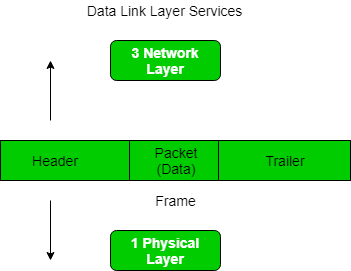
**COMPUTER COMMUNICATION**

**UNIT IV**

**Framing**

Framing is a point-to-point connection between two computers or devices consists of a wire in which data is transmitted as a stream of bits. However, these bits must be framed into discernible blocks of information. 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.



At data link layer, it extracts message from sender and provide it to receiver by providing sender’s and receiver’s address. The advantage of using frames is that data is broken up into recoverable chunks that can easily be checked for corruption.

**Problems in Framing –**

* **Detecting start of the frame:**When a frame is transmitted, every station must be able to detect it. Station detect frames by looking out for special sequence of bits that marks the beginning of the frame i.e. SFD (Starting Frame Delimeter).
* **How do station detect a frame:**Every station listen to link for SFD pattern through a sequential circuit. If SFD is detected, sequential circuit alerts station. Station checks destination address to accept or reject frame.
* **Detecting end of frame:**When to stop reading 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.

**Drawback:**It suffers from internal fragmentation if data size is less than frame size

**Solution:**Padding

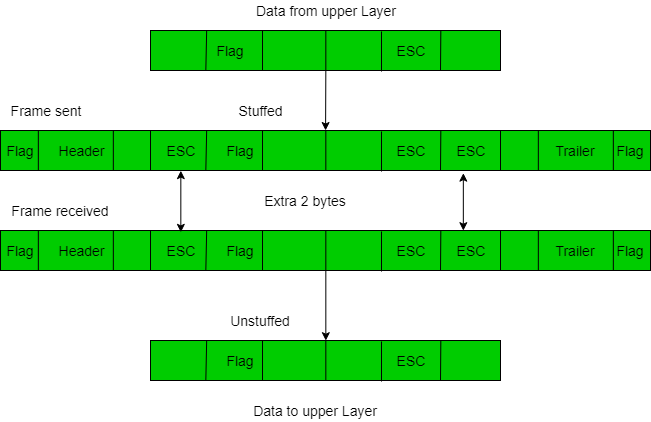
**2. Variable size –** In this there is need to define end of frame as well as beginning of next frame to distinguish. This can be done in two ways:

**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 Delimeter (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:

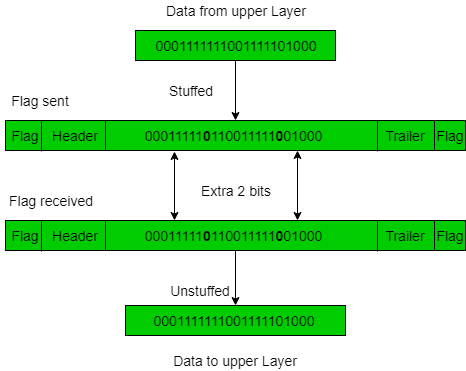
**1. Character/Byte Stuffing:**Used when frames consist of character. If data contains ED then, byte is stuffed into data to diffentiate it from ED.

Let ED = “$” –> if data contains ‘$’ anywhere, it can be escaped using ‘O’ character.  
–> if data contains ‘O$’ then, use ‘OOO$'($ is escaped using O and O is escaped using O).



**Disadvantage –** It is very costly and obsolete method.

**2. Bit Stuffing:**Let ED = 01111 and if data = 01111  
**–>** Sender stuffs a bit to break the pattern i.e. here appends a 0 in data = 0111**0**1.  
**–>** Receiver receives the frame.  
**–>** If data contains 011101, receiver removes the 0 and reads the data.



**Examples –**

* If Data –> 011100011110 and ED –> 01111 then, find data after bit stuffing ?

–> 0111**0**000111**0**10

* If Data –> 110001001 and ED –> 1000 then, find data after bit stuffing ?

–> 1100**1**0100**1**1

* [Gate CS 2014](https://tutorialspoint.dev/slugresolver/gate-gate-cs-2014-set-3-question-34/)
* [Gate IT 2004](https://tutorialspoint.dev/slugresolver/gate-gate-it-2004-question-80/)

**Flow control**

If the rate at which the data are absorbed by the receiver is less than the rate at which data are produced in the sender, the data link layer imposes a flow control mechanism to avoid overwhelming the receiver.

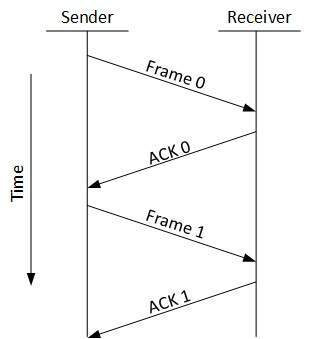
Like the data link layer, the transport layer is responsible for flow control. However, flow control at this layer is performed end to end rather than across a single link.

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

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

**Stop and Wait**

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



**Error control**

The data link layer adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged or lost frames. It also uses a mechanism to recognize duplicate frames. Error control is normally achieved through a trailer added to the end of the frame.

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

Requirements for error control mechanism:

* **Error detection**  - The sender and receiver, either both or any, must ascertain that there is some error in the transit.
* **Positive ACK**  - When the receiver receives a correct frame, it should acknowledge it.
* **Negative ACK**  - When the receiver receives a damaged frame or a duplicate frame, it sends a NACK back to the sender and the sender must retransmit the correct frame.
* **Retransmission**:  The sender maintains a clock and sets a timeout period. If an acknowledgement of a data-frame previously transmitted does not arrive before the timeout the sender retransmits the frame, thinking that the frame or it’s acknowledgement is lost in transit.

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

# **Stop and Wait ARQ**

### **Characteristics**

* Used in Connection-oriented communication.
* It offers error and flow control
* It is used in Data Link and Transport Layers
* Stop and Wait ARQ mainly implements Sliding Window Protocol concept with Window Size 1

### **Useful Terms:**

### **Propagation Delay: Amount of time taken by a packet to make a physical journey from one router to another router.**

Propagation Delay = (Distance between routers) / (Velocity of propagation)

RoundTripTime (**RTT**) = 2\* Propagation Delay

TimeOut (**TO**) =  2\* RTT

Time To Live (**TTL**) = 2\* TimeOut. (Maximum TTL is 180 seconds)

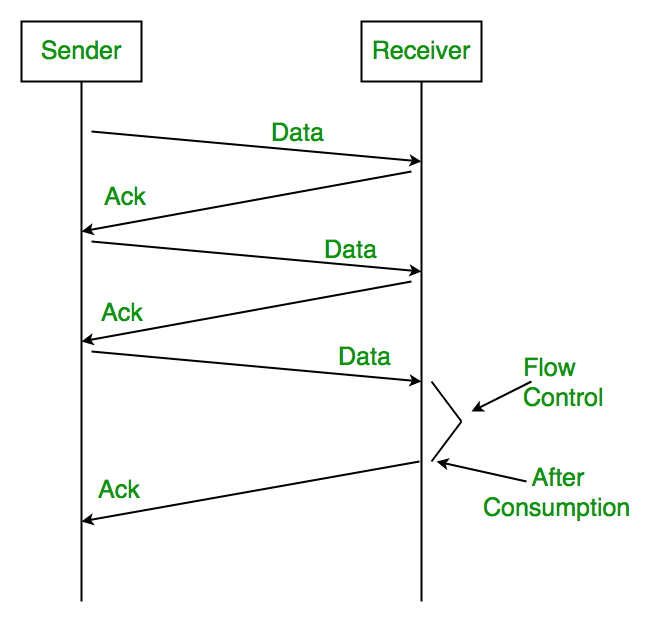
### **Simple Stop and Wait**

#### **Sender:**

Rule 1) Send one data packet at a time.  
Rule 2) Send next packet only after receiving acknowledgement for previous.

#### **Receiver:**

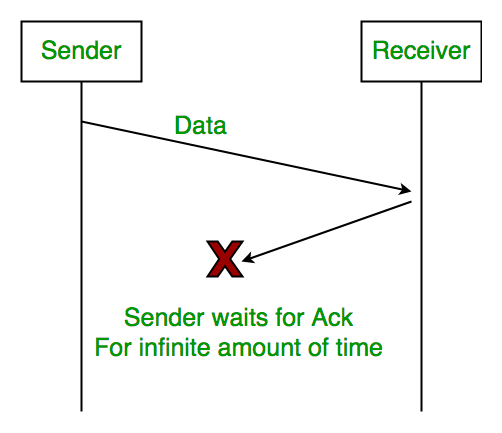
Rule 1) Send acknowledgement after receiving and consuming of data packet.  
Rule 2) After consuming packet acknowledgement need to be sent (Flow Control)



**1. Lost Data**

# 

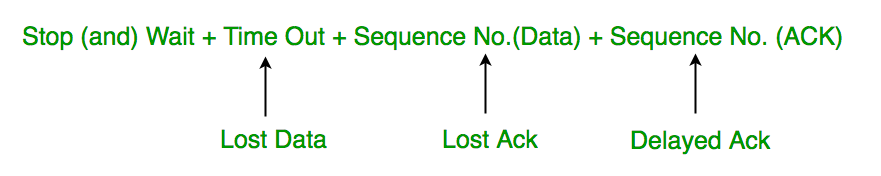
**2. Lost Acknowledgement:**



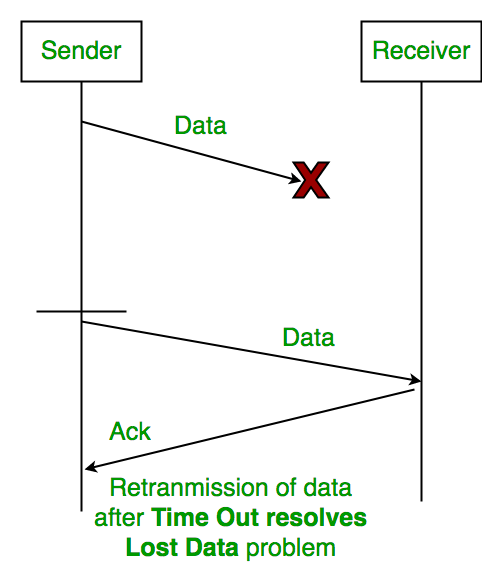
**3. Delayed Acknowledgement/Data:** After timeout on sender side, a long delayed acknowledgement might be wrongly considered as acknowledgement of some other recent packet.

### **Stop and Wait ARQ (Automatic Repeat Request)**

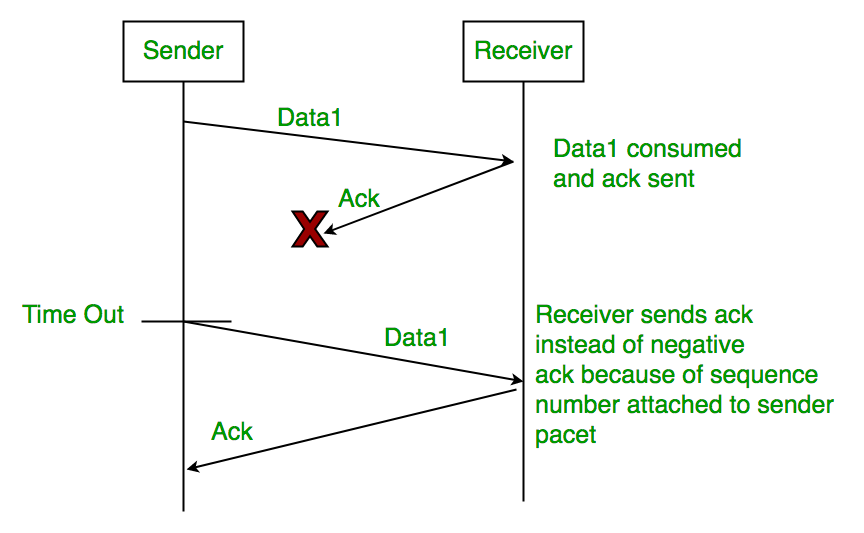
Above 3 problems are resolved by Stop and Wait ARQ (Automatic Repeat Request) that does both error control and flow control.



1. **Time Out:**



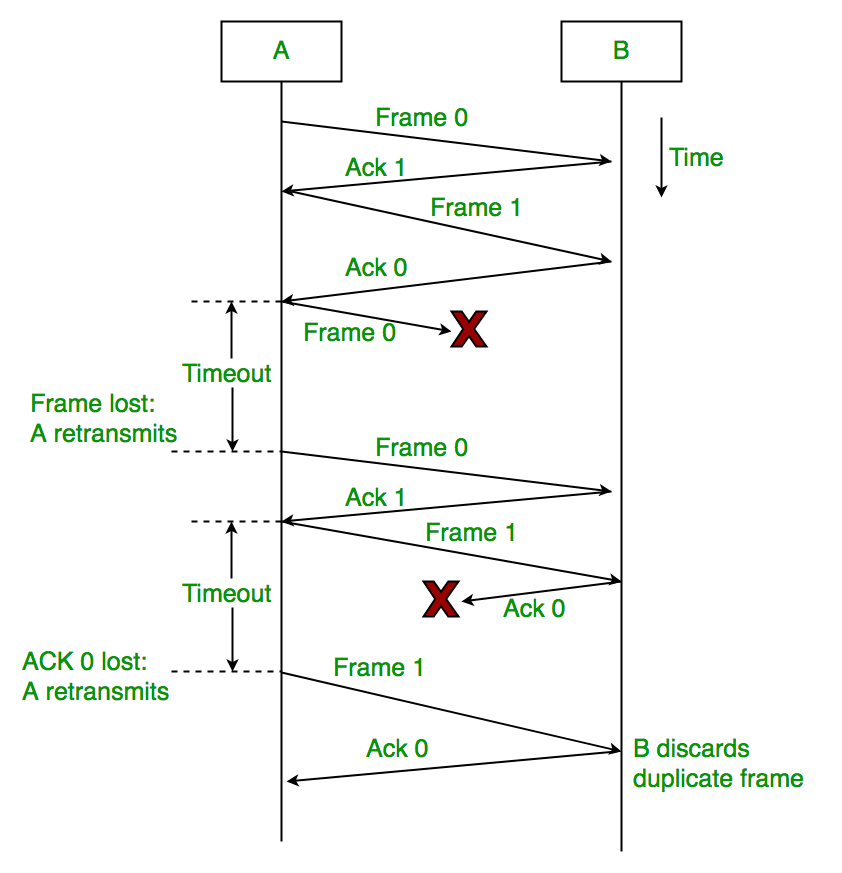
1. **Sequence Number (Data)**



**3. Delayed Acknowledgement:**  
This is resolved by introducing sequence number for acknowledgement also.

### **Working of Stop and Wait ARQ:**

1) Sender A sends a data frame or packet with sequence number 0.  
2) Receiver B, after receiving data frame, sends and acknowledgement with sequence number 1 (sequence number of next expected data frame or packet)  
There is only one bit sequence number that implies that both sender and receiver have buffer for one frame or packet only.



#### **Characteristics of Stop and Wait ARQ:**

* It uses link between sender and receiver as half duplex link
* Throughput = 1 Data packet/frame per  RTT
* If Bandwidth\*Delay product is very high, then stop and wait protocol is not so useful. The sender has to keep waiting for acknowledgements before sending the processed next packet.
* It is an example for “**Closed Loop OR connection oriented**“ protocols
* It is an special category of SWP where its window size is 1
* Irrespective of number of packets sender is having stop and wait protocol  requires only  2 sequence numbers 0 and 1

The Stop and Wait ARQ solves main three problems, but may cause big performance issues as sender always waits for acknowledgement even if it has next packet ready to send. Consider a situation where you have a high bandwidth connection and propagation delay is also high (you are connected to some server in some other country though a high speed connection). To solve this problem, we can send more than one packet at a time with a larger sequence numbers. We will be discussing these protocols in next articles.

So Stop and Wait ARQ may work fine where propagation delay is very less for example LAN connections, but performs badly for distant connections like satellite connection.

# **Sliding Window Protocol**

**Sliding Window Protocol** is actually a theoretical concept in which we have only talked about what should be the sender window size (1+2a) in order to increase the efficiency of stop and wait arq. Now we will talk about the practical implementations in which we take care of what should be the size of receiver window. Practically it is implemented in two protocols namely :

1. Go Back N (GBN)
2. Selective Repeat (SR)

In this article, we will explain you about the first protocol which is GBN in terms of three main characteristic features and in the next part we will be discussing SR as well as comparison of both these protocols

### **Go Back N (GBN) Protocol**

The three main characteristic features of GBN are:

1. **Sender Window Size (WS)**  
   It is N itself. If we say the protocol is GB10, then Ws = 10. N should be always greater than 1 in order to implement pipelining. For N = 1, it reduces to Stop and Wait protocol.

Efficiency Of GBN = N/(1+2a)

where a = Tp/Tt

If B is the bandwidth of the channel, then

Effective Bandwidth or Throughput

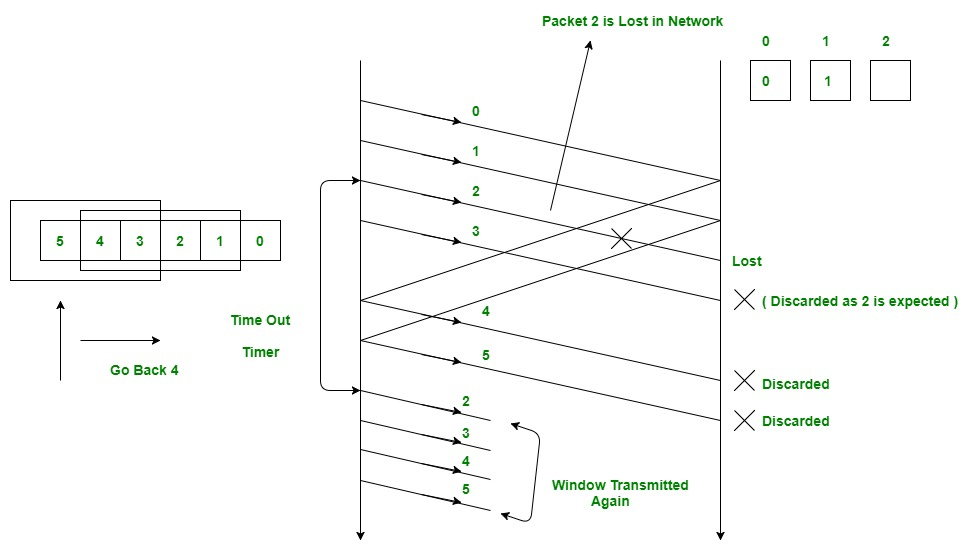
= Efficiency \* Bandwidth

= (N/(1+2a)) \* B

1. **Receiver Window Size (WR)**

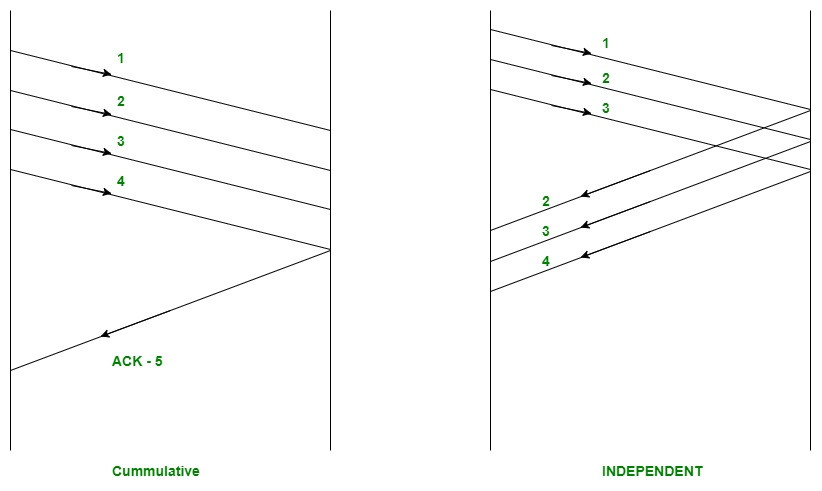
WR is always 1 in GBN.

Now what exactly happens in GBN, we will explain with a help of example. Consider the diagram given below. We have sender window size of 4. Assume that we have lots of sequence numbers just for the sake of explanation. Now the sender has sent the packets 0, 1, 2 and 3. After acknowledging the packets 0 and 1, receiver is now expecting packet 2 and sender window has also slided to further transmit the packets 4 and 5. Now suppose the packet 2 is lost in the network, Receiver will discard all the packets which sender has transmitted after packet 2 as it is expecting sequence number of 2. On the sender side for every packet send there is a time out timer which will expire for packet number 2. Now from the last transmitted packet 5 sender will go back to the packet number 2 in the current window and transmit all the packets till packet number 5. That’s why it is called Go Back N. Go back means sender has to go back N places from the last transmitted packet in the unacknowledged window and not from the point where the packet is lost.



**Acknowledgements**  
There are 2 kinds of acknowledgements namely:

* **Cumulative Ack**: One acknowledgement is used for many packets. The main advantage is traffic is less. A disadvantage is less reliability as if one ack is the loss that would mean that all the packets sent are lost.
* **Independent Ack**: If every packet is going to get acknowledgement independently. Reliability is high here but a disadvantage is that traffic is also high since for every packet we are receiving independent ack.



**GBN uses Cumulative Acknowledgement**. At the receiver side, it starts a acknowledgement timer whenever receiver receives any packet which is fixed and when it expires, it is going to send a cumulative Ack for the number of packets received in that interval of timer. If receiver has received N packets, then the Acknowledgement number will be N+1. Important point is Acknowledgement timer will not start after the expiry of first timer but after receiver has received a packet.

**Selective Repeat Protocol**

The go-back-n protocol works well if errors are less, but if the line is poor it wastes a lot of bandwidth on retransmitted frames. An alternative strategy, the selective repeat protocol, is to allow the receiver to accept and buffer the frames following a damaged or lost one.

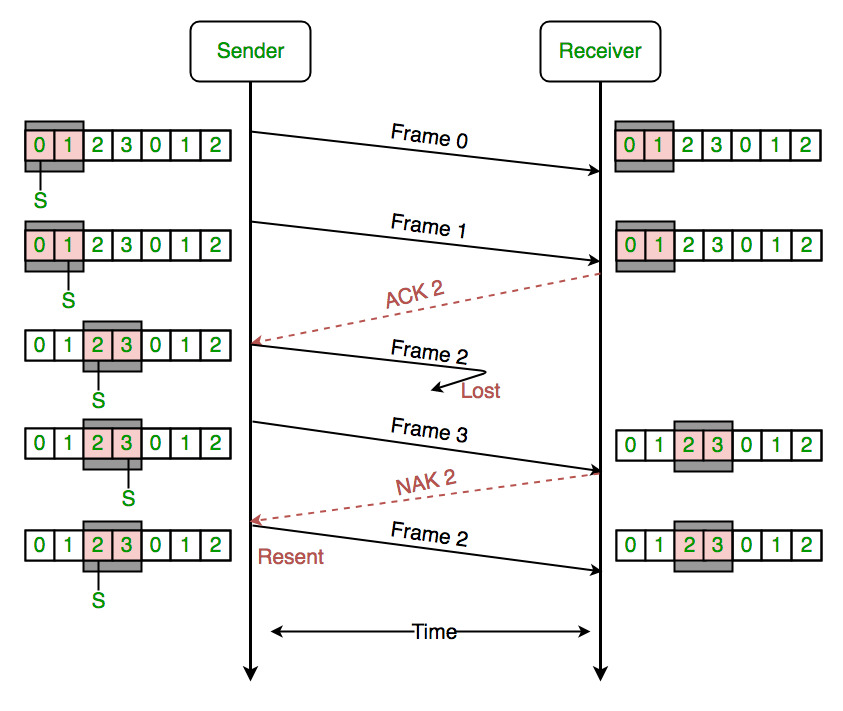
Selective Repeat attempts to retransmit only those packets that are actually lost (due to errors) :

* Receiver must be able to accept packets out of order.
* Since receiver must release packets to higher layer in order, the receiver must be able to buffer some packets.

**Selective Repeat Protocol (SRP) :**

This protocol(SRP) is mostly identical to GBN protocol, except that buffers are used and the receiver, and the sender, each maintain a window of size. SRP works better when the link is very unreliable. Because in this case, retransmission tends to happen more frequently, selectively retransmitting frames is more efficient than retransmitting all of them. SRP also requires full duplex link. backward acknowledgements are also in progress.

* Sender’s Windows ( Ws) = Receiver’s Windows ( Wr).
* Window size should be less than or equal to half the sequence number in SR protocol. This is to avoid packets being recognized incorrectly. If the windows size is greater than half the sequence number space, then if an ACK is lost, the sender may send new packets that the receiver believes are retransmissions.
* Sender can transmit new packets as long as their number is with W of all unACKed packets.
* Sender retransmit un-ACKed packets after a timeout – Or upon a NAK if NAK is employed.
* Receiver ACKs all correct packets.
* Receiver stores correct packets until they can be delivered in order to the higher layer.
* In Selective Repeat ARQ, the size of the sender and receiver window must be at most one-half of 2^m.



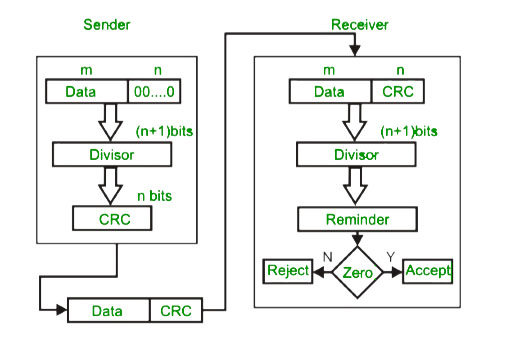
**Cyclic redundancy check (CRC)**

 Unlike checksum scheme, which is based on addition, CRC is based on binary division.

 In CRC, a sequence of redundant 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.



# **Example:**

# 

# 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.

# 

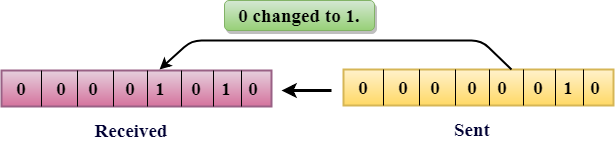
**TYPES ERROR:**

Errors can be classified into two categories:

* Single-Bit Error
* Burst Error

## Single-Bit Error:

The only one bit of a given data unit is changed from 1 to 0 or from 0 to 1.



In the above figure, the message which is sent is corrupted as single-bit, i.e., 0 bit is changed to 1.

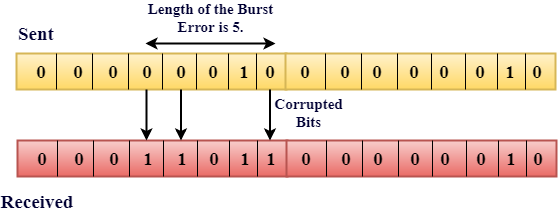
**Single-Bit Error** does not appear more likely in Serial Data Transmission. For example, Sender sends the data at 10 Mbps, this means that the bit lasts only for 1 ?s and for a single-bit error to occurred, a noise must be more than 1 ?s.

Single-Bit Error mainly occurs in Parallel Data Transmission. For example, if eight wires are used to send the eight bits of a byte, if one of the wire is noisy, then single-bit is corrupted per byte.

## Burst Error:

The two or more bits are changed from 0 to 1 or from 1 to 0 is known as Burst Error.

The Burst Error is determined from the first corrupted bit to the last corrupted bit.



The duration of noise in Burst Error is more than the duration of noise in Single-Bit.

Burst Errors are most likely to occurr in Serial Data Transmission.

The number of affected bits depends on the duration of the noise and data rate.

**Forward error correction:**

Forward error correction (FEC) is an error correction technique to detect and correct a limited number of errors in transmitted data without the need for retransmission.

In this method, the sender sends a redundant error-correcting code along with the data frame. The receiver performs necessary checks based upon the additional redundant bits. If it finds that the data is free from errors, it executes error-correcting code that generates the actual frame. It then removes the redundant bits before passing the message to the upper layers.

## **Advantages and Disadvantages**

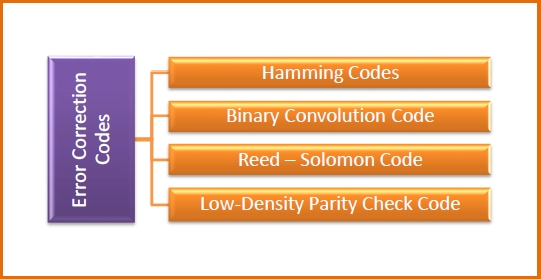
* Because FEC does not require handshaking between the source and the destination, it can be used for broadcasting of data to many destinations simultaneously from a single source.
* Another advantage is that FEC saves bandwidth required for retransmission. So, it is used in real time systems.
* Its main limitation is that if there are too many errors, the frames need to be retransmitted.

## **Error Correction Codes for FEC**

Error correcting codes for forward error corrections can be broadly categorized into two types, namely, block codes and convolution codes.

* **Block codes** − The message is divided into fixed-sized blocks of bits to which redundant bits are added for error correction.
* **Convolutional codes** − The message comprises of data streams of arbitrary length and parity symbols are generated by the sliding application of a Boolean function to the data stream.

There are four popularly used error correction codes.



**Hamming Codes** − It is a block code that is capable of detecting up to two simultaneous bit errors and correcting single-bit errors.

**Binary Convolution Code** − Here, an encoder processes an input sequence of bits of arbitrary length and generates a sequence of output bits.

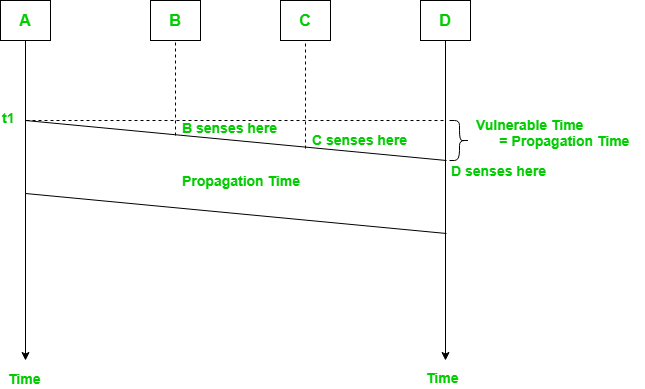
**Reed - Solomon Code** − They are block codes that are capable of correcting burst errors in the received data block.

**Low-Density Parity Check Code** − It is a block code specified by a parity-check matrix containing a low density of 1s. They are suitable for large block sizes in very noisy channels

# **Carrier Sense Multiple Access (CSMA)**

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 –**



The persistence methods can be applied to help the station take action when the channel is busy/idle.

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

Carrier Sense Multiple Access with Collision Detection (CSMA/CD) is a network protocol for carrier transmission that operates in the Medium Access Control (MAC) layer. It senses or listens whether the shared channel for transmission is busy or not, and defers transmissions until the channel is free. The collision detection technology detects collisions by sensing transmissions from other stations. On detection of a collision, the station stops transmitting, sends a jam signal, and then waits for a random time interval before retransmission.

The algorithm of CSMA/CD 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 until the channel becomes idle.
* If the channel is idle, the station starts transmitting and continually monitors the channel to detect collision.
* If a collision is detected, the station starts the collision resolution algorithm.
* The station resets the retransmission counters and completes frame transmission.

The algorithm of Collision Resolution is:

* The station continues transmission of the current frame for a specified time along with a jam signal, to ensure that all the other stations detect collision.
* The station increments the retransmission counter.
* If the maximum number of retransmission attempts is reached, then the station aborts transmission.
* Otherwise, the station waits for a backoff period which is generally a function of the number of collisions and restart main algorithm

**Example:**

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.

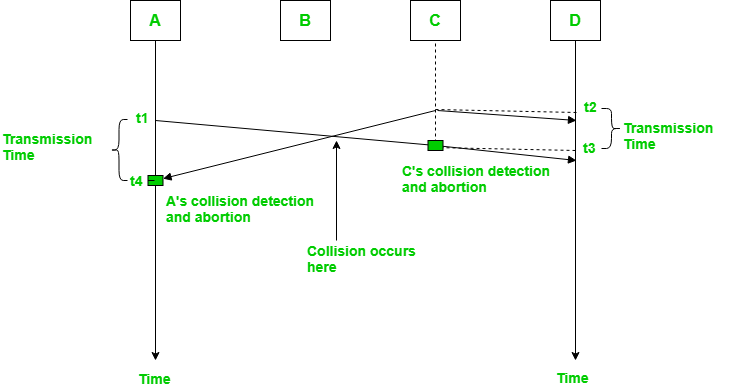
In the diagram, A starts send the first bit of its frame at t1 and since C sees the channel idle at t2, starts sending its frame at t2. C detects A’s frame at t3 and aborts transmission. A detects C’s frame at t4 and aborts its transmission. Transmission time for C’s frame is therefore t3-t2  and for A’s frame is t4-t1.

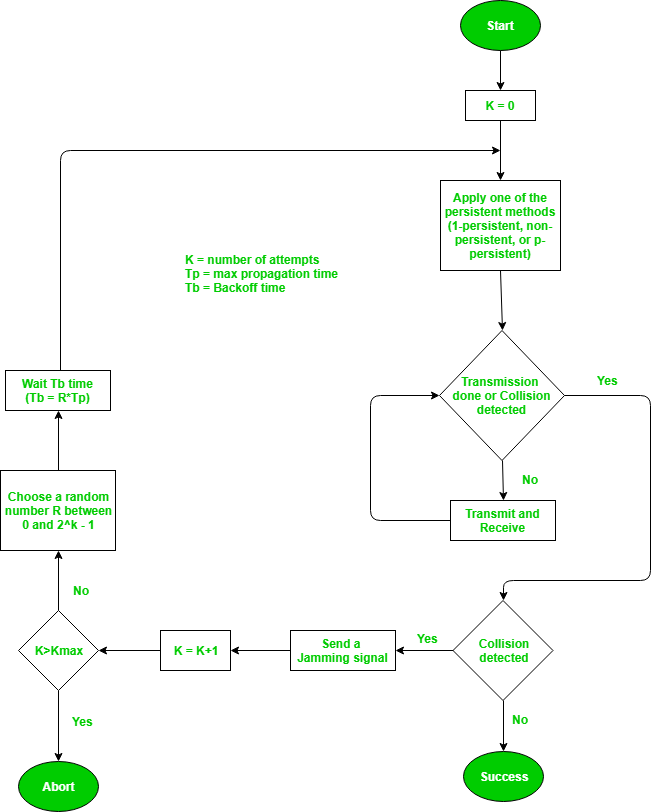
So, the **frame transmission time (Tfr) should be at least twice the maximum propagation time (Tp)**. This can be deduced when the two stations involved in collision are maximum distance apart.

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

* For 1-persistent method throughput is 50% when G=1.

For non-persistent method throughput can go upto 90%





## **Hamming Distance**

Hamming distance is a metric for comparing two binary data strings. While comparing two binary strings of equal length, Hamming distance is the number of bit positions in which the two

bits are different.

The Hamming distance between two strings, a and b is denoted as d(a,b).

It is used for error detection or error correction when data is transmitted over computer networks. It is also using in coding theory for comparing equal length data words.

## ***Calculation of Hamming Distance***

In order to calculate the Hamming distance between two strings, and , we perform their XOR operation, (a⊕ b), and then count the total number of 1s in the resultant string.

## ***Example 1:***

Suppose there are two strings 1101 1001 and 1001 1101.

11011001 ⊕ 10011101 = 01000100. Since, this contains two 1s, the Hamming distance, d(11011001, 10011101) = 2.

## ***Minimum Hamming Distance***

In a set of strings of equal lengths, the minimum Hamming distance is the smallest Hamming distance between all possible pairs of strings in that set.

## ***Example 2:***

Suppose there are four strings 010, 011, 101 and 111.

010 ⊕ 011 = 001, d(010, 011) = 1.

010 ⊕ 101 = 111, d(010, 101) = 3.

010 ⊕ 111 = 101, d(010, 111) = 2.

011 ⊕ 101 = 110, d(011, 101) = 2.

011 ⊕ 111 = 100, d(011, 111) = 1.

101 ⊕ 111 = 010, d(011, 111) = 1.

Hence, the Minimum Hamming Distance, *dmin* = 1.

**Detection Versus Correction**

The correction of errors is more difficult than the detection. In error detection, we are looking only to see if any error has occurred. The answer is a simple yes or no. We are not even interested in the number of errors. A single-bit error is the same for us as a burst error. In error correction, we need to know the exact number of bits that are corrupted and more importantly, their location in the message. The number of the errors and the size of the message are important factors. If we need to correct one single error in an 8-bit data unit, we need to consider eight possible error locations; if we need to correct two errors in a data unit of the same size, we need to consider 28 possibilities. You can imagine the receiver's difficulty in finding 10 errors in a data unit of 1000 bits.

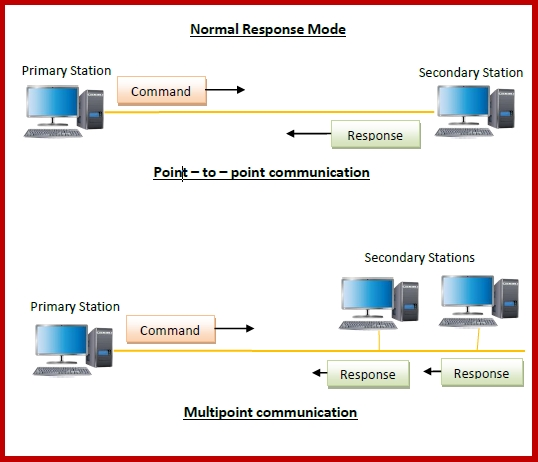
# **High-level Data Link Control (HDLC)**

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

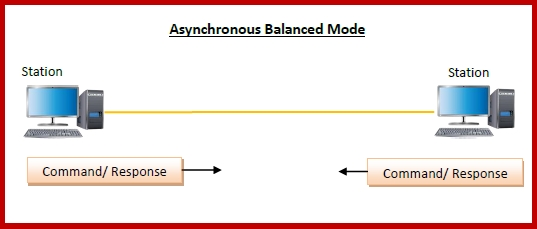
## ***Transfer Modes***

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

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



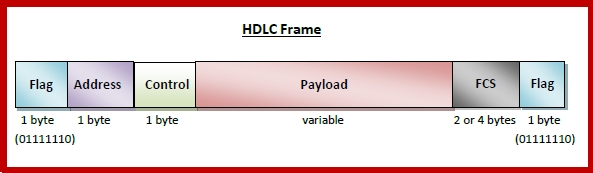
**Asynchronous Balanced Mode (ABM)** − Here, the configuration is balanced, i.e. each station can both send commands and respond to commands. It is used for only point - to - point communications.



## ***HDLC Frame***

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

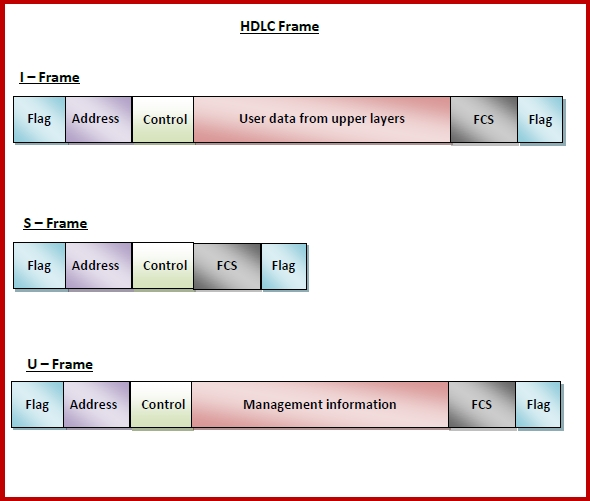
* **Flag** − It is an 8-bit sequence that marks the beginning and the end of the frame. The bit pattern of the flag is 01111110.
* **Address** − It contains the address of the receiver. If the frame is sent by the primary station, it contains the address(es) of the secondary station(s). If it is sent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



## ***Types of HDLC Frames***

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

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



# **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 is widely used in broadband communications having heavy loads and high speeds. Since it is a data link layer protocol, data is transmitted in frames. It is also known as RFC 1661.

## ***Services Provided by PPP***

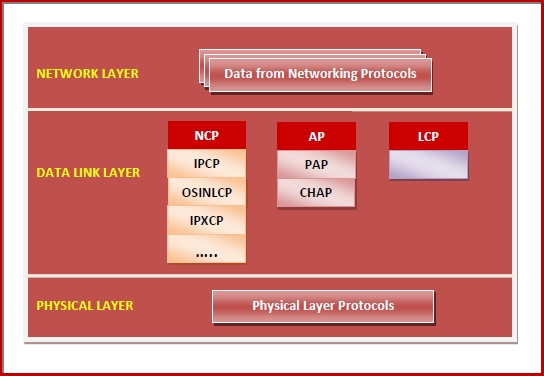
The main services provided by Point - to - Point Protocol are −

* Defining the frame format of the data to be transmitted.
* Defining the procedure of establishing link between two points and exchange of data.
* Stating the method of encapsulation of network layer data in the frame.
* Stating authentication rules of the communicating devices.
* Providing address for network communication.
* Providing connections over multiple links.
* Supporting a variety of network layer protocols by providing a range os services.

## ***Components of PPP***

Point - to - Point Protocol is a layered protocol having three components −

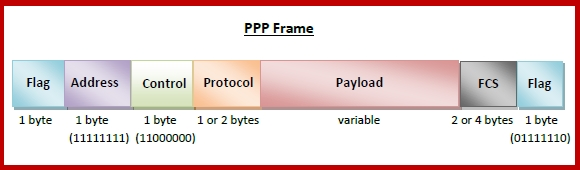
* **Encapsulation Component** − It encapsulates the datagram so that it can be transmitted over the specified physical layer.
* **Link Control Protocol (LCP)** − It is responsible for establishing, configuring, testing, maintaining and terminating links for transmission. It also imparts negotiation for set up of options and use of features by the two endpoints of the links.
* **Authentication Protocols (AP)** − These protocols authenticate endpoints for use of services. The two authentication protocols of PPP are:
  + Password Authentication Protocol (PAP)
  + Challenge Handshake Authentication Protocol (CHAP)
* **Network Control Protocols (NCPs)** − These protocols are used for negotiating the parameters and facilities for the network layer. For every higher-layer protocol supported by PPP, one NCP is there. Some of the NCPs of PPP are:
  + Internet Protocol Control Protocol (IPCP)
  + OSI Network Layer Control Protocol (OSINLCP)
  + Internetwork Packet Exchange Control Protocol (IPXCP)
  + DECnet Phase IV Control Protocol (DNCP)
  + NetBIOS Frames Control Protocol (NBFCP)
  + IPv6 Control Protocol (IPV6CP)



## ***PPP Frame***

PPP is a byte - oriented protocol where each field of the frame is composed of one or more bytes. The fields of a PPP frame are −

* **Flag** − 1 byte that marks the beginning and the end of the frame. The bit pattern of the flag is 01111110.
* **Address** − 1 byte which is set to 11111111 in case of broadcast.
* **Control** − 1 byte set to a constant value of 11000000.
* **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)



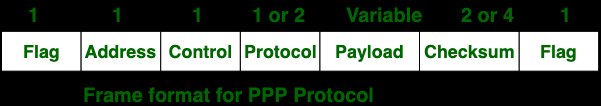
**Byte Stuffing in PPP Frame** − Byte stuffing is used is PPP payload field whenever the flag sequence appears in the message, so that the receiver does not consider it as the end of the frame. The escape byte, 01111101, is stuffed before every byte that contains the same byte as the flag byte or the escape byte. The receiver on receiving the message removes the escape byte before passing it onto the network layer.

# **Difference Between High-level Data Link Control (HDLC) and Point-to-Point Protocol (PPP)**

The main difference between **High-level Data Link Control (HDLC)** and **Point-to-Point Protocol (PPP)** is that High-level Data Link Control is the bit-oriented protocol, on the other hand, Point-to-Point Protocol is the byte-oriented protocol.

Another difference between HDLC and PPP is that HDLC is implemented by Point-to-point configuration and also multi-point configurations on the other hand While PPP is implemented by Point-to-Point configuration only.





| **S.NO** | **HDLC** | **PPP** |
| --- | --- | --- |
| 1. | HDLC stands for High-level Data Link Control. | PPP stands for Point-to-Point Protocol. |
| 2. | HDLC is a bit oriented protocol. | PPP is a byte oriented protocol. |
| 3. | HDLC is implemented by Point-to-point configuration and also multi-point configurations. | PPP is implemented by Point-to-Point configuration only. |
| 4. | Dynamic addressing is not offered by HDLC. | While in this Dynamic addressing is offered. |
| 5. | HDLC is used in synchronous media. | PPP is used in synchronous media as well as asynchronous media. |
| 6. | HDLC is not compatible with non-Cisco devices. | PPP is compatible with non-Cisco devices. |