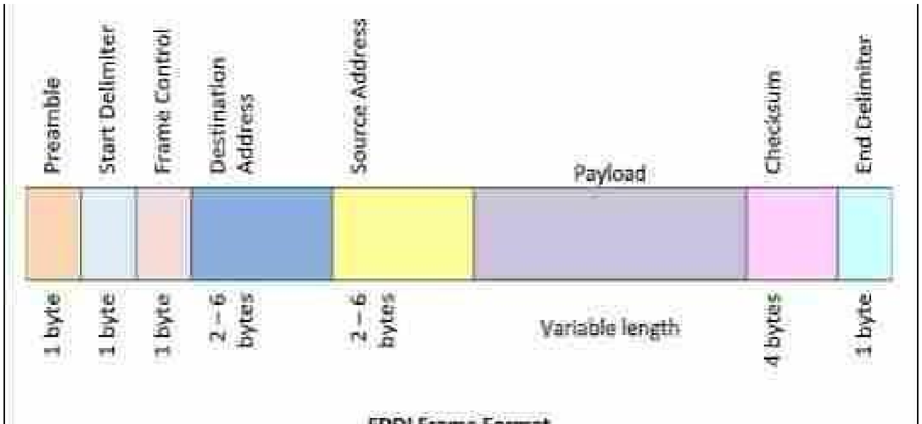
FDDI technology can also be used as a backbone for a wide area network (WAN).

The following diagram shows FDDI −



Frame Format

The frame format of FDDI is similar to that of token bus as shown in the following diagram −



FDDI Frame Format

The fields of an FDDI frame are −

Preamble: 1 byte for synchronization of receiver.

Start Delimiter: 1 byte that marks the beginning of the frame.

Frame Control: 1 byte that specifies whether this is a data frame or control frame.

Destination Address: 2-6 bytes that specifies address of destination station.

Source Address: 2-6 bytes that specifies address of source station.

Payload: A variable length field that carries the data from the network layer.

Checksum: 4 bytes frame check sequence for error detection.
End Delimiter: 1 byte that marks the end of the frame.

3.8. Wireless LAN : IEEE 802.11x and Bluetooth Standards

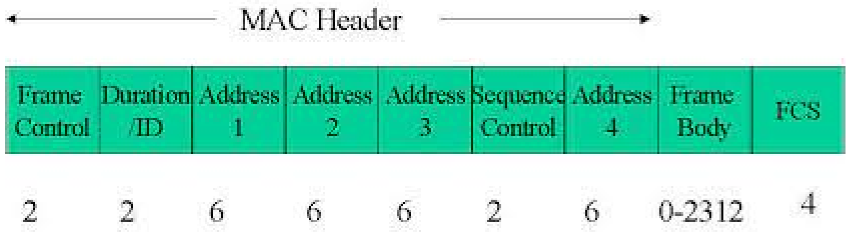
802.11x is generic term to refer to the IEEE 802.11 standard for defining communication over a wireless LAN (WLAN). 802.11, commonly known **as Wi -Fi**, specifies an over-the-air interface between a wireless client and a base station or between two wireless clients

- It refers to the common flavors of Wi-Fi, most notably 802.11a, 802.11b, 802.11g, and 802.11n.

Frame Format of 802.11

The MAC layer frame consists of nine fields.

1. **Frame Control (FC).** This is a 2 byte field and defines the type of frame and some control information.



← MAC Header →

Frame Control	Duration /ID	Address 1	Address 2	Address 3	Sequence Control	Address 4	Frame Body	FCS
2	2	6	6	6	2	6	0-2312	4

2. **D .** It stands for **duration** and is of 2 bytes. This field defines the duration for which the frame and its acknowledgement will occupy the channel.

3. **Addresses.** There are 4 address fields of 6 bytes length. These four addresses represent source, destination, source base station and destination base station.

4. **Sequence Control (SC).** This 2 byte field defines the sequence number of frames to be used in flow control.

5. **Frame body**. This field can be between 0 and 2312 bytes. It contains the information.

6. **Frame Check Sequence (FCS)**. This field is 4 bytes long and contains an error detection sequence.

BLUETOOTH

- IEEE 802.15
- It is a wireless LAN technology using short-range radio links, intended to replace the cable(s) connecting portable and/or fixed electronic devices.
- It is a network where devices can automatically find each other, establish connections, and discover what they can do for each other.
- range 10-100 mtrs.
- features are robustness, low complexity, low power and low cost.
- uses a 2.4-GHz ISM band divided into 79 channels of 1 MHz each
- A Bluetooth device has a built-in short-range radio transmitter.
- It uses Frequency Hop Spread Spectrum (FHSS) to avoid any interference.

Symbol of Bluetooth

An example of a Bluetooth device

The usage of Bluetooth has widely increased for its special features.

- Bluetooth offers a uniform structure for a wide range of devices to connect and communicate with each other.
- Bluetooth technology has achieved global acceptance such that any Bluetooth enabled device, almost everywhere in the world, can be connected with Bluetooth enabled devices.
- Low power consumption of Bluetooth technology and an offered range of up to ten meters has paved the way for several usage models.
- Bluetooth offers interactive conferences by establishing a network of laptops.
- Bluetooth usage model includes cordless computer, intercom, cordless phone and mobile phones.

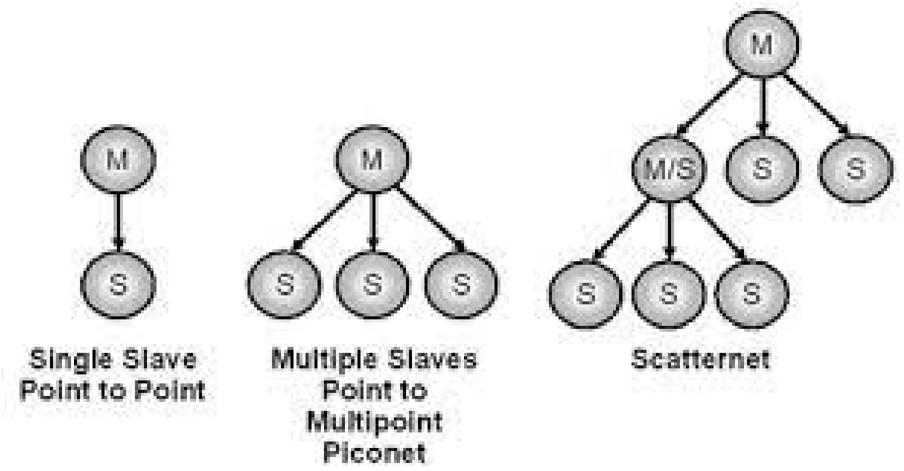
Types of Bluetooth Wireless Technology

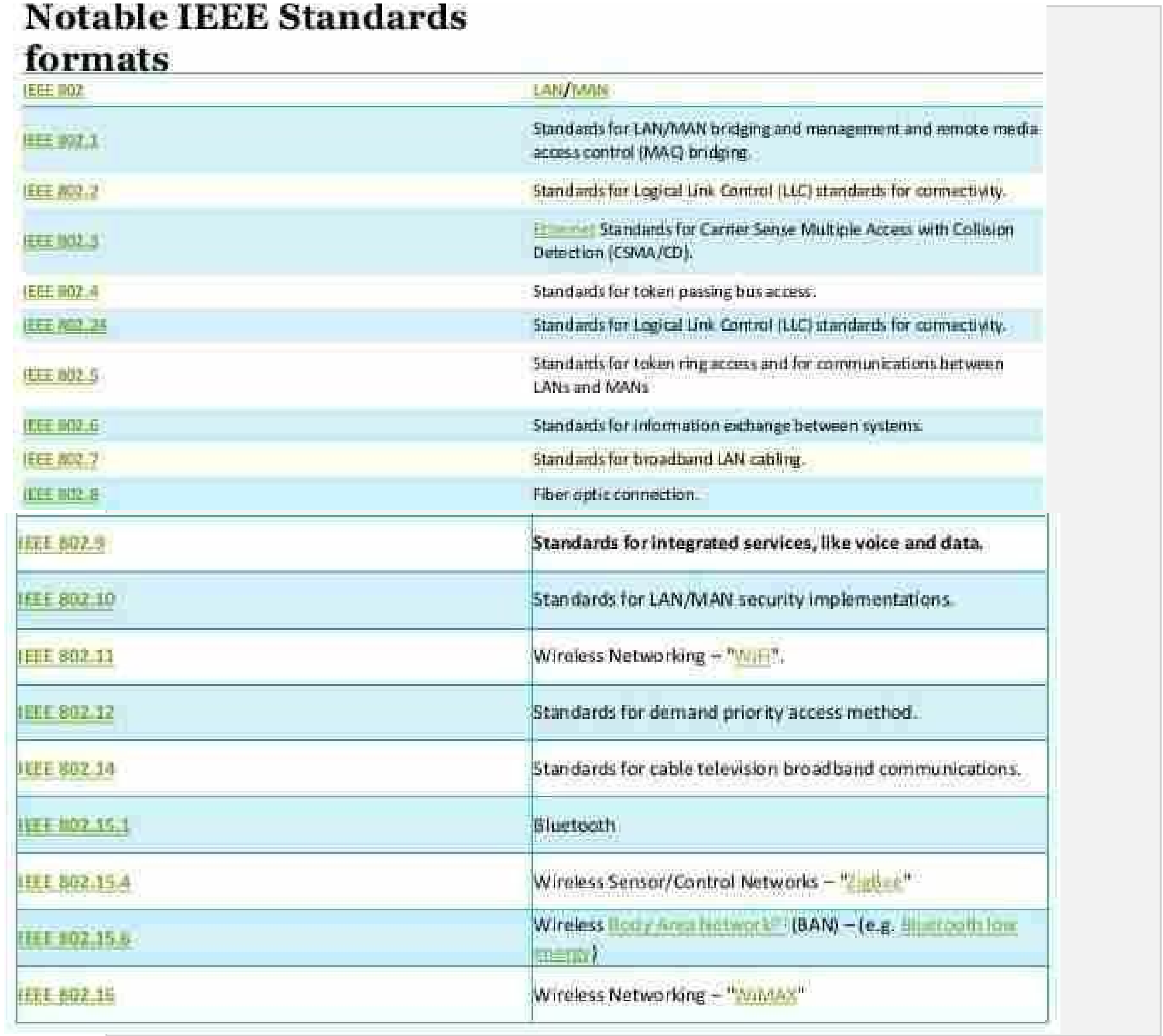
🎬 Depending on the power consumption and range of the device, there are 3 Bluetooth Classes as:

1. Class 1: Max Power – 100mW ; Range – 100 m
2. Class : Max Power – 2.5mW ; Range – 10 m
3. Class : Max Power – 1mW ; Range – 1 m

Bluetooth defines two types of network topology:

Piconet	*Scatternet*
In this bluetooth network, the device can function either as master or slave.	***In this bluetooth network, device can function as master or slave or (master+slave)***
It serves a smaller coverage area.	***It serves a larger coverage area.***
It supports a maximum 8 nodes.	***It supports more than 8 nodes.***
It allows less efficient use of available bluetooth channel bandwidth.	***It allows more efficient use of available bluetooth channel bandwidth.***





Notable IEEE Standards formats

IEEE 802	LAN/MAN
IEEE 802.1	Standards for LAN/MAN bridging and management and remote media access control (MAC) bridging.
IEEE 802.2	Standards for Logical Link Control (LLC) standards for connectivity.
IEEE 802.3	Ethernet Standards for Carrier Sense Multiple Access with Collision Detection (CSMA/CD).
IEEE 802.4	Standards for token passing bus access.
IEEE 802.2A	Standards for Logical Link Control (LLC) standards for connectivity.
IEEE 802.5	Standards for token ring access and for communications between LANs and MANs
IEEE 802.6	Standards for information exchange between systems.
IEEE 802.7	Standards for broadband LAN cabling.
IEEE 802.8	Fiber optic connection.
IEEE 802.9	**Standards for integrated services, like voice and data.**
IEEE 802.10	Standards for LAN/MAN security implementations.
IEEE 802.11	Wireless Networking – "WiFi".
IEEE 802.12	Standards for demand priority access method.
IEEE 802.14	Standards for cable television broadband communications.
IEEE 802.15.1	Bluetooth
IEEE 802.15.4	Wireless Sensor/Control Networks – "ZigBee"
IEEE 802.15.6	Wireless Body Area Network" (BAN) – (e.g. Bluetooth low energy)
IEEE 802.16	Wireless Networking – "WiMAX"

3.9	**Token Bus, Token Ring and Virtual LAN**		**0.5**

802.4 Token Bus

- The 802.4 IEEE standard defines the Token Bus protocol for a token-passing access method on a bus topology.
- In a token-passing access method, a special packet called a token is passed from station to station and only the token holder is permitted to transmit packets onto the LAN.
- No collisions can occur with this protocol (Only One Station can transfer)
- When a station is done transmitting its packets, it passes the token to the "next" station.
- The next station does not need to be physically closest to this one on the bus, just the next logical station.

- A station can hold the token for only a certain amount of time before it must pass it on -even if it has not completed transmitting all of its data.

This assures access to all stations on the bus within a specified period of time.

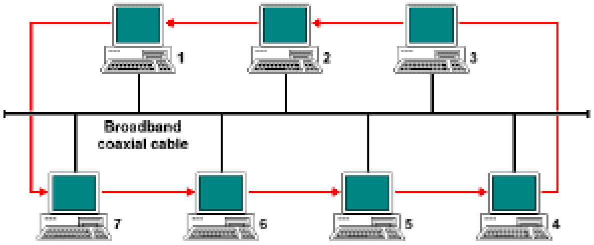


Figure : Token Bus Network (Red Arrow Indicates Token Passing Sequence)

802.5 Token Ring

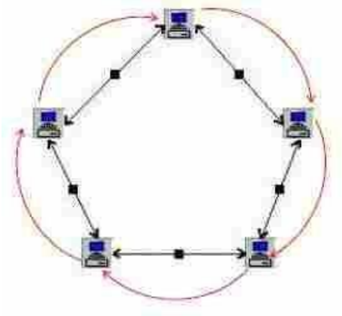


Figure : Token Bus Network (Red Arrow Indicates Token Passing Sequence)

- The 802.5 IEEE standard defines the Token Ring protocol which, like Token Bus, is another token- passing access method, but for a ring topology
- A ring topology consists of a series of individual point-to-point links that form a circle
- A token is passed from station to station in one direction around the ring, and only the station holding the token can transmit packets onto the ring
- Data packets travel in only one direction around the ring
- When a station receives a packet addressed to it, it copies the packet and puts it back on the ring
- When the originating station receives the packet, it removes the packet.

Virtual LANs

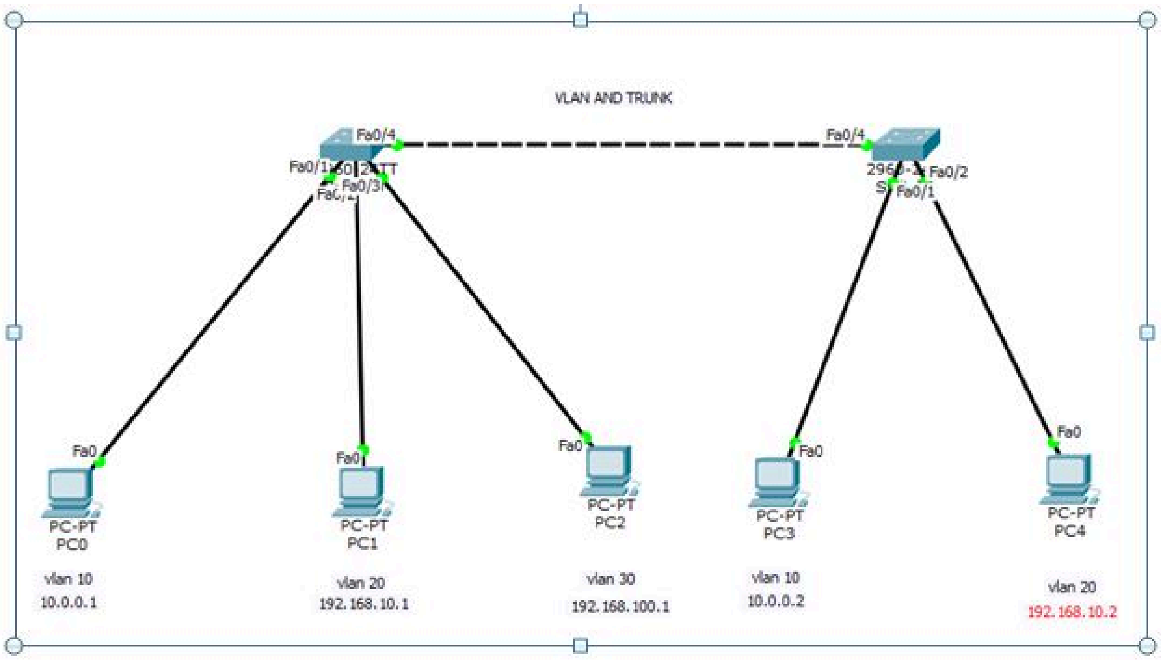
- A virtual local area network (**VLAN**) is a logical group of workstations, servers and network devices that appear **to be** on the same LAN despite their geographical distribution.

- VLANs make it easy for network administrators to partition a single switched network to match the functional and security requirements of their systems.

- VLANs are often set up by larger businesses to re-partition devices for better traffic management.

- VLANs are also important because they can help improve the overall performance of a network by grouping together devices that communicate most frequently. VLANs also provide security on larger networks by allowing a higher degree of control over which devices have access to each other.

- If any of the end stations are in different buildings (not the same physical LAN segment), you can then group them into a VLAN.



Types:

Types of VLANs

Types of VLANs include Protocol based, static and dynamic VLANs.

- Static VLAN- also referred to as port-based VLAN, needs a network administrator to assign the ports on a network switch to a virtual network; while:
- Dynamic VLAN- allows a network administrator just to define network membership based on device characteristics, as opposed to switch port location.