## UNIT-II

## DATA LINK LAYER

The data link layer in the OSI (Open System Interconnections) Model is in between the physical layer and the network layer. This layer converts the raw transmission facility provided by the physical layer to a reliable and error-free link.

### Design issues of Data Link Layer

1. Service Provided to Network Layer
2. Framing
3. Error Control
4. Flow Control

### 1. Service Provided to Network Layer

The types of services provided can be of three types

- Unacknowledged connectionless service
- Acknowledged connectionless service
- Acknowledged connection - oriented service

### Unacknowledged connectionless service

- It consists of having the source machine send independent frames to destination machine without having the destination machine acknowledge them.

  Eg: Ethernet
- The service rate is used when the error rate is low.

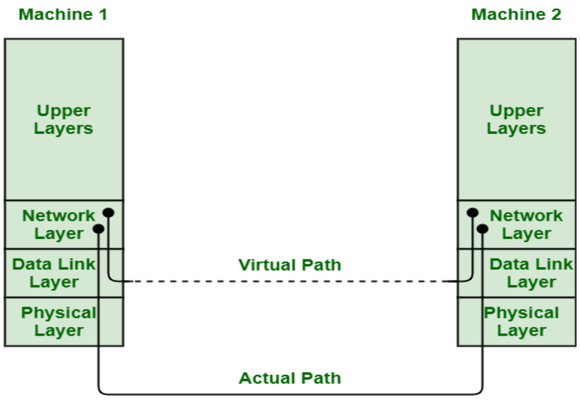
### Acknowledged connectionless service

- There is no logical connections b/w the sender and receiver.
- Each and every frame sent is individually acknowledged
- This service simply provides acknowledged connectionless service i.e. packet delivery is simply acknowledged, with help of stop and wait for protocol.

## Acknowledged connection - oriented service

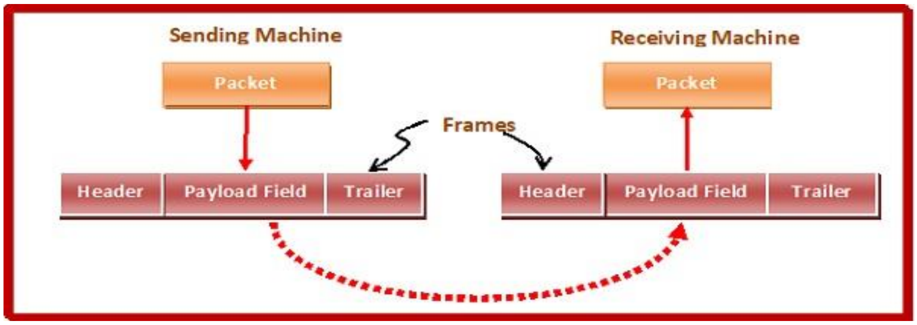
- In this type of service, connection is established first among sender and receiver or source and destination before data is transferred.
- It is guarantees that are frame is received exactly once.

## This process is shown in diagram



## Framing

- The Data Link Layer should detect and correct the errors
- In this purpose, DLL will break up the bit stream into discrete frames, compute a small token called a checksum each frame and include the checksum in the frame when it is transmitted.
- When the frame arrives at the destination, the checksum is recomputed.
- After dividing the data into frames, we should be able to identify the starting & and ending of each frame.



There are two types of framing

1. **Fixed Length**: The frame is of fixed size and there is no need to provide boundaries to the frame, the length of the frame itself acts as a delimiter.

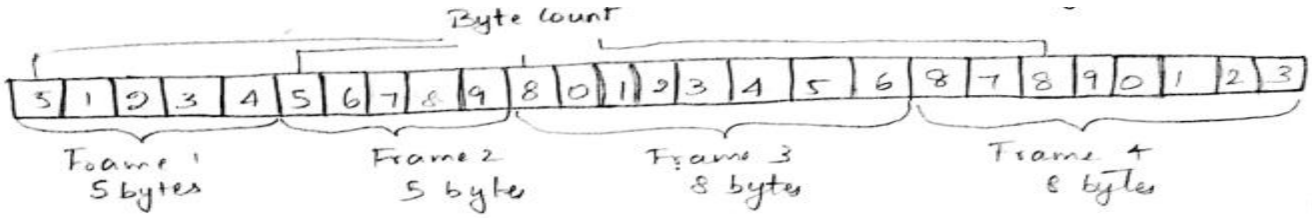
2. **Variable Length**: In this, there is a need to define the end of the frame as well as the beginning of the next frame to distinguish

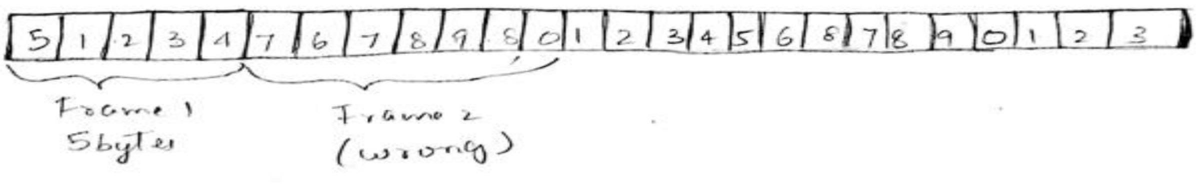
**There are four methods in the framing**

- Character Set
- Flag Byte or Character Stuffing or Byte Stuffing
- Bit Stuffing
- Violation of Physical layer

**Character Set**

- This is a method uses a field in the header to specify the number of bytes in the frame.
- When the DLL at the destination sees the byte count, it knows how many bytes follow & hence where the end of the frame.
- This problem occurs if the byte count is changed by any transmission error.
- If the byte count of 5 becomes 7due to error in this method is used rarely.

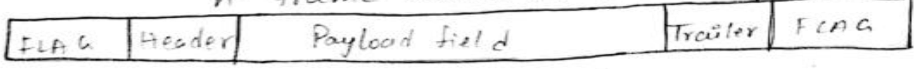


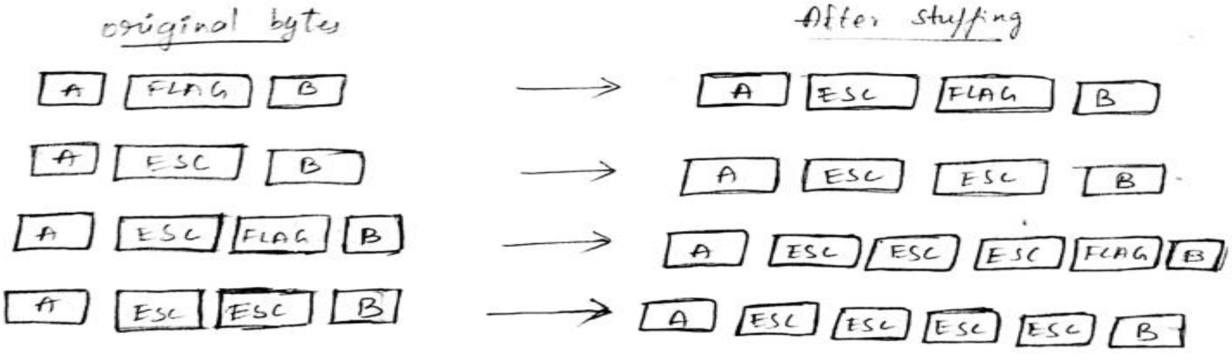


## Flag Byte or Character Stuffing or Byte Stuffing

- In this method special byte called flag byte is used as both the starting & ending delimiter of each frame.
- Two consequent flag bytes indicate end of the frame and start the next frame.
- If the receiver looses the synchronization, it can search for two flag bytes to find the end of the frame.
- One way to solve this problem is to insert a special byte called (ESC) just before each flag byte in the data.
- The DLL on the receiving end removes the escape bytes before giving the data to the network layer.
- This technique is called as Byte Stuffing.

A frame delimiter with flag bytes

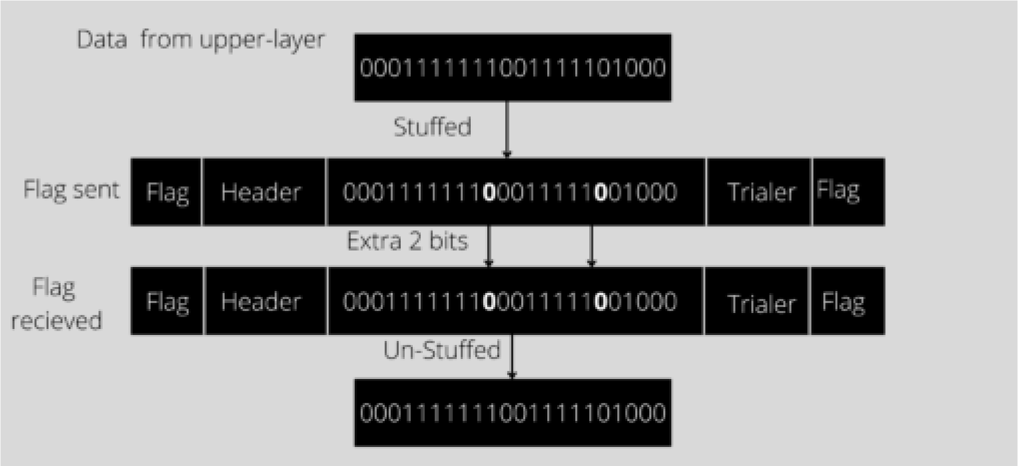




Four examples of byte sequences before & after byte stuffing

## Bit Stuffing

- Most protocols used a special 8bit pattern flag 01111110 as the delimiter the beginning and ending of the frame. The bit stuffing is done at the sender end and bit removal at the receiver end.
- If u has 0 after five consequent 1s we still stuff a 0.The receiver remove the 0.



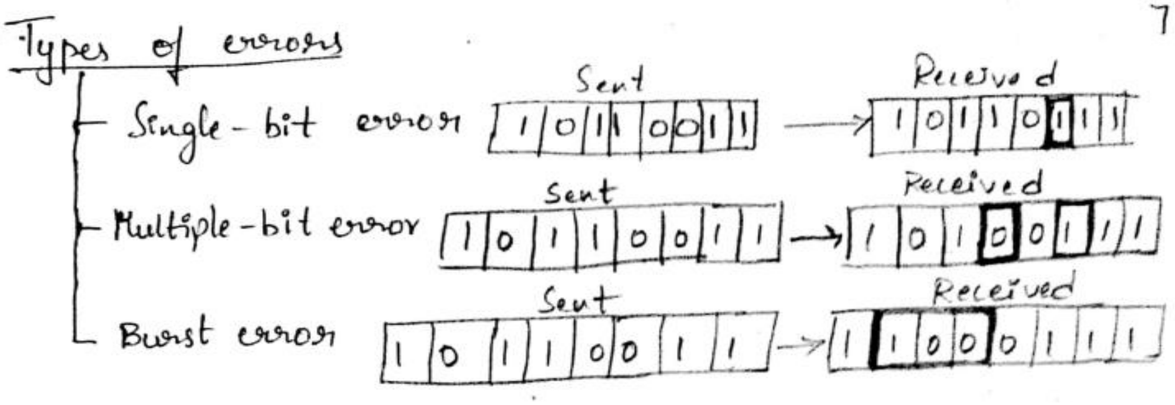
Bit Stuffing

**Violation of Physical layer**

- In order to operate a division between frames in Data Link Layer this approach exploits the redundancy in Physical Layer Encoding that represents data as 00 error, 01 low, 10 high, 11 error

**Error Control**

- To ensure reliable delivery, the sender should be provided with some feedback about what is happening at receiver.
- In this purpose ,receiver sends special control frames having positive or negative acknowledgement
- If sender receives positive acknowledgment, it means that the frame has transmitted safely.
- If sender receives negative acknowledgment, it means that the frame is lost and the sender must retransmit the frame.
- To overcome this timer are used in DLL
- When sender transmits 0 frames, it also starts a timer.
- The timer is set to the time interval required for the data to reach the destination and ACK to reach the source.
- If the timer expire ,it means that either the frame is lost or ACK is lost



**Flow control**

- Flow is controlled by sending the data according to the capability of the receiver.

There are two types

1. **Feedback based flow control**: The receiver sends some feedback to the sender .This feedback includes:-

- When to send the data
- How much data the sender can transmit
- At what rate data can be transmitted

2. **Rate based flow control:** There is built in mechanism that limits the rate at which senders can transmit data. Without using feedback from receiver.

## Error detection and Corrections

- Error detections codes are used when the error rate high.

**Error detection codes**

1. Parity 2.Checksum 3.CRC (Cyclic Redundancy check)

1. **Parity**: It can detect single bit error.

There are two types: 1.Even parity 2. Odd parity

| Dataword<br>Original data | even parity | Codeword<br>Transmitted data | odd parity | Codeword<br>transmitted data |
|---|---|---|---|---|
| 1011010 | 0 | 10110100 | 1 | 10110101 |
| 100101 | 1 | 1001011 | 0 | 1001010 |

If transmitted data is | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | → parity bit

~~B~~ At the receiver,

If the received data is | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | → parity bit

Receiver calculates the parity bit

⇒ parity bit = 1 [Consider even parity]

the transmitted parity bit & receiver calculated parity bit are not equal. So, error is occured.

**2. Checksum or 1's Representation Method**

**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-

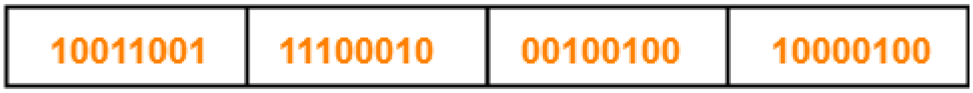
10011001111000100010010010000100

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.

3. **CRC (Cyclic Redundancy Check)**

- Cyclic Redundancy Check (CRC) is an error detection method.
- It is based on binary division.
- CRC generator is an algebraic polynomial represented as a bit pattern.
- Bit pattern is obtained from the CRC generator using the following rule is The power of each term gives the position of the bit and the coefficient gives the value of the bit.
- The polynomial code bit strings are representation of polynomials with coefficients of 0 and 1 only.
- When the polynomial code is employed, the sender and receiver must agree upon a generator polynomial G(x).

- The result is 0 check summed frame to be transmitted T(X).

$$1x^7 + 1x^6 + 0x^5 + 1x^4 + 1x^3 + 0x^2 + 1x^1 + 1x^0$$

$$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow$$

$$1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1$$

Here the Binary pattern is 11011011

**CRC Generator**

If CRC generator or divisor is n bits the CRC bits are (n-1)

**Formula for CRC bits =data+ (n-1)/CRC Generator**

Ex:- Frame M(x) = 1101011111

Generator G(x) = 10011

code to be appended at the end of M(x)

= [(No of bits in G(x)) −1] = 5-1 = 4 (0000)

Sender side

```
10011 ) 11010111110000 ( 110000111
        10011
        -----
         10011
         10011
         -----
          00001
          00000
          -----
           00011
           00000
           -----
            00111
            00000
            -----
             01111
             00000
             -----
              11110
              10011
              -----
               11010
               10011
               -----
                10010
                10011
                -----
                 00010
                 00000
                 -----
                  0010
```

Transmitted frame T(x) =

11010111110010

Receiver side :

```
10011 ) 11010111110010 ( 1100001110
        10011
        -----
         10011
         10011
         -----
          00001
          00000
          -----
           00011
           00000
           -----
            00111
            00000
            -----
             01111
             00000
             -----
              11110
              10011
              -----
               11010
               10011
               -----
                10011
                10011
                -----
                 00000
                 00000
                 -----
                  (0)
```

At the receiver, the above calculation is done. If the remainder is '0'. It means there is no error.

## Error Detection and Correction Method-Hamming Code

Hamming code is a block code that is capable of detecting up to two simultaneous bit errors and correcting single-bit errors. It was developed by **R.W. Hamming** for error correction.

In this coding method, the source encodes the message by inserting redundant bits within the message. These redundant bits are extra bits that are generated and inserted at specific positions in the message itself to enable error detection and correction. When the destination receives this message, it performs recalculations to detect errors and find the bit position that has error.

## Encoding a message by Hamming Code (Sender)

The procedure used by the sender to encode the message encompasses the following steps −

**Step 1** − Calculation of the number of redundant bits.
**Step 2** − Positioning the redundant bits.
**Step 3** − Calculating the values of each redundant bit.

Once the redundant bits are embedded within the message, this is sent to the user.

**Step 1** − Calculation of the number of redundant bits.

If the message contains $mm$number of data bits, $rr$number of redundant bits are added to it so that $mr$ is able to indicate at least $(m + r+ 1)$ different states. Here, $(m + r)$ indicates location of an error in each of $(m + r)$ bit positions and one additional state indicates no error. Since, $rr$ bits can indicate $2^r r$ states, $2^r r$ must be at least equal to $(m + r + 1)$. Thus the following equation should hold $2r \geq m+r+1$

**Step 2** − Positioning the redundant bits.

The $r$ redundant bits placed at bit positions of powers of 2, i.e. 1, 2, 4, 8, 16 etc. They are referred in the rest of this text as $r_1$ (at position 1), $r_2$ (at position 2), $r_3$ (at position 4), $r_4$ (at position 8) and so on.

**Step 3** − Calculating the values of each redundant bit.

The redundant bits are parity bits. A parity bit is an extra bit that makes the number of 1s either even or odd. The two types of parity are −

**Even Parity** − Here the total number of bits in the message is made even.
**Odd Parity** − Here the total number of bits in the message is made odd

## Decoding a message in Hamming Code (Receiver)

Once the receiver gets an incoming message, it performs recalculations to detect errors and correct them. The steps for recalculation are −

**Step 1** − Calculation of the number of redundant bits.
**Step 2** − Positioning the redundant bits.
**Step 3** − Parity checking.
**Step 4** − Error detection and correction

**Ex: Let us assume the even parity hamming code from the above example (111001101) is transmitted and the received code is (110001101). Now from the received code, let us detect and correct the error.**

*To detect the error, let us construct the bit location table.*

| Bit Location | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|---|
| Bit designation | D5 | **P4** | D4 | D3 | D2 | **P3** | D1 | **P2** | **P1** |
| Binary representation | 1001 | 1000 | 0111 | 0110 | 0101 | 0100 | 0011 | 0010 | 0001 |
| Received code | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |

*Checking the parity bits*

For P1 : Check the locations 1, 3, 5, 7, 9. There is three 1s in this group, which is wrong for even parity. Hence the bit value for P1 is 1.

For P2 : Check the locations 2, 3, 6, 7. There is one 1 in this group, which is wrong for even parity. Hence the bit value for P2 is 1.

For P3 : Check the locations 3, 5, 6, 7. There is one 1 in this group, which is wrong for even parity. Hence the bit value for P3 is 1.

For P4 : Check the locations 8, 9. There are two 1s in this group, which is correct for even parity. Hence the bit value for P4 is 0.

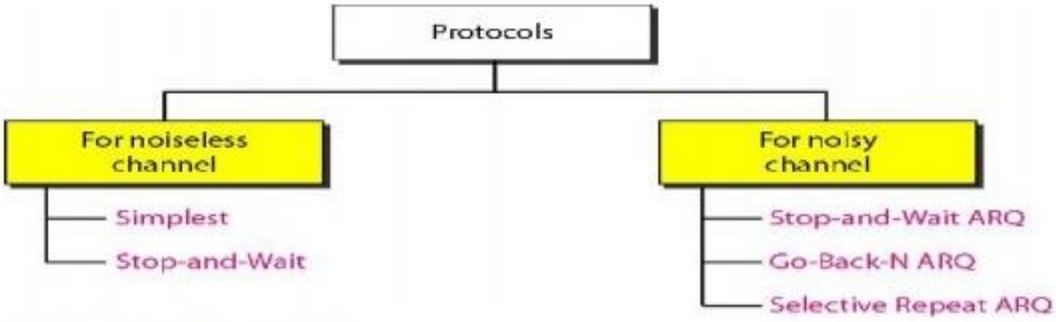
The resultant binary word is 0111. It corresponds to the bit location 7 in the above table. The error is detected in the data bit D4. The error is 0 and it should be changed to 1. **Thus the corrected code is 111001101**.

## Elementary Data Link Layer Protocols

Based on error control and flow control mechanism the elementary data link layer protocols are divided in two transmission channels.

1. Noise Less

2. Noisy



## Simplex Protocol or Unrestricted Protocol or Utopian Protocol

a) Unrestricted Simplex protocol :

- It is also called as Utopian Simplex protocol.
- This protocol can be used if the following conditions exist :-

① Data are transmitted in one direction only.

② Both $\overset{\text{Sender}}{\text{transmitting}}$ and $\overset{\text{receiver}}{\text{receiving}}$ systems are always ready

③ Processing time can be ignored.

④ Infinite buffer space is available

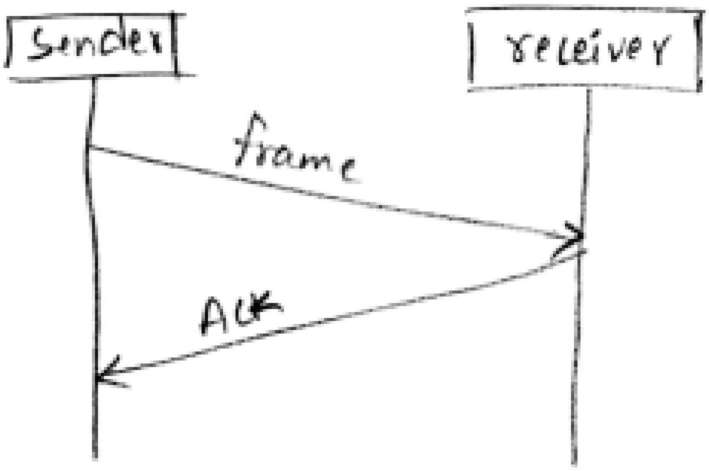
⑤ Communication channel never damages or looses frames.

⑥ No sequence numbers or Acknowledgements are used here.

- This protocol is unrealistic because it does n't handle either flow control or error control.
- Its processing is close to that of an unacknowledged connectionless service.

## Simplex Stop and Wait Protocol error free channel

- The communication channel is assumed to be error free.
- The data traffic is half-duplex.
- In this protocol, receiver provides a feedback to the sender.
- It means that when the sender sends the data, the receiver receives it & sends a little dummy frame back to the sender giving permission to the sender to transmit the next frame.
- After having sent a frame, the sender is required by the protocol to wait until the dummy(ACK) frame arrives.
- Protocols in which the sender sends one frame & then waits for an ACK before proceeding to the next frame are called stop-and-wait protocols.



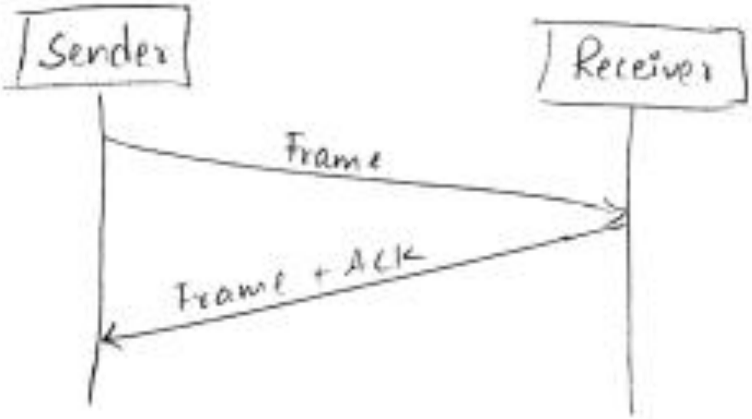
## Simplex stop and wait protocol for noisy channel

- Consider the communication channel is a Noisy channel (ie., the channel that makes errors).
- Frames may be either damaged or lost completely.
- If a frame is damaged, the error is detected by using the checksum.
- If the data is lost or the ACK is lost, the send it can be identified by using timers.
- When the sender transmits a frame, it also starts a timer.
- If the timer expires, then the sender retransmits the frame.
- Sequence numbers are used to distinguish b/w the original frame & the retransmitted frame.
- Protocols in which the sender wait for a +ve Ack before advancing to the next data item are often called ARQ( Automatic Repeat ReQuest) or PAR ( Positive Acknowledgement with Retransmission).

## Sliding window protocol(one bit)

Piggybacking: when a data frame arrives, instead of immediately sending a separate ACK, the receiver waits until the next frame.

- The ack is attached to the outgoing data frame.
- The technique of temporarily delaying outgoing acknowledgements so that they can be attached onto the next outgoing data frame is known as piggybacking



adv :- better use of channel bandwidth

disadv :- If the receiver waits too long, then at the sender the timer will be off & the sender retransmits the frame.

- sending window :- At any instant of time, the sender maintains a set of sequence numbers corresponding to the frames it is permitted to send.

- receiving window : At any instant of time, the receiver maintains a set of frames it is permitted to receive.

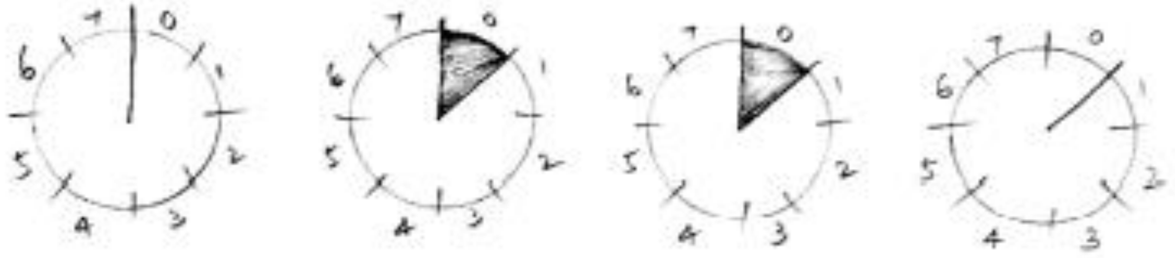
- Sequence number range is 0 to $2^n-1$

Sliding window protocols

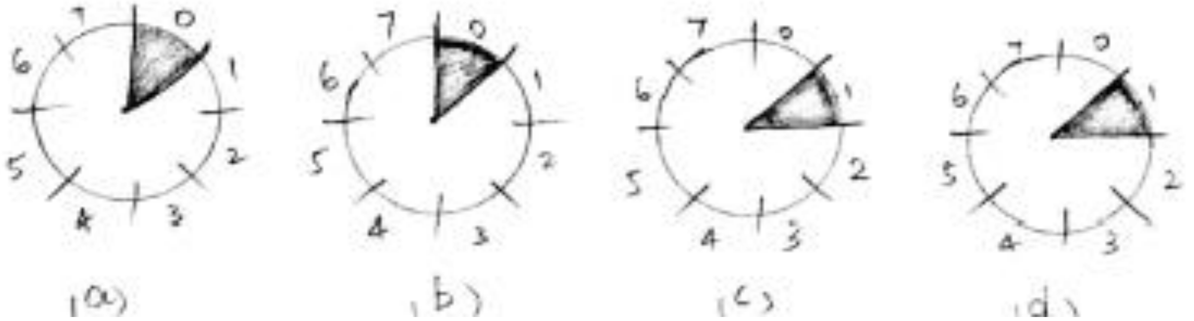
- one-bit sliding window protocol
- A protocol using Go-Back-N
- A protocol using Selective Repeat.

(a) one-bit sliding window protocol.

Sender



Receiver



→ The above example has a sliding window of size 1, with a 3-bit sequence number.

↳ sequence number range is 0 to $2^3-1$

$= 0$ to $7$

(a) Initially

(b) After the first frame has been sent

(c) After the first frame has been received

(d) After the first Ack has been received.

(a) <u>Sender</u> :- Initially when data transmission is not yet started.

<u>Receiver</u> :- waiting for a frame of sequence num = 0.

(b) <u>Sender</u> :- Sender sends a frame of sequence num = 0

<u>Receiver</u> :- waiting for a frame of sequence num = 0

(c) <u>Sender</u> :- Sender waiting for an ACK for frame of seq num = 0

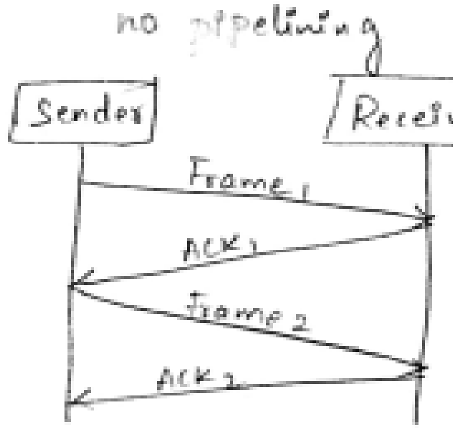
<u>Receiver</u> :- sends the ACK and waits for the next frame of seq num = 1

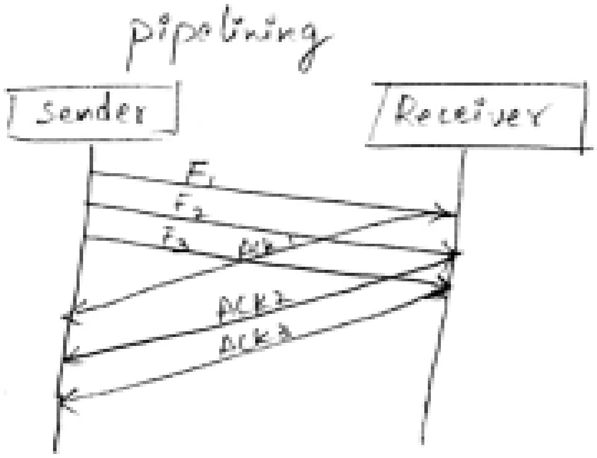
(d) <u>Sender</u> :- Data transmission of frame with seq num = 0 is completed & not sending any data (idle).

<u>Receiver</u> : waiting for a frame of seq num = 1.

# G0 BACK N ARQ

pipelining :- It is a technique in which multiple frames are sent at a time without waiting for the corresponding individual acknowledgements.

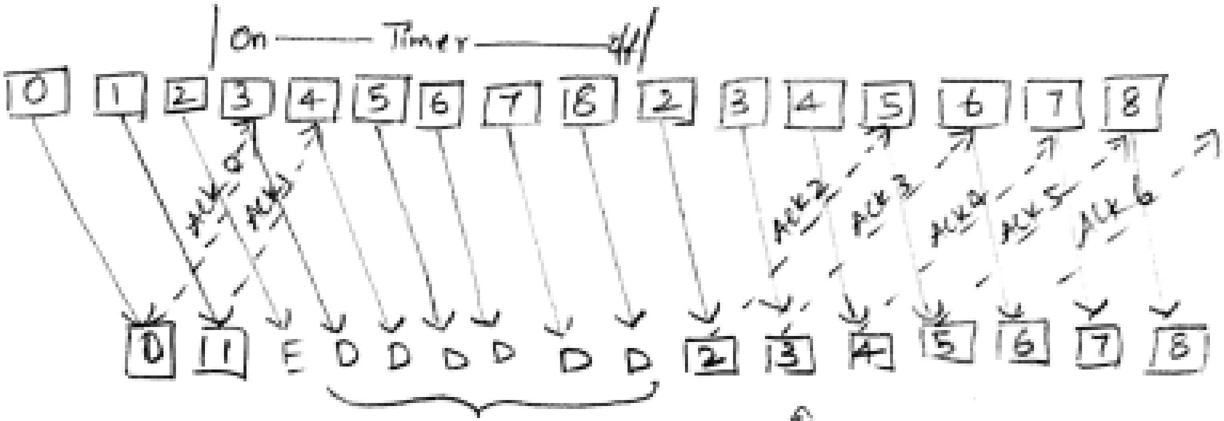




ⓑ A protocol using Go-Back-N :-

- In this protocol, the sender retransmits all the frames that are transmitted after the damaged/lost frame.
- It error rate is high, it wastes a lot of bandwidth.
- In this protocol, the receiver do not store the frames received after the damaged frame until the damaged frame is retransmitted.
- It is a mechanism to detect & control the errors.
- Go-Back-N protocol is shown in the below diagram.
- Frame 2 is lost, so all the frames followed by frame 2 are deleted (discarded).
- All the frames from frame 2 to 8 are retransmitted.

- Go-Back-N protocol performs pipelining. Hence all the frames from 0 to 8 are sent at a time without waiting for individual acknowledgements.
- The damaged or discarded frames will be retransmitted after all the 8 frames are transmitted.



Frames discarded

Go-Back-N for Frame lost and delayed ACK

receiver's window size is 1

## SELECTIVE REPEAT ARQ

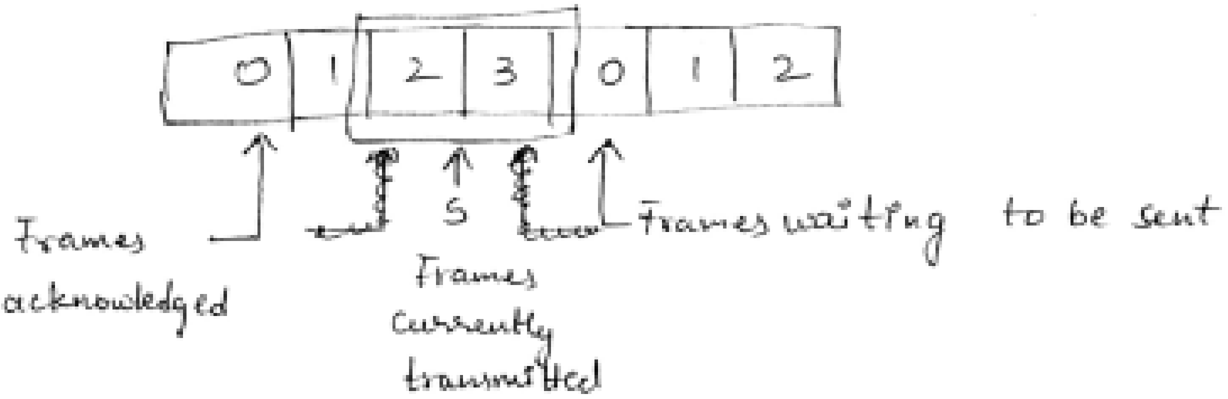
(c) A protocol using selective repeat.

- The go-back-n protocol works well if errors are rare, but if the channel has high error rate, it wastes lot of bandwidth on retransmitted frames.
- An alternative approach is selective repeat.
- The selective repeat protocol retransmits only that frame which is damaged or lost.
- The sender maintains a buffer (Sender window) having the

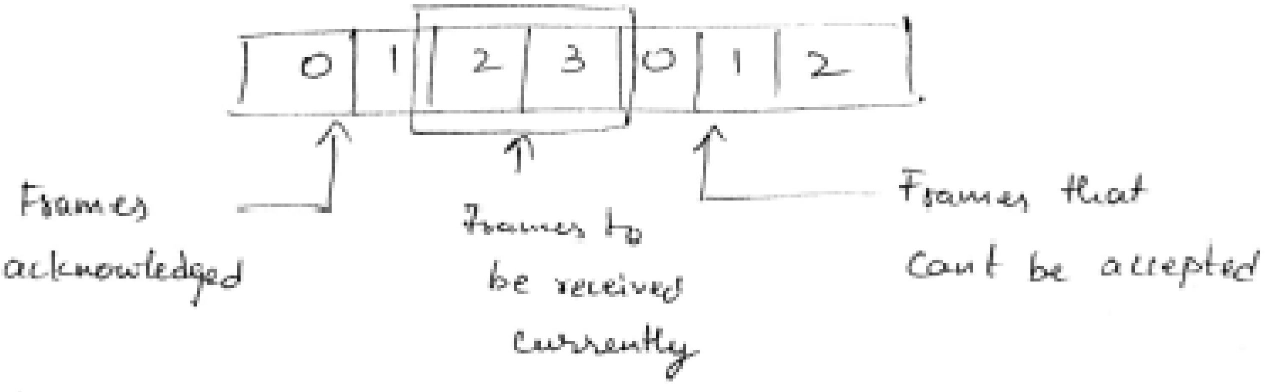
sequence numbers of the frames that the sender is allowed to send.

- The receiver also maintains a buffer (receiver window) having the sequence numbers of the frames that the receiver is allowed to receive.
- In this protocol, receiver stores the frames received after the damaged frame in the buffer until the damaged frame is replaced.

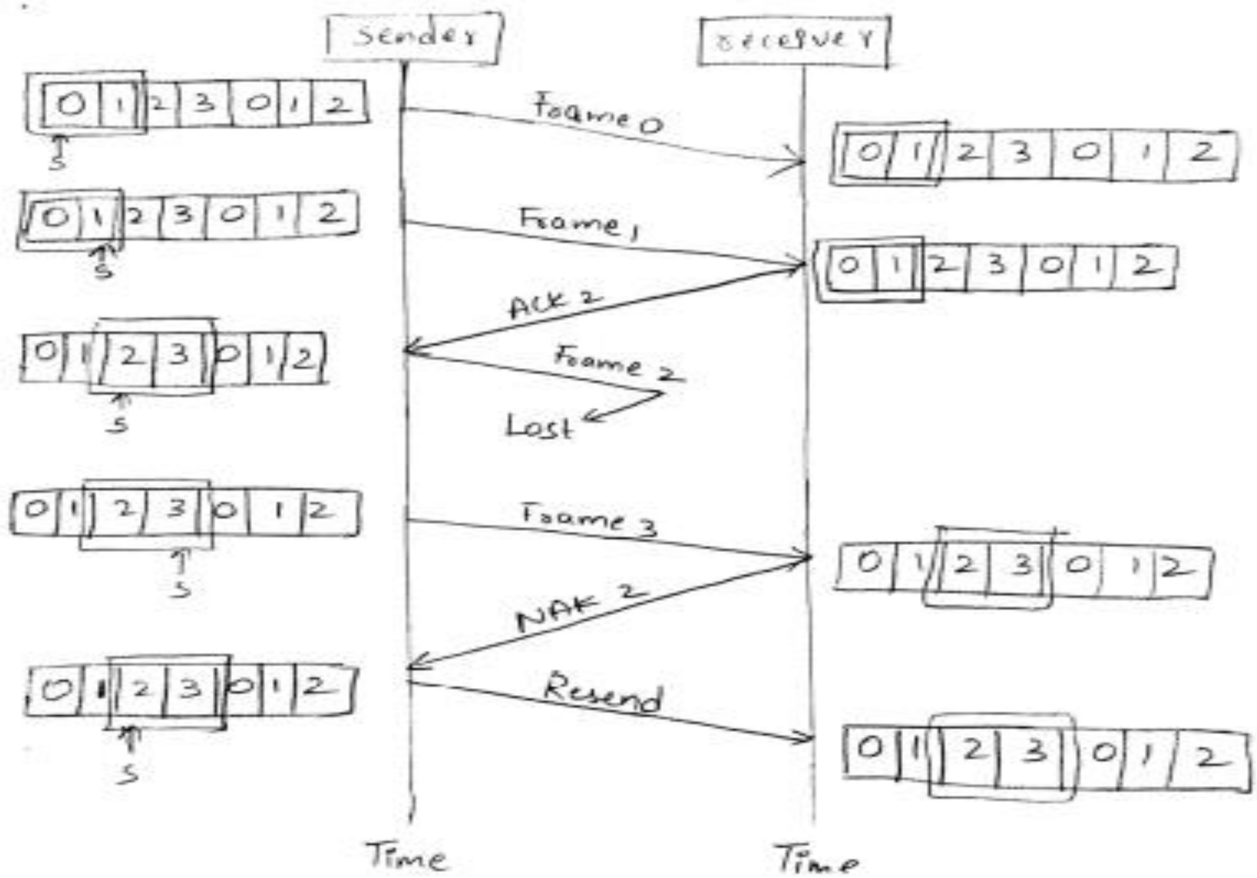
Sender window :



receiver window :



∴ The selective repeat protocol is shown in the below example. The Frame 2 is lost, only that frame is retransmitted.



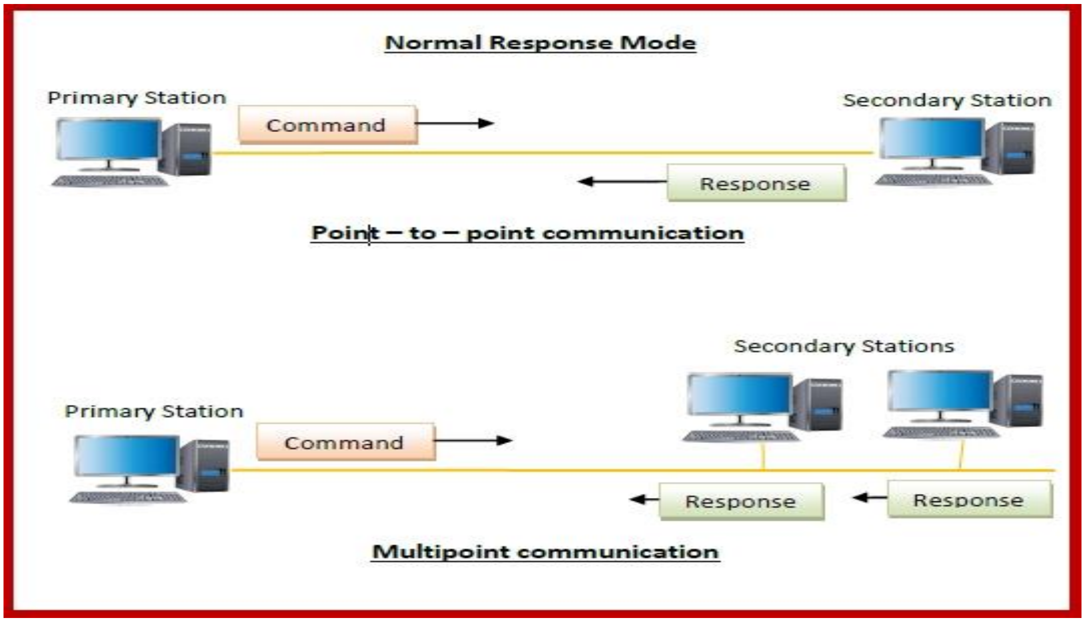
## Data Link Layer Protocol in HDLC

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. It is a bit - oriented protocol that is applicable for both point - to - point and multipoint communications.

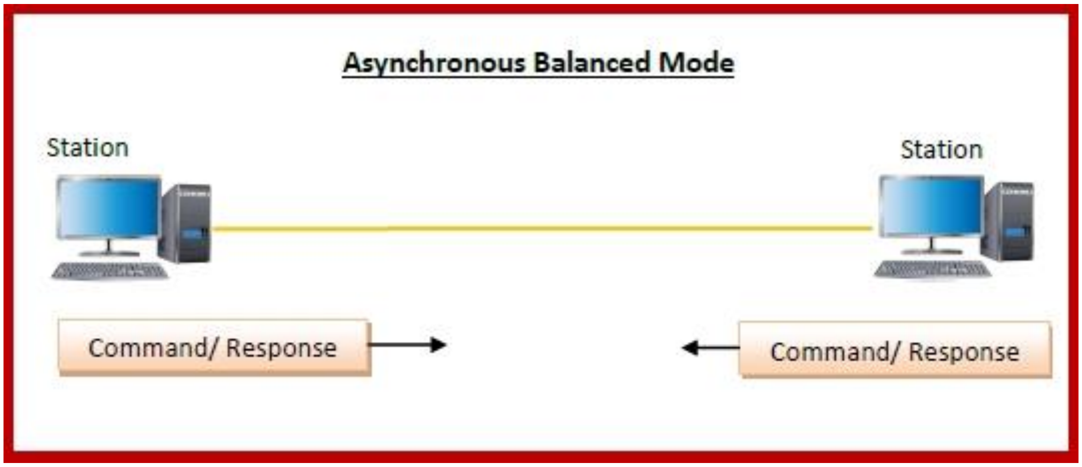
### Transfer Modes

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

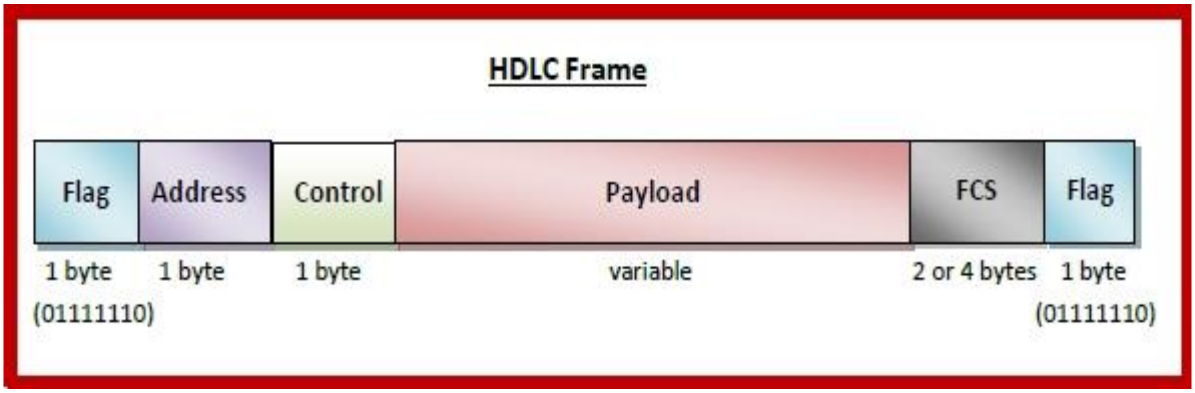
**Normal Response Mode (NRM)** − Here, two types of stations are there, a primary station that send commands and secondary station that can respond to received commands. It is used for both point - to - point and multipoint communications.



**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 Structure



**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 bit 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 bit of control field of U-frame is 11.

## Point to point protocol (PPP)

- The PPP stands for Point-to-Point protocol. It is the most commonly used protocol for point-to-point access. Suppose the user wants to access the internet from the home, the PPP protocol will be used.
- The PPP protocol can be used on synchronous link like ISDN as well as asynchronous link like dial-up. It is mainly used for the communication between the two devices.
- It is also known as a byte-oriented protocol

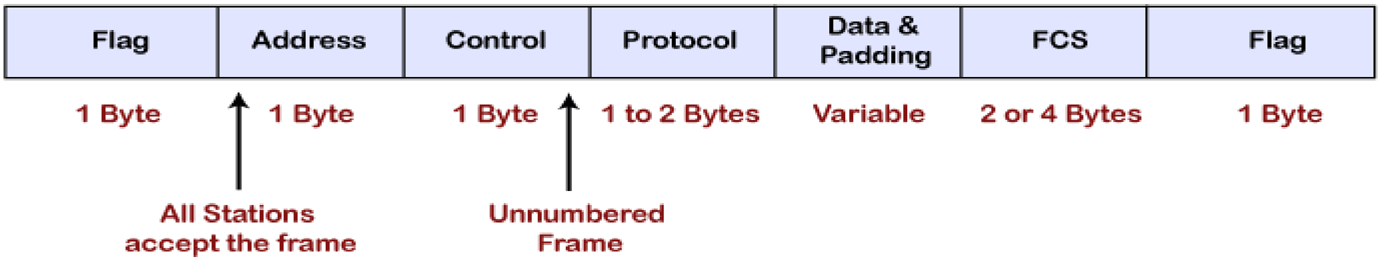
## Services provided by PPP

- It defines the format of frames through which the transmission occurs.
- It defines the link establishment process.
- It defines data exchange process, i.e., how data will be exchanged, the rate of the exchange.
- The main feature of the PPP protocol is the encapsulation. It defines how network layer data and information in the payload are encapsulated in the data link frame.
- It defines the authentication process between the two devices. The authentication between the two devices, handshaking and how the password will be exchanged between two devices are decided by the PPP protocol.

## Services not provided by the PPP protocol(RFC1661)

- It does not support flow control mechanism.
- It has a very simple error control mechanism.
- As PPP provides point-to-point communication, so it lacks addressing mechanism to handle frames in multipoint configuration.

## Frame format of PPP protocol



| Flag | Address | Control | Protocol | Data & Padding | FCS | Flag |
|---|---|---|---|---|---|---|
| 1 Byte | 1 Byte | 1 Byte | 1 to 2 Bytes | Variable | 2 or 4 Bytes | 1 Byte |

**Flag:** The flag field is used to indicate the start and end of the frame. The flag field is a 1-byte field that appears at the beginning and the ending of the frame. The pattern of the flag is similar to the bit pattern in HDLC, i.e., 01111110.

**Address:** It is a 1-byte field that contains the constant value which is 11111111. These 8 ones represent a broadcast message.

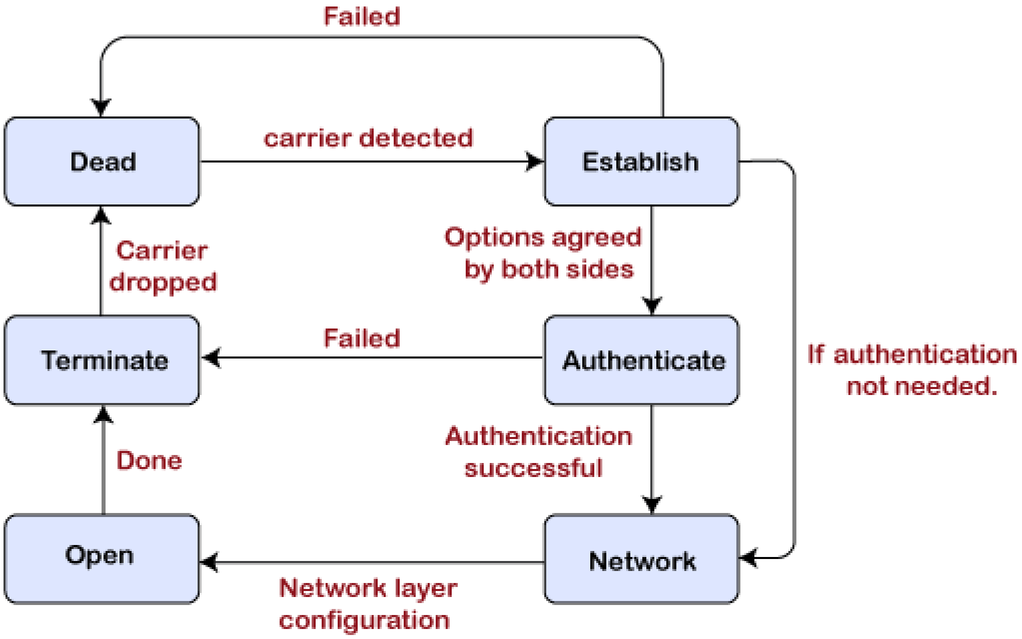
**Control:** It is a 1-byte field which is set through the constant value, i.e., 11000000. It is not a required field as PPP does not support the flow control and a very limited error control mechanism. The control field is a mandatory field where protocol supports flow and error control mechanism.

**Protocol:** It is a 1 or 2 bytes field that defines what is to be carried in the data field. The data can be a user data or other information.

**Payload:** The payload field carries either user data or other information. The maximum length of the payload field is 1500 bytes

**CRC:** It is a 16-bit field which is generally used for error detection

## Transition phases of PPP protocol



Failed
Dead
carrier detected
Establish
Carrier dropped
Options agreed by both sides
Terminate
Failed
Authenticate
If authentication not needed.
Done
Authentication successful
Open
Network
Network layer configuration

Transition phases

**Dead:** Dead is a transition phase which means that the link is not used or there is no active carrier at the physical layer.

**Establish:** If one of the nodes starts working then the phase goes to the establish phase. In short, we can say that when the node starts communication or carrier is detected then it moves from the dead to the establish phase.

**Authenticate:** It is an optional phase which means that the communication can also moves to the authenticate phase. The phase moves from the establish to the authenticate phase only when both the communicating nodes agree to make the communication authenticated.

**Network:** Once the authentication is successful, the network is established or phase is network. In this phase, the negotiations of network layer protocols take place.

**Open:** After the establishment of the network phase, it moves to the open phase. Here open phase means that the exchange of data takes place. Or we can say that it reaches to the open phase after the configuration of the network layer.

**Terminate:** When all the work is done then the connection gets terminated, and it moves to the terminate phase.

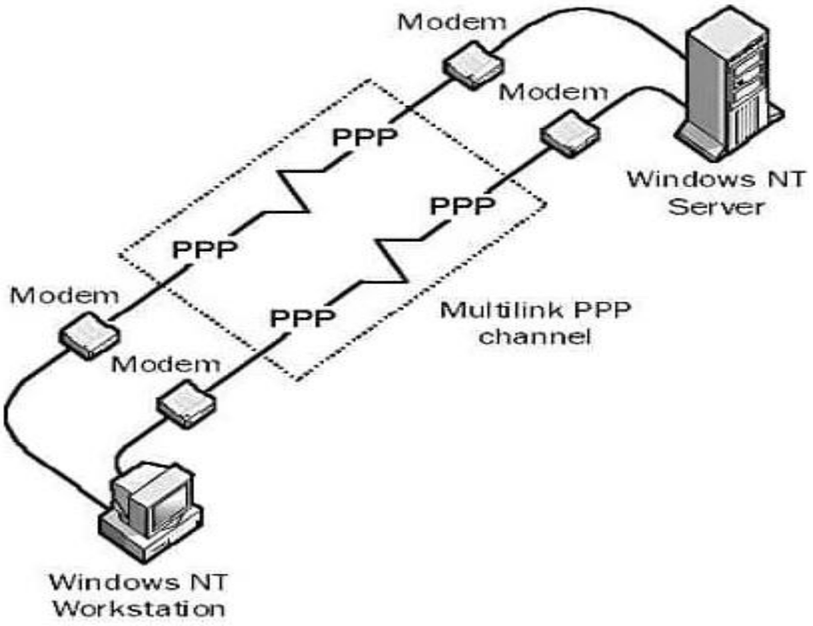
## Components in PPP

- Link Control Protocol (LCP)
- Authentication protocols

**Link Control Protocol (LCP):** The role of LCP is to establish, maintain, configure, and terminate the links. It also provides negotiation mechanism.

**Authentication protocols:** This protocol plays a very important role in the PPP protocol because the PPP is designed for use over the dial-up links where the verification of user identity is necessary. Thus this protocol is mainly used to authenticate the endpoints for the use of other services.

## Multilink PPP



**Multilink PPP** (also referred to as **MLPPP**, **MP**, **MPPP**, **MLP**, or Multilink) provides a method for spreading traffic across multiple distinct PPP connections. It is defined in RFC 1990. It can be used, for example, to connect a home computer to an Internet Service Provider using two traditional 56k modems, or to connect a company through two leased lines.

On a single PPP line frames cannot arrive out of order, but this is possible when the frames are divided among multiple PPP connections. Therefore, Multilink PPP must number the fragments so they can be put in the right order again when they arrive.

Multilink PPP is an example of a link aggregation technology. Cisco IOS Release 11.1 and later supports Multilink PPP.

Multilink PPP is a communications strategy that makes use of the basic concept of **point-to-point protocol**. Essentially, the approach allows for the utilization of more than one PPP communication port in order to achieve a higher amount of **bandwidth** to work with. This type of **communications protocol** can often be employed with a personal computer and thus enhance the overall efficiency of many tasks The utilization of Multilink PPP can be especially helpful in locations where dial up connections to the Internet are the only alternative. The end user will make use of two different modems to establish independent connections to the Internet. The connections are made to the same Internet **Service Provider**. Assuming that the **ISP** allows for this type of connectivity, the end user can effectively double the operating speed, as the two connections divide the requested data into packets that are simultaneously transmitted through each connection, then recombined at the user end.

However, there are other applications for Multilink PPP that go beyond simply increasing the speed associated with dial up service. The same concept is often employed in fiber optic systems that function primarily as a means of providing audio signaling. **Cable modems** can also employ Multilink PPP to enhance **transmission**. Satellite transmissions can also make use of the principles of Multilink PPP in order to increase the efficiency of data transfers. The overall simplicity of employing this type of multiple connection protocol makes it easily adaptable to a number of different situations, and can often be an ideal solution for persons located in relatively isolated areas.