PERFORMANCE OF NETWORK

Performance of a network depends upon 3 factors

1. Bandwidth (must be high)
2. Throughput (must be high)
3. Delay (must be low)

Bandwidth

Bandwidth can be measured in bits per second or hertz Bandwidth in hertz means that maximum frequency minus minimum frequency (as studied earlier)

Bandwidth In bits per second means bit rate itself i.e., number of bits transmitted per second and it is the mostly used definition in networking for bandwidth

Throughput

The throughput is a measure of how fast we can actually send data through a network.

The bandwidth is a potential measurement of a link but the throughput is an actual measurement of how fast we can send data. For example, we may have a link with a bandwidth of 1 Mbps, but the devices connected to the end of the link may handle only 200 kbps. This means that we cannot send more than 200 kbps through this link.

Latency (Delay)

The latency or delay defines how long it takes for an entire message to completely arrive at the destination from the time the first bit is sent out from the source. We can say that latency is made of four components: propagation time, transmission time, ANA queuing time and processing delay.

Latency=Propagation Time +Transmission Time +Queuing Time + Processing Delay

Propagation Time

Propagation time measures the time required for a bit to travel from the source to the destination. The propagation time is calculated by dividing the distance by the propagation speed.

$$Propagation\ Time = \frac{Distance}{Propagation\ Speed}$$

Transmission Time

Time between the first bit leaving the sender and the last bit arriving at the receiver is known

$$Transmission\ Time = \frac{Message\ Size}{Bandwidth}$$

Queuing Time

The third component in latency is the queuing time, the time needed for each intermediate or end device to hold the message before it can be processed. The queuing time is not a fixed factor, it changes with the load imposed on the network. When there is heavy traffic on the network, the queuing time increases. An intermediate device, such as a router, queues the arrived messages and processes them one by one. If there are many messages, each message will have to wait.

Processing Delay

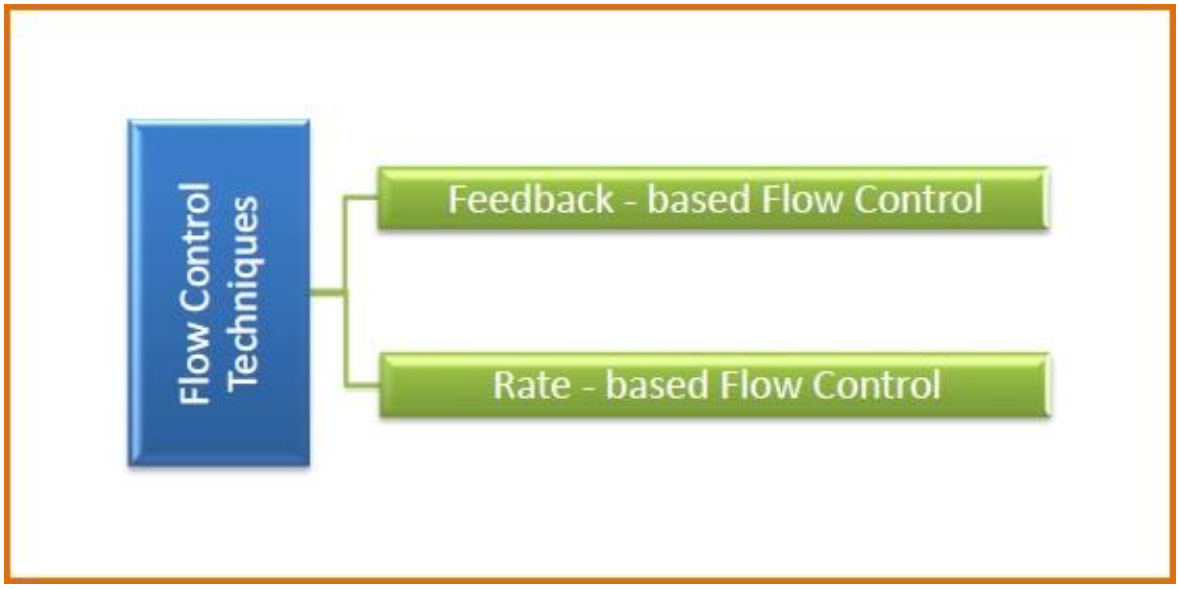
Amount of time needed by intermediate devices to process the packet i.e., amount of time the router needs to route the packet

Flow control in Data Link Layer

Flow control is a technique that allows two stations working at different speeds to communicate with each other. It is a set of measures taken to regulate the amount of data that a sender sends so that a fast sender does not overwhelm a slow receiver. In data link layer, flow control restricts the number of frames the sender can send before it waits for an acknowledgment from the receiver.

Approaches of Flow Control

Flow control can be broadly classified into two categories –

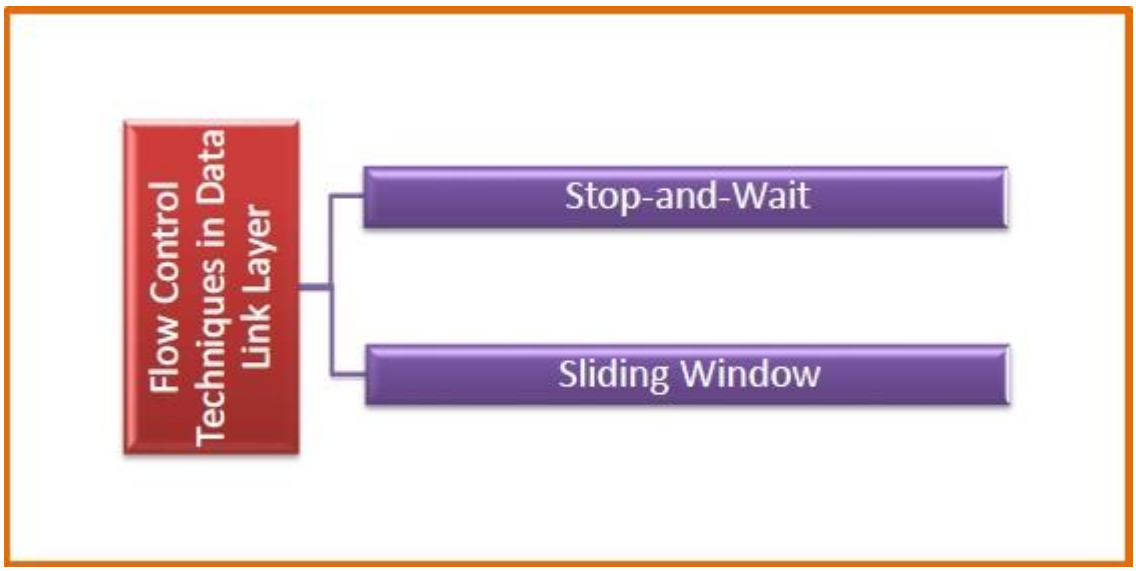


- Feedback based Flow Control In these protocols; the sender sends frames after it has received acknowledgments from the user. This is used in the data link layer.

- Rate based Flow Control These protocols have built in mechanisms to restrict the rate of transmission of data without requiring acknowledgment from the receiver. This is used in the network layer and the transport layer.

Flow Control Techniques in Data Link Layer

Data link layer uses feedback-based flow control mechanisms. There are two main techniques –



Stop and Wait

This protocol involves the following transitions −

- The sender sends a frame and waits for acknowledgment.
- Once the receiver receives the frame, it sends an acknowledgment frame back to the sender.
- On receiving the acknowledgment frame, the sender understands that the receiver is ready to accept the next frame. So, its sender the next frame in queue.

Sliding Window

This protocol improves the efficiency of stop and wait protocol by allowing multiple frames to be transmitted before receiving an acknowledgment.

The working principle of this protocol can be described as follows −

- Both the sender and the receiver have finite sized buffers called windows. The sender and the receiver agree upon the number of frames to be sent based upon the buffer size.
- The sender sends multiple frames in a sequence, without waiting for acknowledgment. When its sending window is filled, it waits for acknowledgment. On receiving acknowledgment, it advances the window and transmits the next frames, according to the number of acknowledgments received.

Error control in Data Link Layer

Error control in data link layer is the process of detecting and correcting data frames that have been corrupted or lost during transmission.

In case of lost or corrupted frames, the receiver does not receive the correct data-frame and sender is ignorant about the loss. Data link layer follows a technique to detect transit errors and take necessary actions, which is retransmission of frames whenever error is detected or frame is lost. The process is called Automatic Repeat Request (ARQ).

Phases in Error Control

The error control mechanism in data link layer involves the following phases −

Detection of Error − Transmission error, if any, is detected by either the sender or the receiver.

Acknowledgment − acknowledgment may be positive or negative.

Positive ACK − On receiving a correct frame, the receiver sends a positive acknowledge.

Negative ACK − On receiving a damaged frame or a duplicate frame, the receiver sends a negative acknowledgment back to the sender.

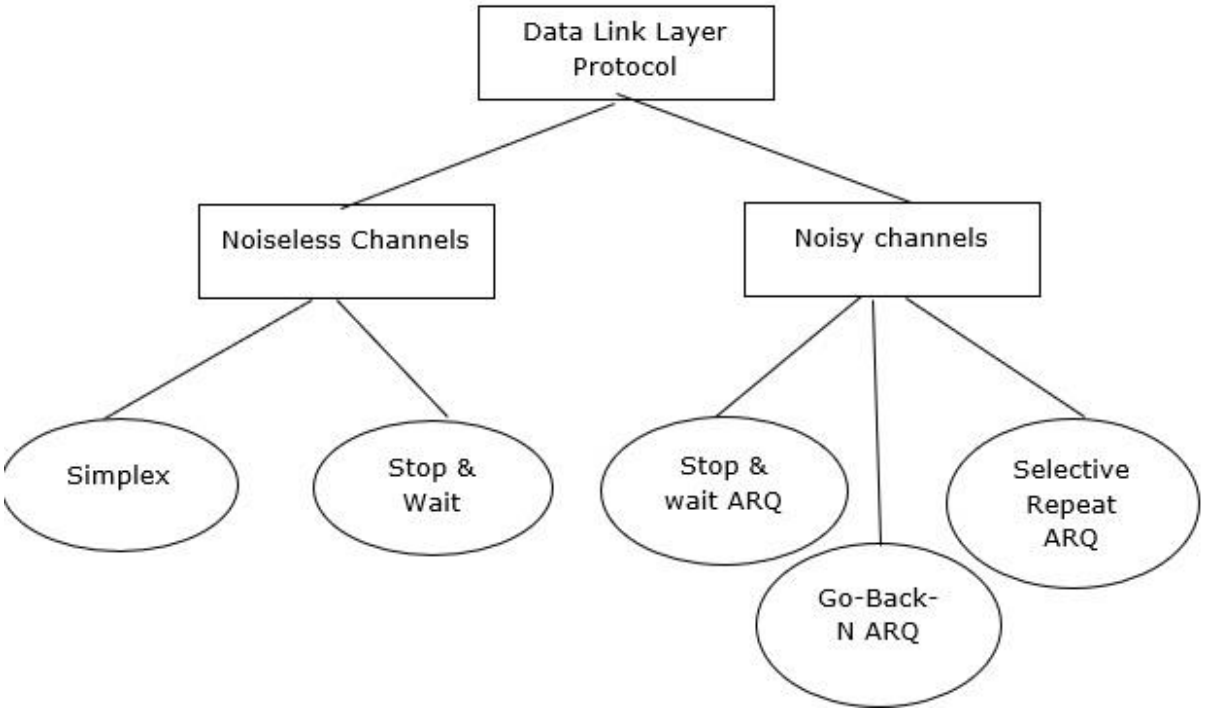
Retransmission − The sender maintains a clock and sets a timeout period. If an acknowledgment of a data-frame previously transmitted does not arrive before the timeout, or a negative acknowledgment is received, the sender retransmits the frame.

Noiseless Channel

A protocol is a set of rules used by two devices to communicate. These sets of rules are usually decided by headers (fixed headers determined by the protocol). These headers specify the content of the message and the way this message is processed. To detect the error, the header must be the address of the destination, the address of the source, the checksum of the message.

Categorization of protocol:

The exploration of protocols is split into those that can be applied for noiseless(error-free) channels and those that can be used for noisy(error-causing) channels. The first category of protocols cannot be used in actual life, but they serve as a basis for protocols for noise channels.



Noiseless Channel:

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.

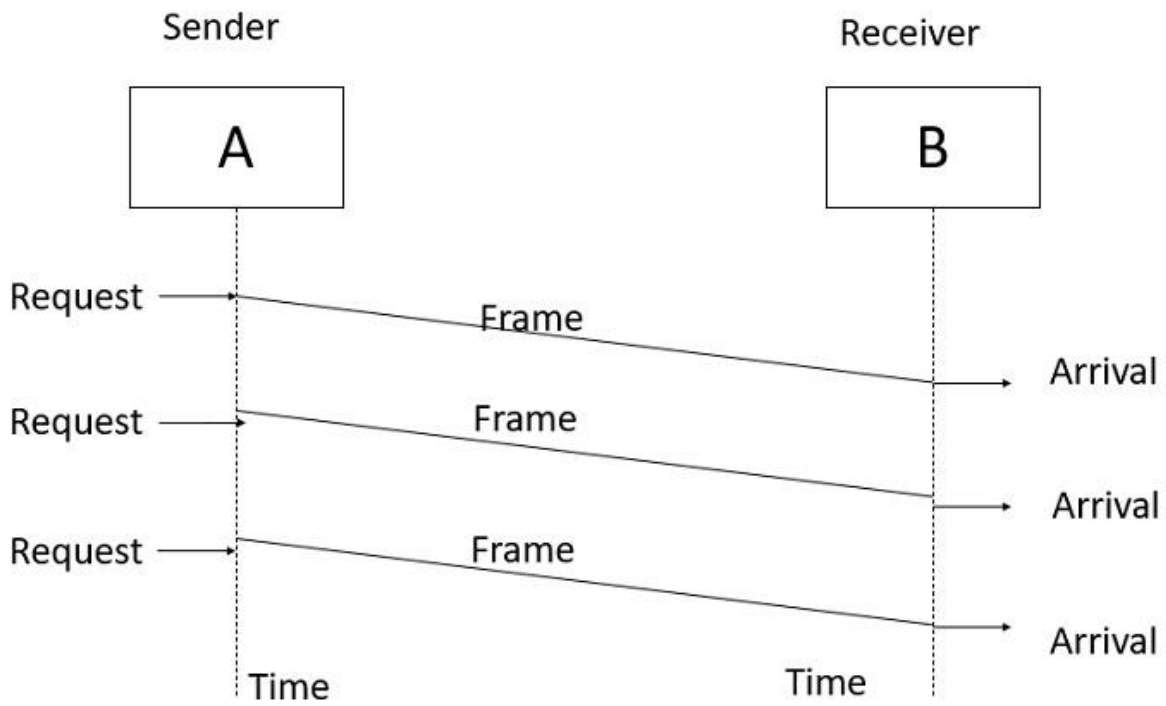
1. Simplex channel
2. Stop & wait channel

Simplest Protocol

Step 1 − Simplest protocol that does not have flow or error control.

Step 2 − It is a unidirectional protocol where data frames are traveling in one direction that is from the sender to receiver.

Step 3 − Let us assume that the receiver can 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.



Stop-and-Wait Protocol

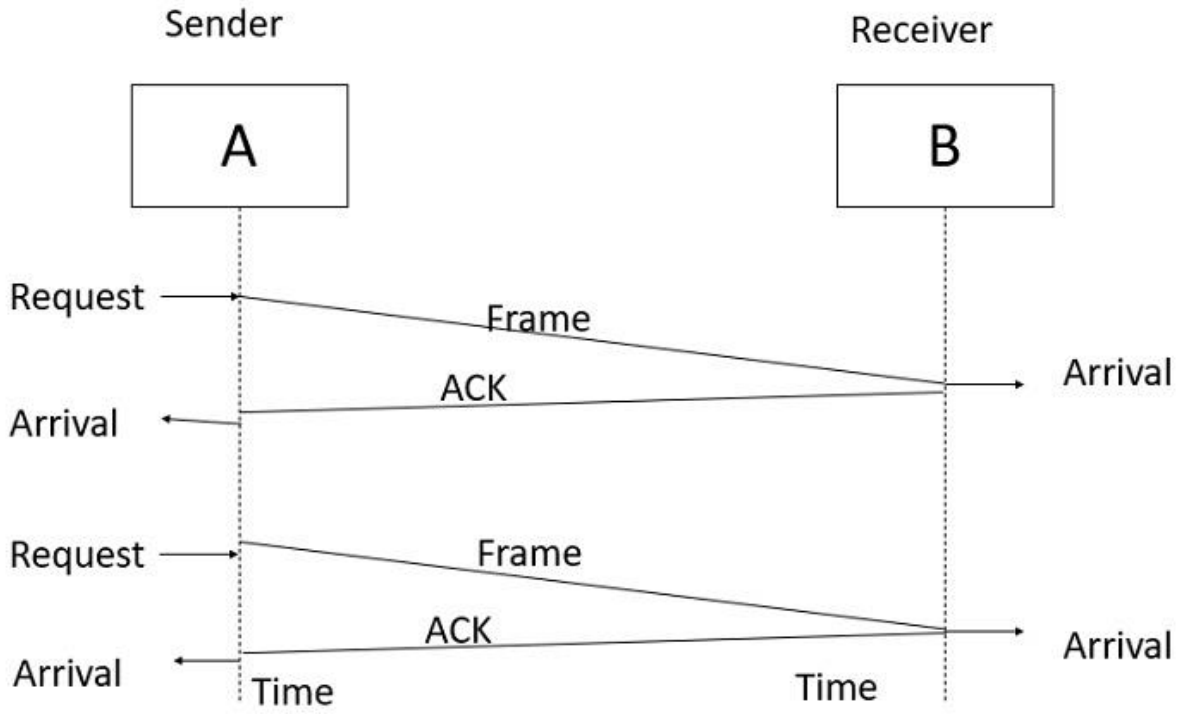
Step 1 − If the data frames that arrive at the receiver side are faster than they can be processed, the frames must be stored until their use.

Step 2 − Generally, the receiver does not have enough storage space, especially if it is receiving data from many sources. This may result in either discarding of frames or denial of service.

Step 3 − To prevent the receiver from becoming overwhelmed with frames, the sender must slow down. There must be ACK from the receiver to the sender.

Step 4 − In this protocol the sender sends one frame, stops until it receives confirmation from the receiver, and then sends the next frame.

Step 5 − We still have unidirectional communication for data frames, but auxiliary ACK frames travel from the other direction. We add flow control to the previous protocol.



Noisy Channels

Noisy channel protocols are commonly known as communication protocols. These protocols are designed to ensure a reliable data transmission over a channel in which there is a high probability of errors or data loss. In computer networks, these protocols are typically used to overcome the effects of channel noise during data transmission which can result from electromagnetic interference, attenuation etc. that cause signal degradation. So, noisy channel protocols perform an important role to ensure reliable communication in computer networks, especially in environments where channel noise is a significant concern.

There are three types of requests for the noisy channels, which are as follows −

1. Stop & wait Automatic Repeat Request.
2. Go-Back-N Automatic Repeat Request.
3. Selective Repeat Automatic Repeat Request.

Requirements for Error Control

There are some requirements for error control mechanisms and they are as follows −

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

Positive ACK − Whenever a receiver receives a correct frame, it should acknowledge it.

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

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

Stop and Wait Automatic Repeat Request

Step 1 − In a noisy channel, if a frame is damaged during transmission, the receiver will detect with the help of the checksum.

Step 2 − If a damaged frame is received, it will be discarded, and the transmitter will retransmit the same frame after receiving a proper acknowledgement.

Step 3 − If the acknowledgement frame gets lost and the data link layer on 'A' eventually times out. Not having received an ACK, it assumes that its data frame was lost or damaged and sends the frame containing packet 1 again. This duplicate frame also arrives at the data link layer on 'B', thus part of the file will be duplicated and protocol is said to be failed.

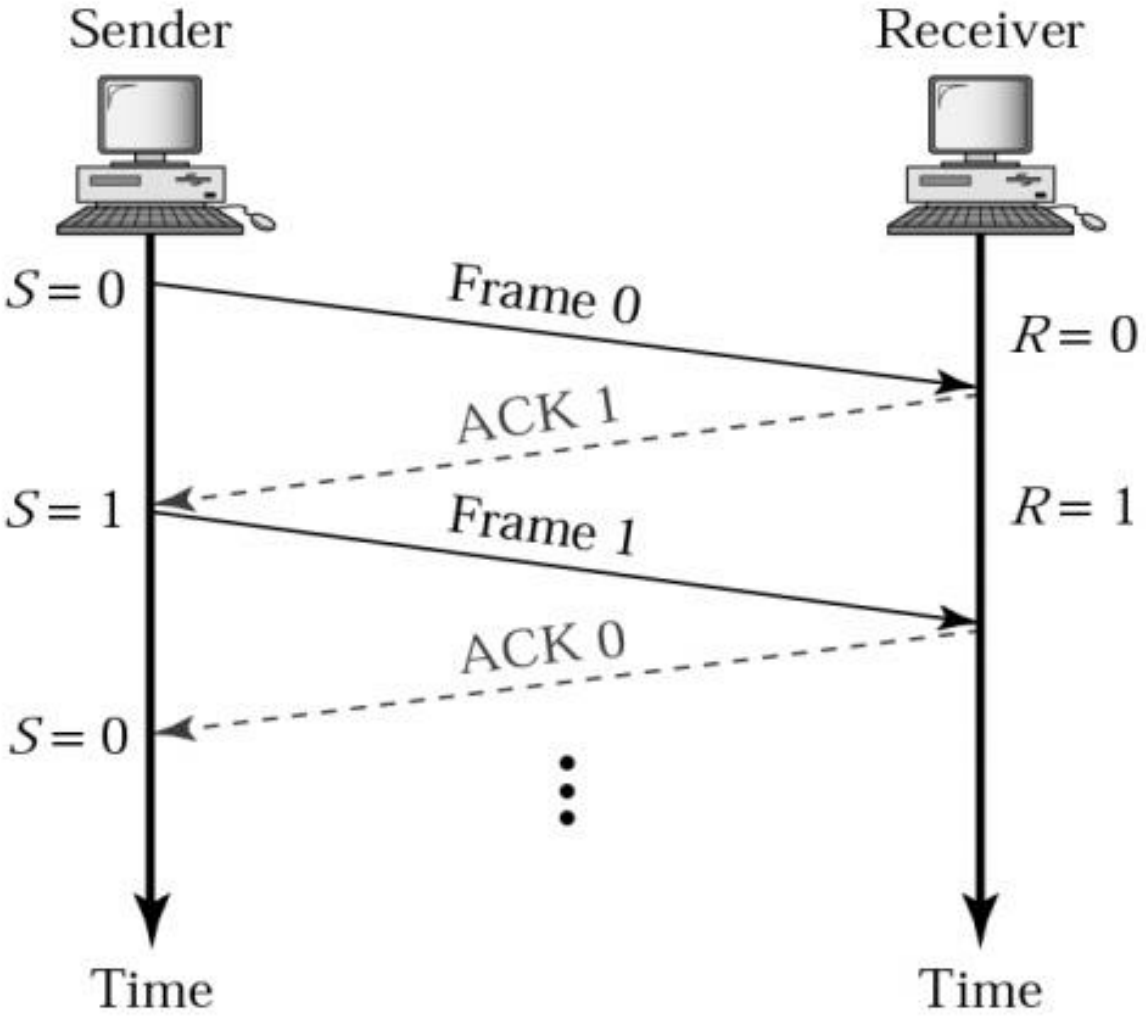
Step 4 − To solve this problem, assign a sequence number in the header of the message.

Step 5 − The receiver checks the sequence number to determine if the message is a duplicate since only the message is transmitted at any time.

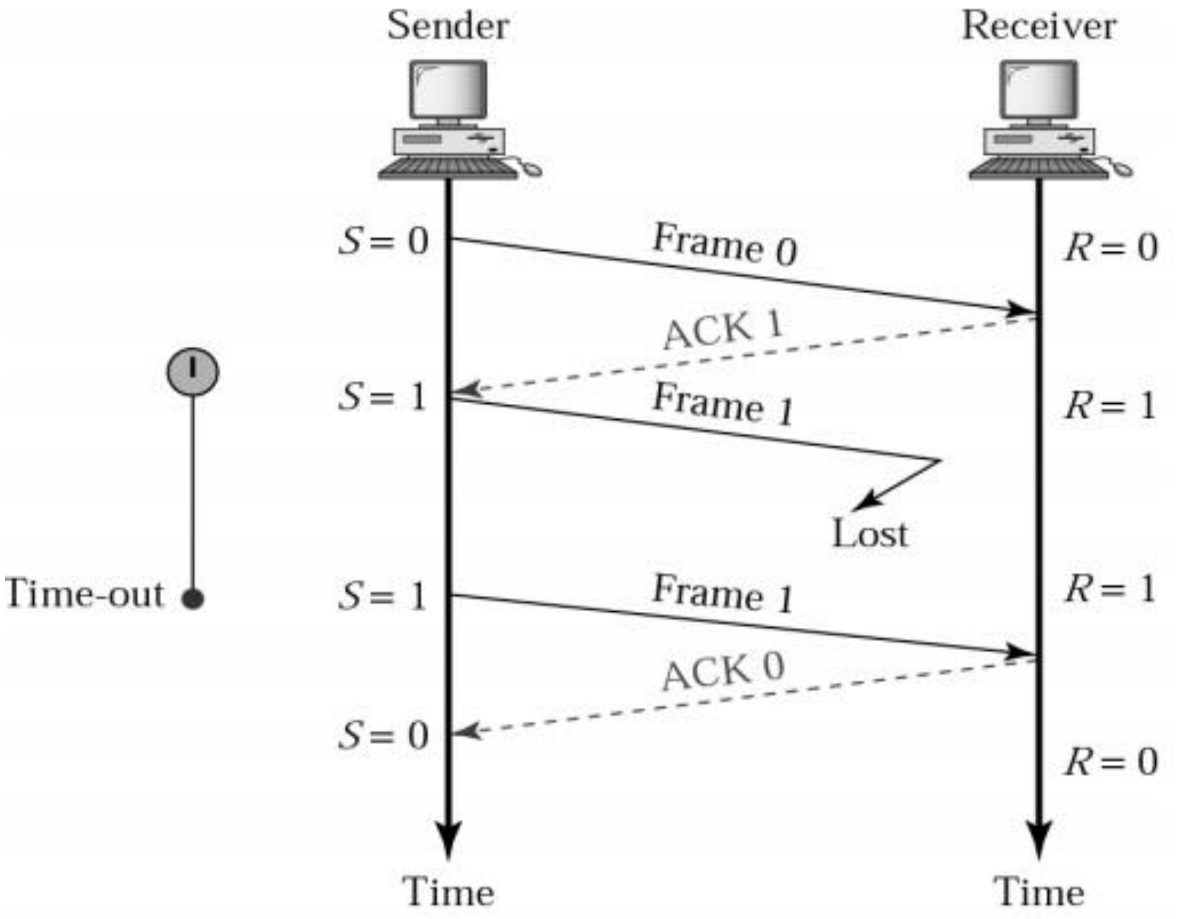
Step 6 − The sending and receiving station needs only a 1-bit alternating sequence of '0' or '1' to maintain the relationship of the transmitted message and its ACK/ NAK.

Step 7 − A modulo-2 numbering scheme is used where the frames are alternatively labelled with '0' or '1' and positive acknowledgements are of the form ACK 0 and ACK 1.

Normal operation of Stop & Wait ARQ is given below –



Stop & Wait ARQ with Lost frame is as follows –


Sender
Receiver
S = 0
Frame 0
R = 0
ACK 1
S = 1
Frame 1
R = 1
Lost
Time-out
S = 1
Frame 1
R = 1
ACK 0
S = 0
R = 0
Time
Time

Go-Back-N ARQ

To improve the transmission efficiency, we need more than one frame to be outstanding to keep the channel busy while the sender is waiting for acknowledgement.

There are two protocols developed for achieving this goal and they are as follows –

1. Go – Back - N – Automatic – Repeat Request
2. Sliding window protocol

Go-Back-N ARQ

Step 1 − In this protocol we can send several frames before receiving acknowledgements.

Step 2 − we keep a copy of these frames until the acknowledgment arrives.

Step 3 − Frames from a sending station are numbered sequentially. However, we need to include the sequence number of each frame in the header; we need to set a limit.

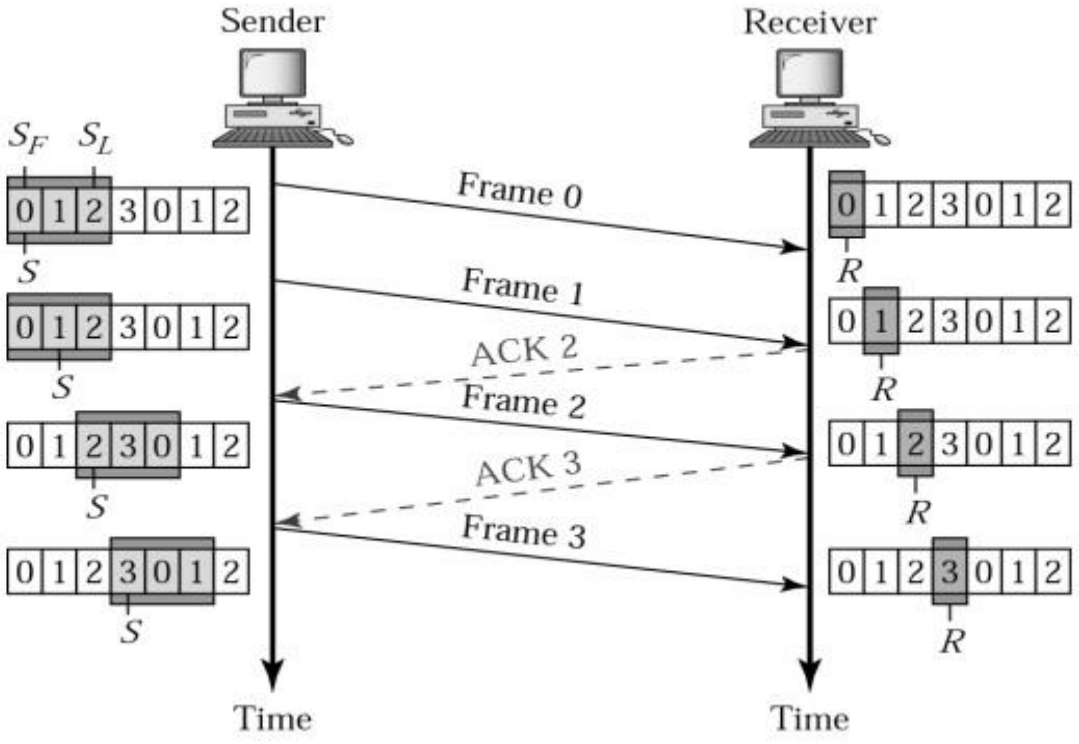
Step 4 − If the header of the frame allows m bits for the sequence number, the sequence numbers range from 0 to 2m-1. We can also repeat the sequence numbers.

Example

For m = 2, the range of sequence numbers is: 0 to 3, i.e.

0,1,2,3, 0,1,2,3,…

The Go-Back-N ARQ is shown below in diagram format –



Selective Repeat ARQ

It is also known as Sliding Window Protocol and used for error detection and control in the data link layer.

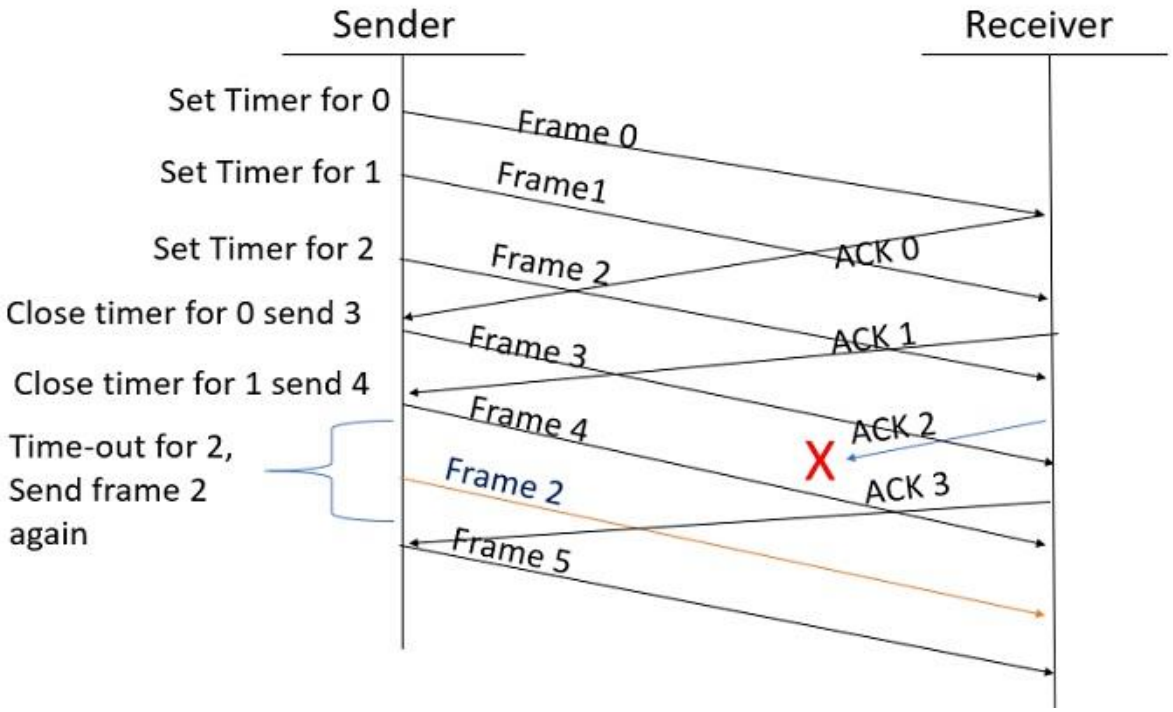
In the selective repeat, the sender sends several frames specified by a window size even without the need to wait for individual acknowledgement from the receiver as in Go-Back-N ARQ. In selective repeat protocol, the retransmitted frame is received out of sequence.

In Selective Repeat ARQ only the lost or error frames are retransmitted, whereas correct frames are received and buffered.

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 will send/retransmit a packet for which NACK is received.

Example

Given below is an example of the Selective Repeat ARQ –



Explanation

Step 1 − Frame 0 sends from sender to receiver and set timer.

Step 2 − Without waiting for acknowledgement from the receiver another frame, Frame1 is sent by sender by setting the timer for it.

Step 3 − In the same way frame2 is also sent to the receiver by setting the timer without waiting for previous acknowledgement.

Step 4 − Whenever sender receives the ACK0 from receiver, within the frame 0 timer then it is closed and sent to the next frame, frame 3.

Step 5 − whenever the sender receives the ACK1 from the receiver, within the frame 1 timer then it is closed and sent to the next frame, frame 4.

Step 6 − If the sender doesn’t receive the ACK2 from the receiver within the time slot, it declares timeout for frame 2 and resends the frame 2 again, because it thought the frame2 may be lost or damaged.

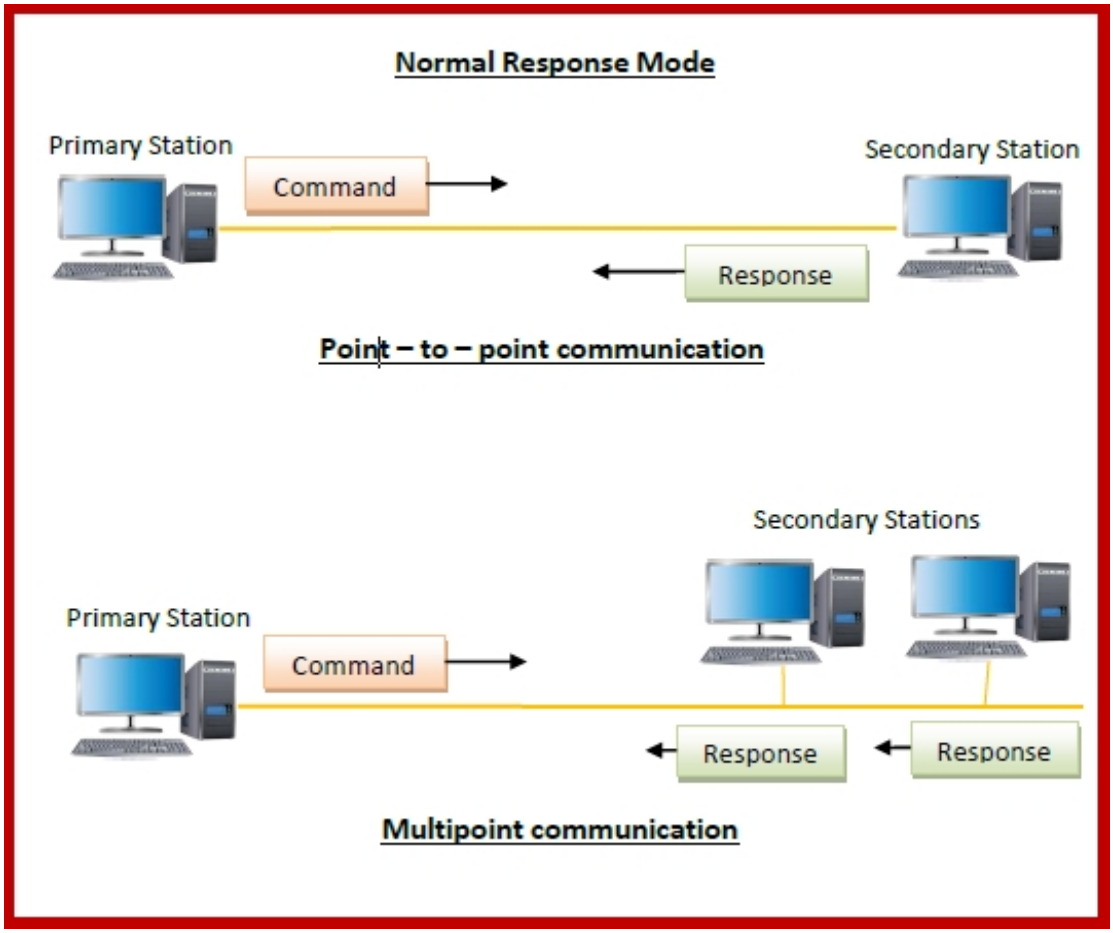
High-level Data Link Control (HDLC)

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.

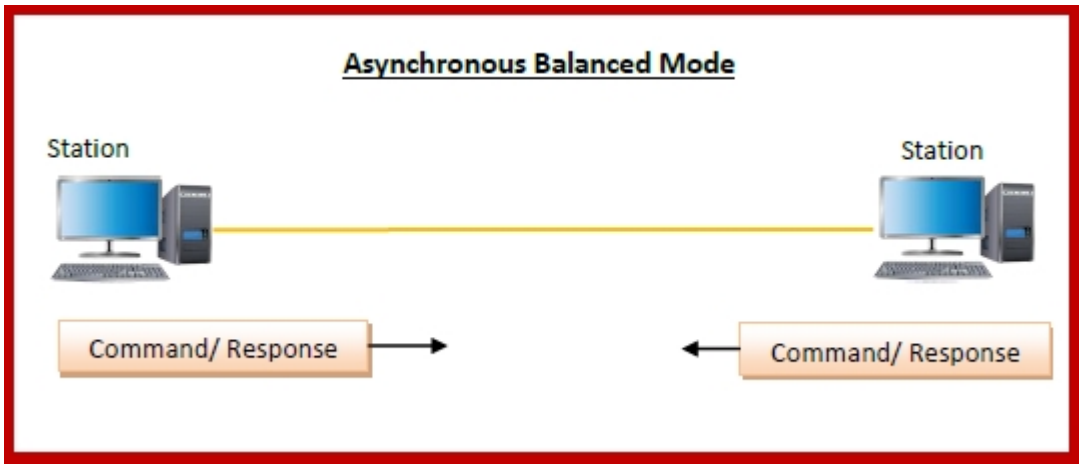
Transfer Modes

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.



Data Link Protocols

The data link protocols operate in the data link layer of the Open System Interconnections (OSI) model, just above the physical layer.

The services provided by the data link protocols may be any of the following −

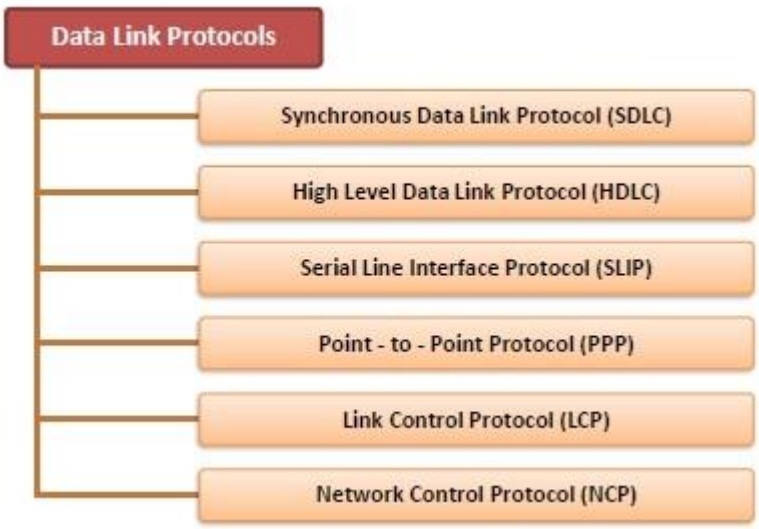
Framing − The stream of bits from the physical layer are divided into data frames whose size ranges from a few hundred to a few thousand bytes. These frames are distributed to different systems, by adding a header to the frame containing the address of the sender and the receiver.

Flow Control − Through flow control techniques, data is transmitted in such a way so that a fast sender does not drown a slow receiver.

Error Detection and/or Correction − These are techniques of detecting and correcting data frames that have been corrupted or lost during transmission.

Multipoint transmission − Access to shared channels and multiple points are regulated in case of broadcasting and LANs.

Common Data Link Protocols



- **Synchronous Data Link Protocol (SDLC)** − SDLC was developed by IBM in the 1970s as part of Systems Network Architecture. It was used to connect remote devices to mainframe computers. It ascertained that data units arrive correctly and with right flow from one network point to the next.

- **High Level Data Link Protocol (HDLC)** − HDLC is based upon SDLC and provides both unreliable service and reliable service. It is a bit – oriented protocol that is applicable for both point – to – point and multipoint communications.

- **Serial Line Interface Protocol (SLIP)** − This is a simple protocol for transmitting data units between an Internet service provider (ISP) and home user over a dial-up link. It does not provide error detection / correction facilities.

- **Point - to - Point Protocol (PPP)** − This 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.

- **Link Control Protocol (LCP)** − It one of PPP protocols that 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.

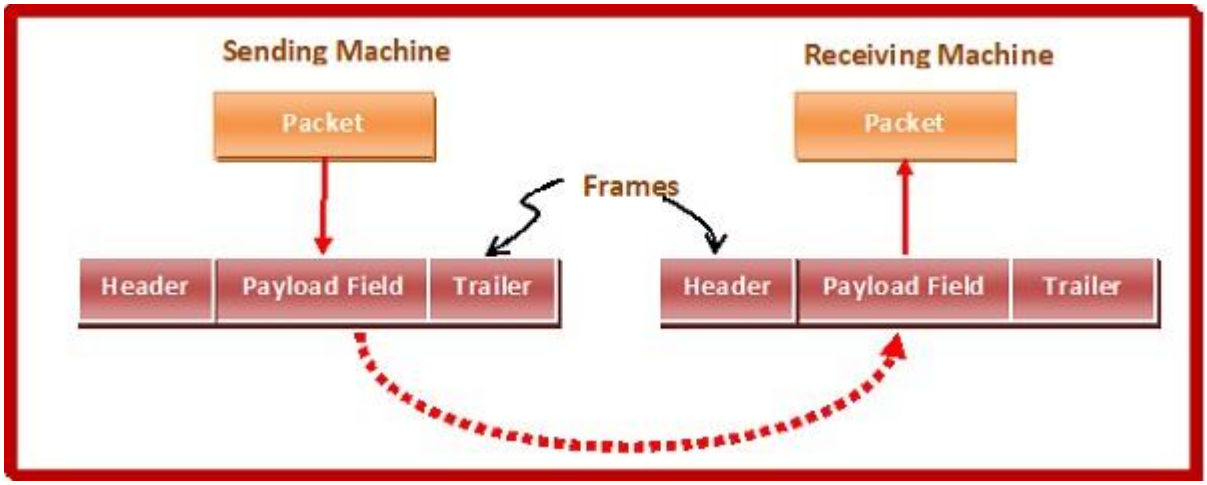
- **Network Control Protocol (NCP)** − These protocols are used for negotiating the parameters and facilities for the network layer. For every higher-layer protocol supported by PPP, one NCP is there.

Framing

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

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

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



Parts of a Frame

A frame has the following parts −

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

Types of Framing

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

Fixed-sized Framing

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

Example − ATM cells.

Variable – Sized Framing

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

It is used in local area networks.

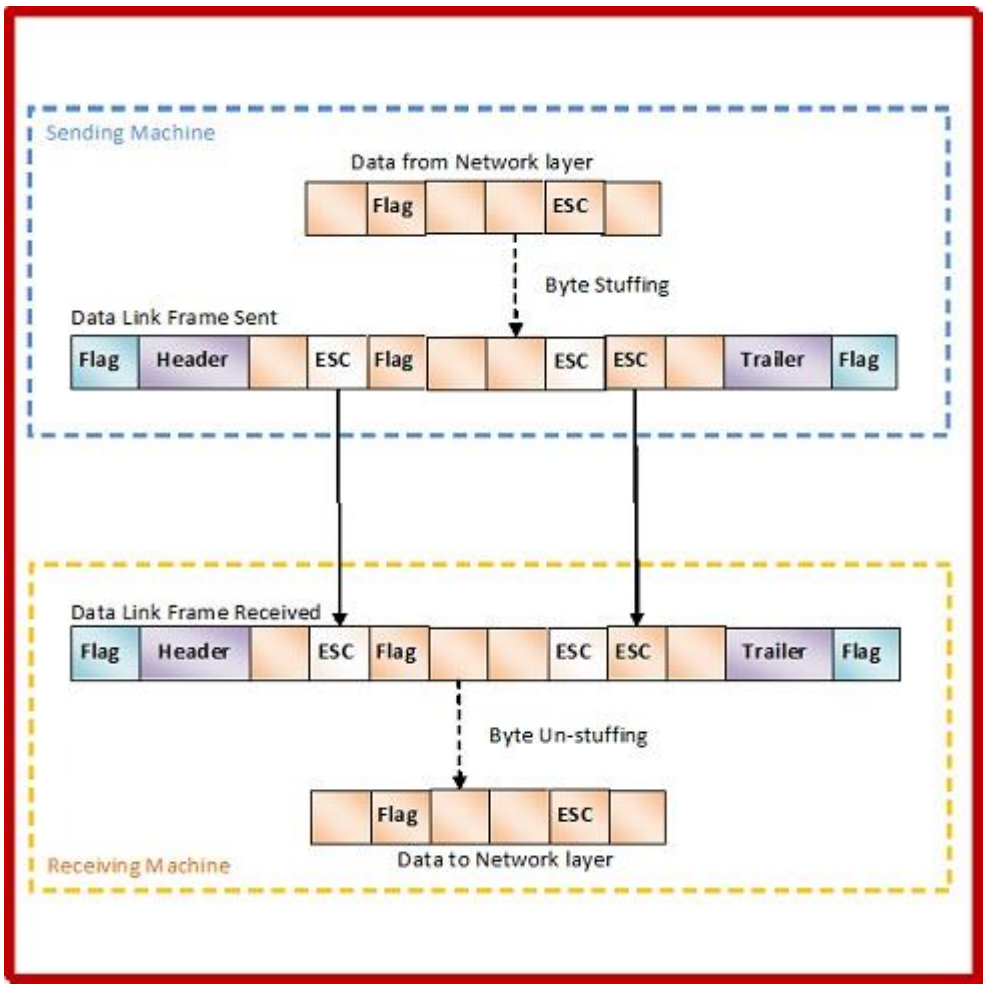
Two ways to define frame delimiters in variable sized framing are −

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

Byte Stuffing Mechanism

If the pattern of the flag byte is present in the message byte, there should be a strategy so that the receiver does not consider the pattern as the end of the frame. In character – oriented protocol, the mechanism adopted is byte stuffing.

In byte stuffing, a special byte called the escape character (ESC) is stuffed before every byte in the message with the same pattern as the flag byte. If the ESC sequence is found in the message byte, then another ESC byte is stuffed before it.



Bit Stuffing Mechanism

In a data link frame, the delimiting flag sequence generally contains six or more consecutive 1s. In order to differentiate the message from the flag in case of the same sequence, a single bit is stuffed in the message. Whenever a 0 bit is followed by five consecutive 1bits in the message, an extra 0 bit is stuffed at the end of the five 1s.

When the receiver receives the message, it removes the stuffed 0s after each sequence of five 1s. The un-stuffed message is then sent to the upper layers.