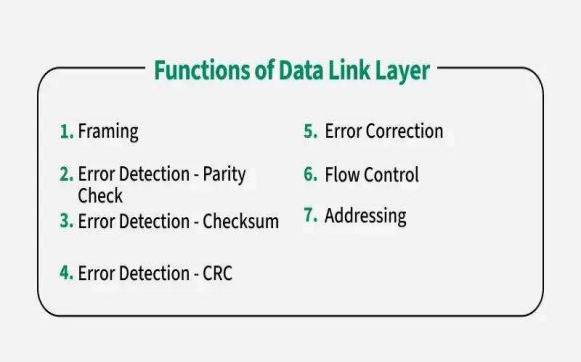
Unit 4

Introduction of data link layer, Various protocols for data link layer, MAC-Layer, HDLC and CSDA/CD

Data Link Layer

- In the OSI model, the data link layer is a 4th layer from the top and 2nd layer from the bottom.
- The communication channel that connects the adjacent nodes is known as links, and in order to move the datagram from source to the destination, the datagram must be moved across an individual link.
- The main responsibility of the Data Link Layer is to transfer the datagram across an individual link.
- The Data link layer protocol defines the format of the packet exchanged across the nodes as well as the actions such as Error detection, retransmission, flow control, and random access.
- The Data Link Layer protocols are Ethernet, token ring, FDDI and PPP.
- An important characteristic of a Data Link Layer is that datagram can be handled by different link layer protocols on different links in a path. For example, the datagram is handled by Ethernet on the first link, PPP on the second link.



1.Framing

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

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

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.

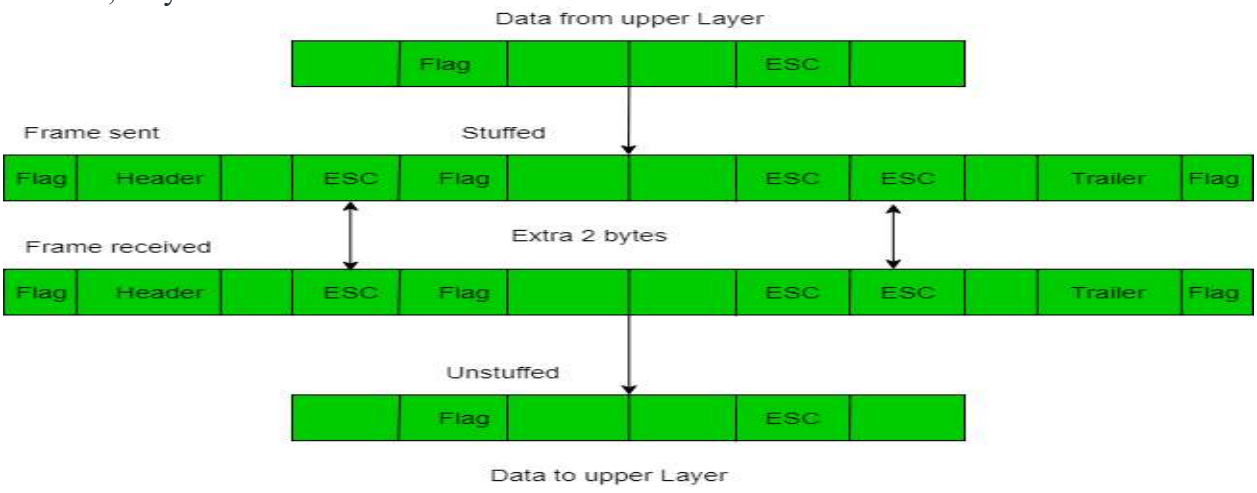
Types of framing
There are two types of framing:

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

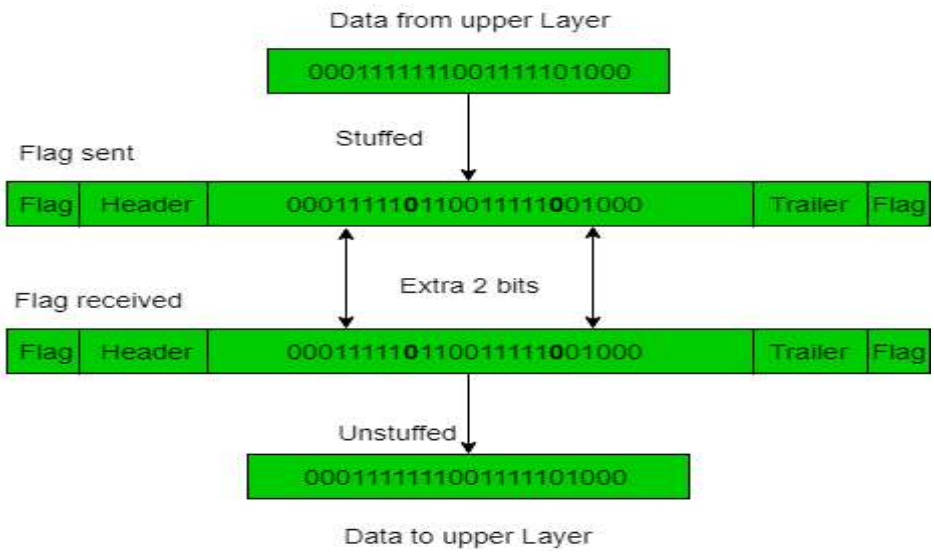
- **Drawback:** It suffers from internal fragmentation if the data size is less than the frame size

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

1. **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.
2. **End Delimiter (ED) –** We can introduce an ED(pattern) to indicate the end of the frame. Used in **Token Ring**. The problem with this is that ED can occur in the data. This can be solved by:
   1. Character/Byte Stuffing: Used when frames consist of characters. If data contains ED then, a byte is stuffed into data to differentiate it from ED.



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 01110**1**, receiver removes the 0 and reads the data.



2.Error Detection

When data is transmitted from one device to another device, the system does not guarantee whether the data received by the device is identical to the data transmitted by another device. An Error is a situation when the message received at the receiver end is not identical to the message transmitted.

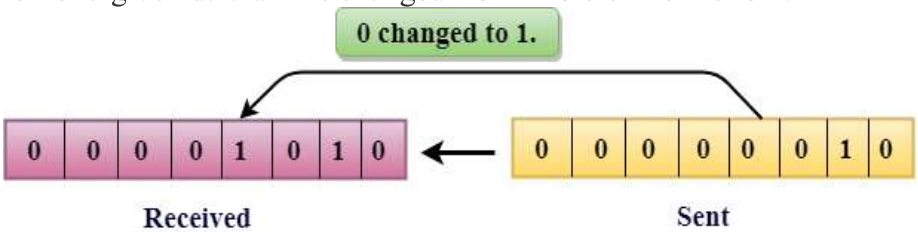
Types Of Errors

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.

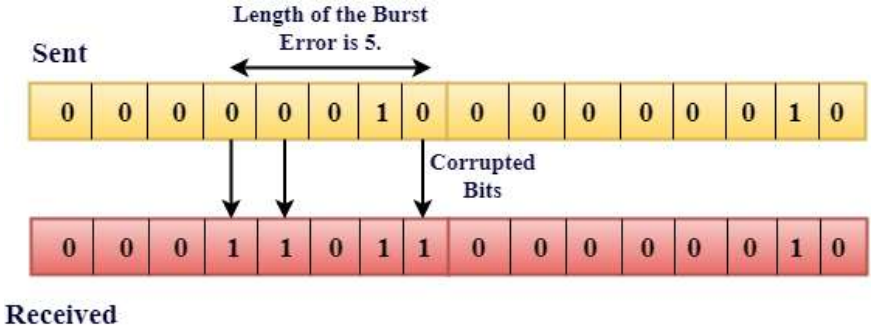


In the above figure, the message which is sent is corrupted as single-bit, i.e., 0 bit is changed to 1.**Single-Bit Error** does not appear more likely in Serial Data Transmission. For example, Sender sends the data at 10 Mbps, this means that the bit lasts only for 1 ?s and for a single-bit error to 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.



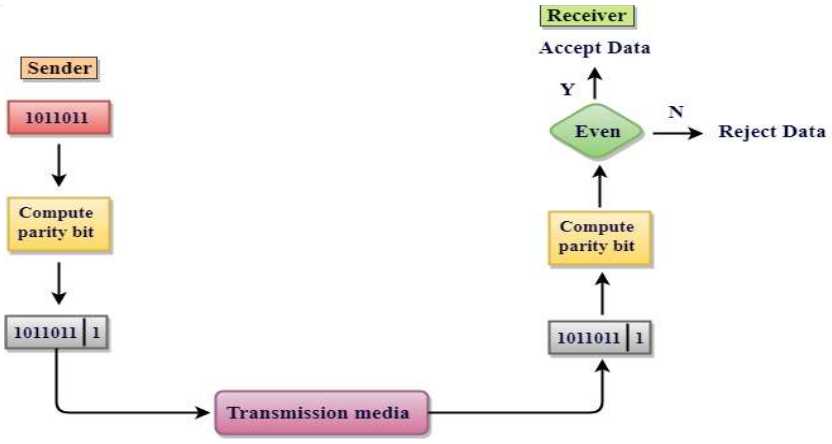
Error Detecting Techniques:

The most popular Error Detecting Techniques are:

1. **Single parity check**
2. **Checksum**
3. **Cyclic redundancy check**

1.Single Parity Check

- Single Parity checking is the simple mechanism and inexpensive to detect the errors.
- In this technique, a redundant bit is also known as a parity bit which is appended at the end of the data unit so that the number of 1s becomes even. Therefore, the total number of transmitted bits would be 9 bits.
- If the number of 1s bits is odd, then parity bit 1 is appended and if the number of 1s bits is even, then parity bit 0 is appended at the end of the data unit.
- At the receiving end, the parity bit is calculated from the received data bits and compared with the received parity bit.
- This technique generates the total number of 1s even, so it is known as even-parity checking.

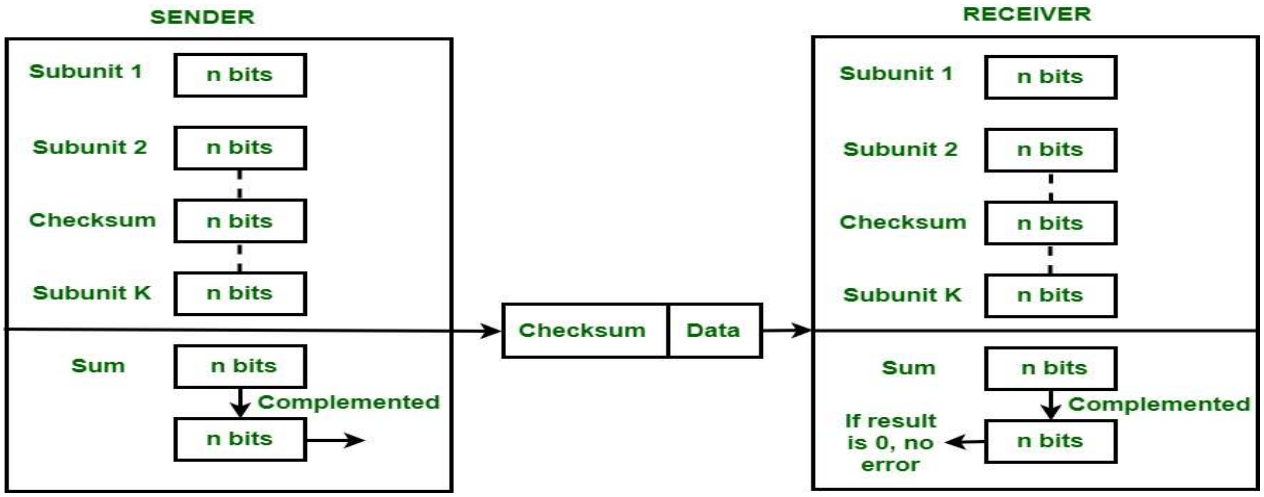


2.Checksum

A checksum is a value that represents the number of bits in a transmission message. IT professionals use it to detect high-level errors within data transmissions.**Checksum** is the error detection method used by upper-layer protocols and is considered to be more reliable than Cyclic Redundancy Check (CRC). This method uses a **Checksum Generator** on the sender side and a **Checksum Checker** on the receiver side.

How Checksum Works?

On the Sender side, the data is divided into equal subunits of n bit length by the checksum generator. This bit is generally of 16-bit length. These subunits are then added together using one's complement method. This sum is of n bits. The resultant bit is then complemented. This complemented sum which is called checksum is appended to the end of the original data unit and is then transmitted to the receiver.



The Receiver after receiving data + checksum passes it to checksum checker. Checksum checker divides this data unit into various subunits of equal length and adds all these subunits. These subunits also contain checksum as one of the **subunits**. The resultant bit is then complemented. If the complemented result is zero, it means the data is error-free. If the result is non-zero it means the data contains an error and Receiver rejects it.

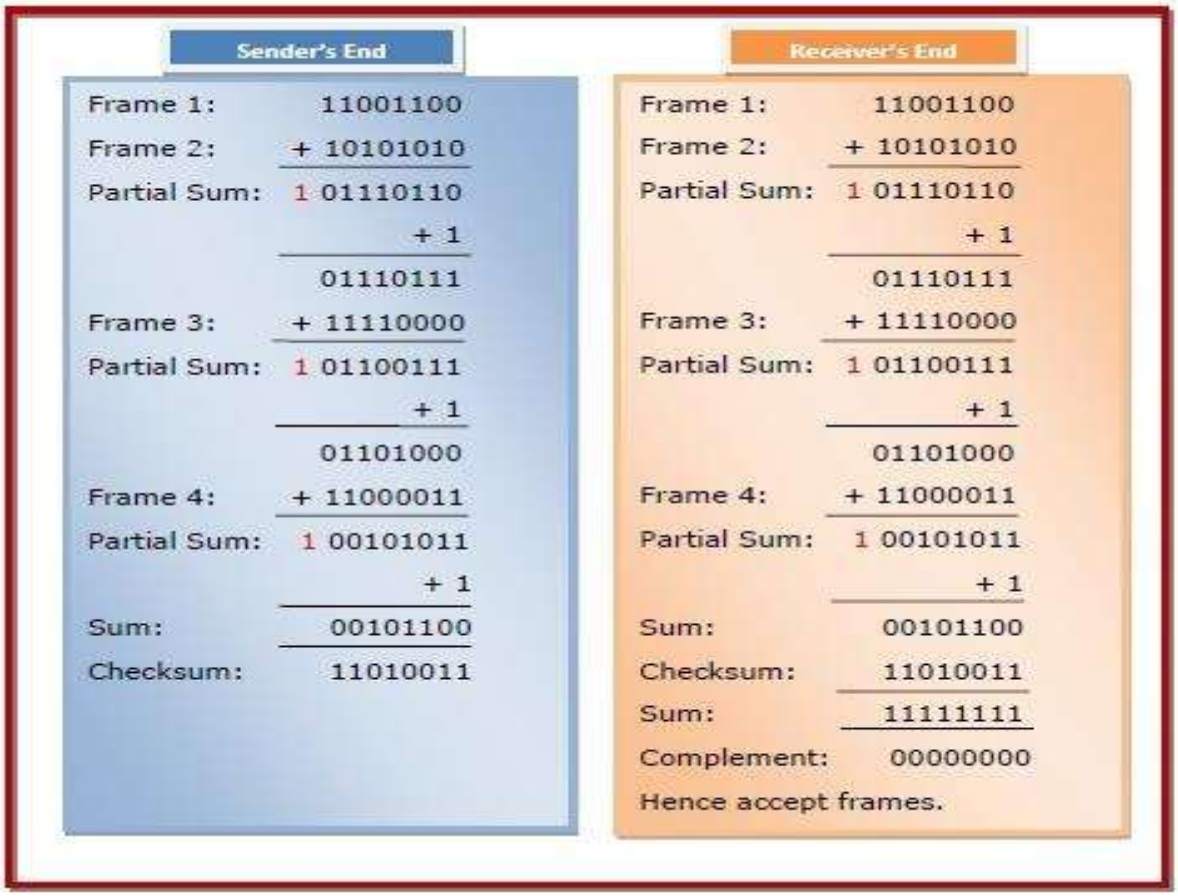
Example

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

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

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

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



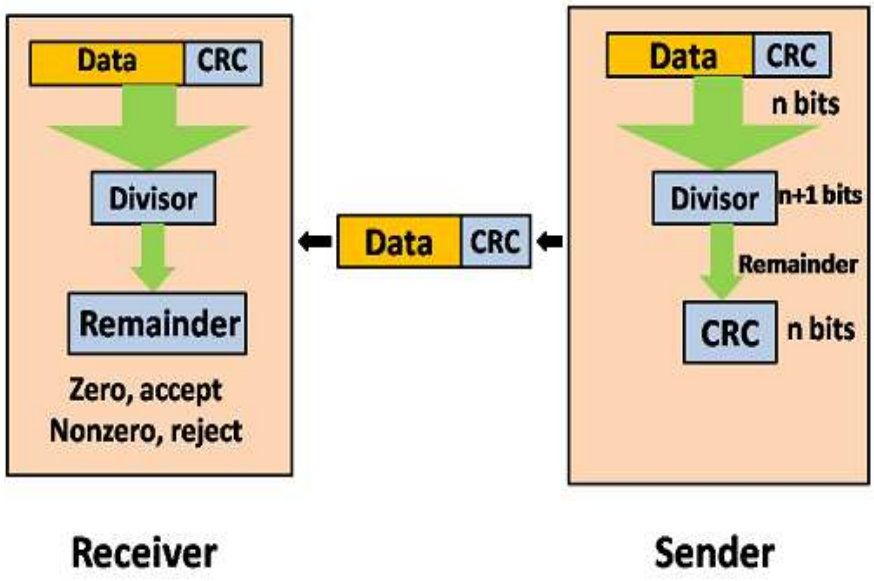
3.Cyclic Redundancy Check (CRC)

CRC is a redundancy error technique used to determine the error.

Following are the steps used in CRC for error detection:

- In CRC technique, a string of n 0s is appended to the data unit, and this n number is less than the number of bits in a predetermined number, known as division which is n+1 bits.
- Secondly, the newly extended data is divided by a divisor using a process is known as binary division. The remainder generated from this division is known as CRC remainder.
- Thirdly, the CRC remainder replaces the appended 0s at the end of the original data. This newly generated unit is sent to the receiver.
- The receiver receives the data followed by the CRC remainder. The receiver will treat this whole unit as a single unit, and it is divided by the same divisor that was used to find the CRC remainder.

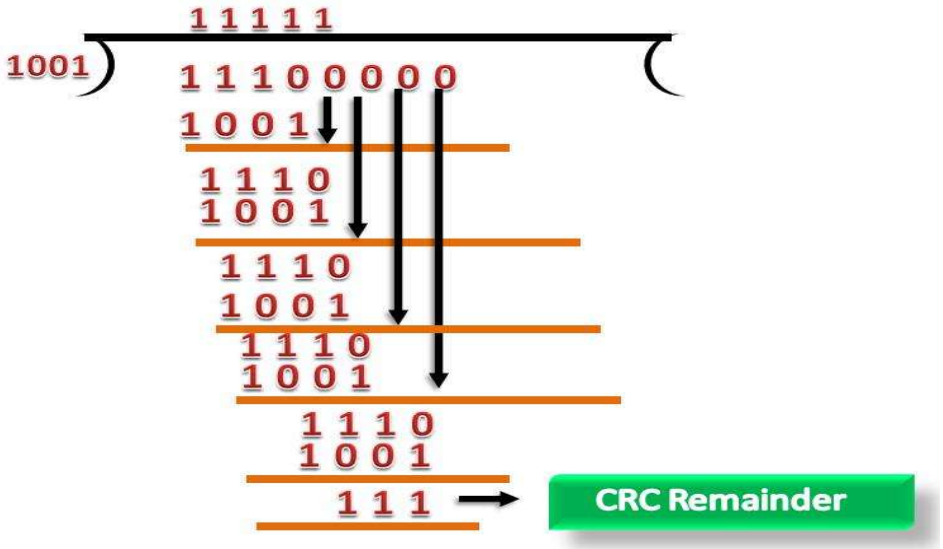
If the resultant of this division is zero which means that it has no error, and the data is accepted.
If the resultant of this division is not zero which means that the data consists of an error. Therefore, the data is discarded.



Suppose the original data is 11100 and divisor is 1001.

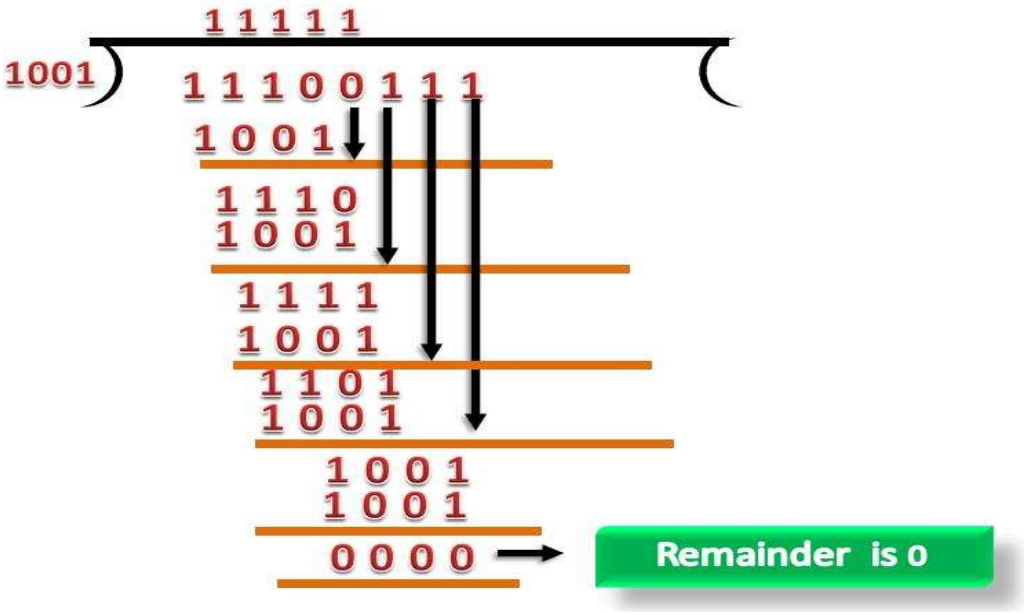
CRC Generator

- A CRC generator uses a modulo-2 division. Firstly, three zeroes are appended at the end of the data as the length of the divisor is 4 and we know that the length of the string 0s to be appended is always one less than the length of the divisor.
- Now, the string becomes 11100000, and the resultant string is divided by the divisor 1001.
- The remainder generated from the binary division is known as CRC remainder. The generated value of the CRC remainder is 111.
- CRC remainder replaces the appended string of 0s at the end of the data unit, and the final string would be 11100111 which is sent across the network.



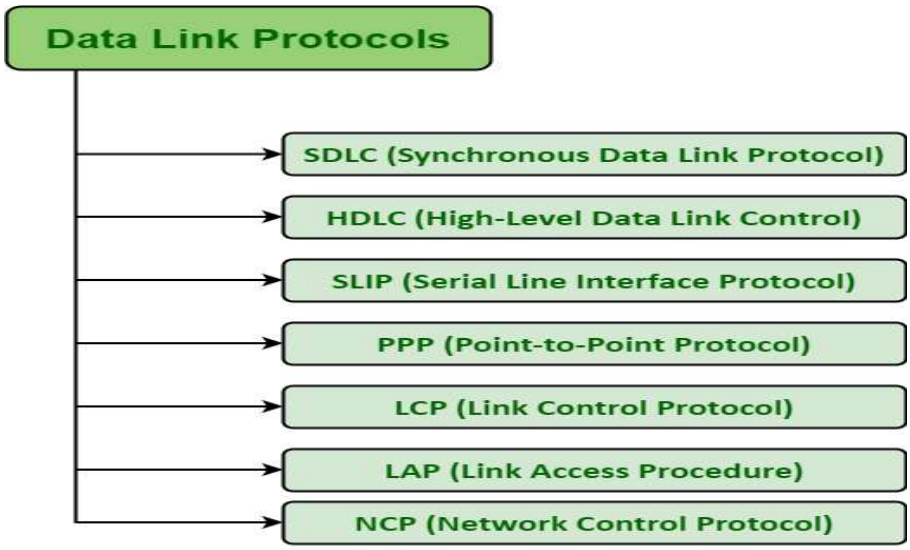
CRC Checker

- The functionality of the CRC checker is similar to the CRC generator.
- When the string 11100111 is received at the receiving end, then CRC checker performs the modulo-2 division.
- A string is divided by the same divisor, i.e., 1001.
- In this case, CRC checker generates the remainder of zero. Therefore, the data is accepted.



Various protocols for data link layer

Data Link Layer protocols are generally responsible for ensuring and confirming that the bits and bytes received are identical to the bits and bytes being transferred. Some of the data link protocols are given below :



1. **Synchronous Data Link Control (SDLC)** – SDLC is basically a communication protocol of computer. It usually supports multipoint links even error recovery or error correction also. It is usually used to carry SNA (Systems Network Architecture) traffic and is present precursor to HDLC. It is also designed and developed by IBM in 1975. It is also used to connect all of the remote devices to mainframe computers at central locations may be in point-to-point (one-to-one) or point-to-multipoint (one-to-many) connections. It is also used to make sure that the data units should arrive correctly and with right flow from one network point to next network point.

2. **High-Level Data Link Protocol (HDLC)** – HDLC is basically a protocol that is now assumed to be an umbrella under which many Wide Area protocols sit. It is also adopted as a part of X.25 network. It was originally created and developed by ISO in 1979. This protocol is generally based on SDLC. It also provides best-effort unreliable service and also reliable service. HDLC is a bit-oriented protocol that is applicable for point-to-point and multipoint communications both.

3. **Serial Line Interface Protocol (SLIP)** – SLIP is generally an older protocol that is just used to add a framing byte at end of IP packet. It is basically a data link control facility that is required for transferring IP packets usually among Internet Service Providers (ISP) and a home user over a dial-up link. It is an encapsulation of the TCP/IP especially designed to work with over serial ports and several router connections simply for communication. It is some limitations like it does not provide mechanisms such as error correction or error detection.

4. **Point to Point Protocol (PPP)** – PPP is a protocol that is basically used to provide same functionality as SLIP. It is most robust protocol that is used to transport other types of packets also along with IP Packets. It can also be required for dial-up and leased router-router lines. It basically provides framing method to describe frames. It is a character-oriented protocol that is also used for error detection. It is also used to provides two protocols i.e. NCP and LCP. LCP is used for bringing lines up, negotiation of options,

bringing them down whereas NCP is used for negotiating network-layer protocols. It is required for same serial interfaces like that of HDLC.

5. **Link Control Protocol (LCP) –** It was originally developed and created by IEEE 802.2. It is also used to provide HDLC style services on LAN (Local Area Network). LCP is basically a PPP protocol that is used for establishing, configuring, testing, maintenance, and ending or terminating links for transmission of data frames.

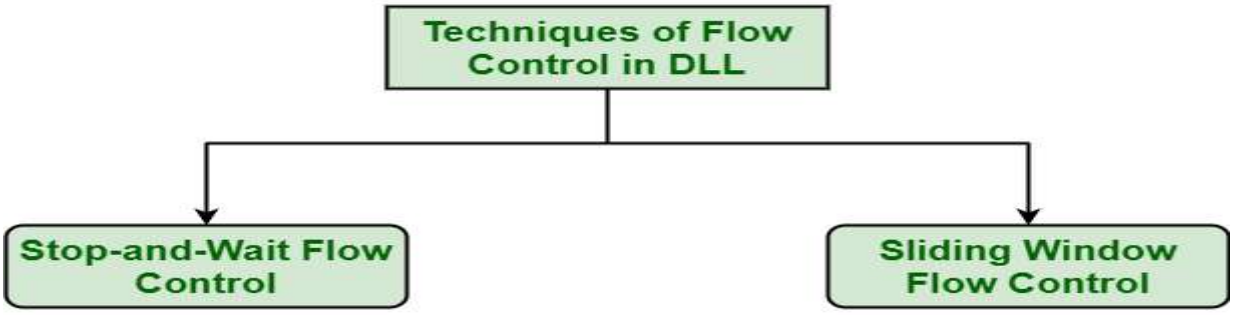
6. **Link Access Procedure (LAP)** – LAP protocols are basically a data link layer protocols that are required for framing and transferring data across point-to-point links. It also includes some reliability service features. There are basically three types of LAP i.e. LAPB (Link Access Procedure Balanced), LAPD (Link Access Procedure D-Channel), and LAPF (Link Access Procedure Frame-Mode Bearer Services). It is actually originated from IBM SDLC, which is being submitted by IBM to the ISP simply for standardization.

7. **Network Control Protocol (NCP) –** NCP was also an older protocol that was implemented by ARPANET. It basically allows users to have access to use computers and some of the devices at remote locations and also to transfer files among two or more computers. It is generally a set of protocols that is forming a part of PPP. NCP is always available for each and every higher-layer protocol that is supported by PPP. NCP was replaced by TCP/IP in the 1980s.

Flow Control in Data Link Layer

Flow control is a technique that allows two stations working at different speeds to communicate with each other. In data link layer, flow control restricts the number of frames the sender can send before it waits for an acknowledgment from the receiver.

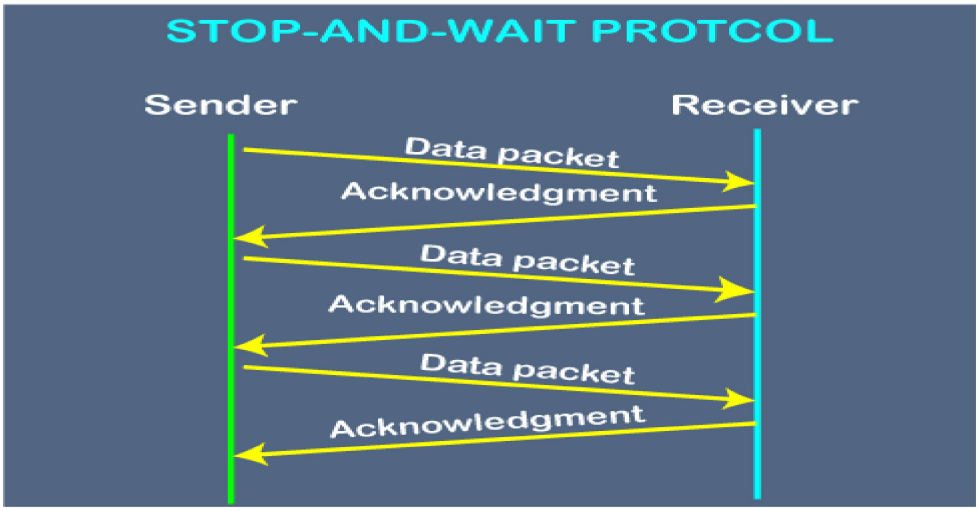
Flow Control Techniques in Data Link Layer



Stop and Wait

This protocol involves the following transitions

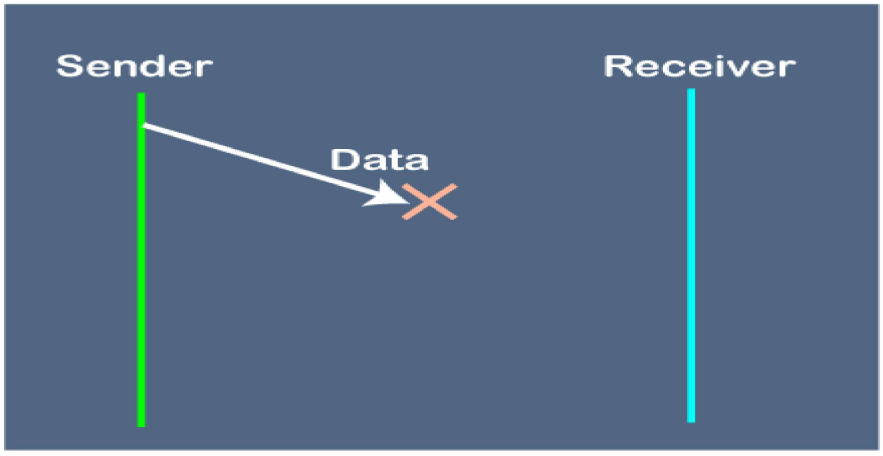
- The sender sends a frame and waits for acknowledgment.
- Once the receiver receives the frame, it sends an acknowledgment frame back to the sender.
- On receiving the acknowledgment frame, the sender understands that the receiver is ready to accept the next frame. So it sender the next frame in queue..



Disadvantages of Stop and Wait protocol

The following are the problems associated with a stop and wait protocol:

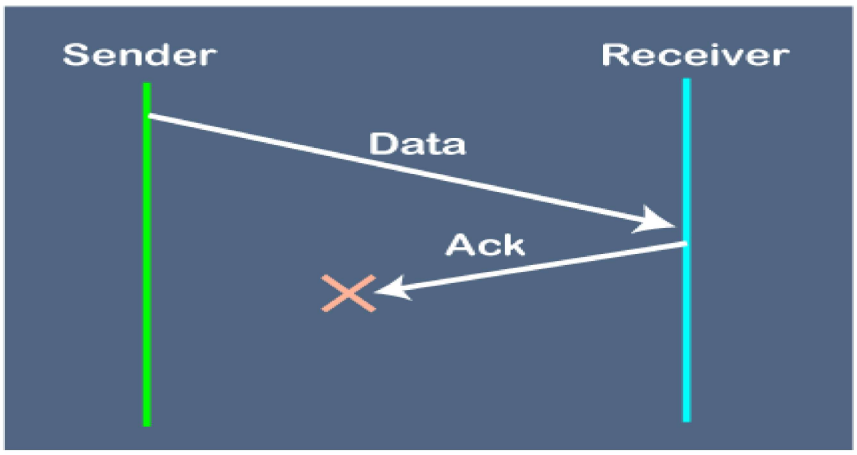
1. Problems occur due to lost data



In this case, two problems occur:

- Sender waits for an infinite amount of time for an acknowledgment.
- Receiver waits for an infinite amount of time for a data.

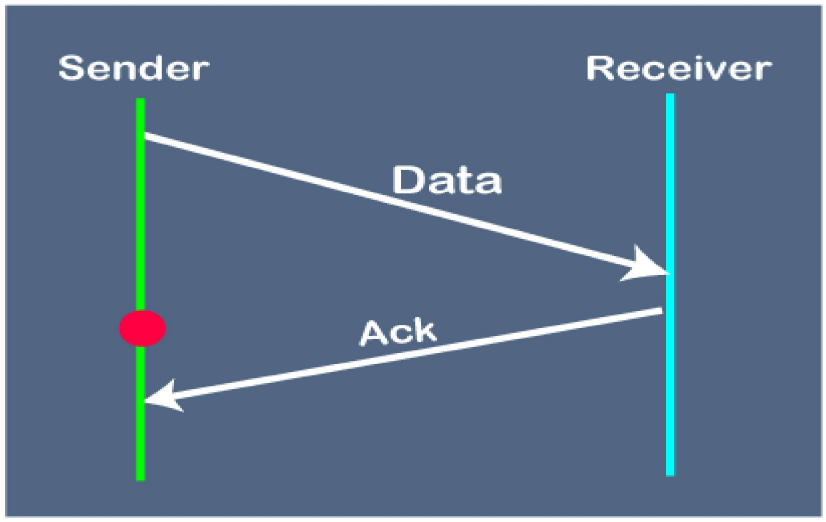
2. Problems occur due to lost acknowledgment



In this case, one problem occurs:

- Sender waits for an infinite amount of time for an acknowledgment.

3. Problem due to the delayed data or acknowledgment



Sliding Window

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

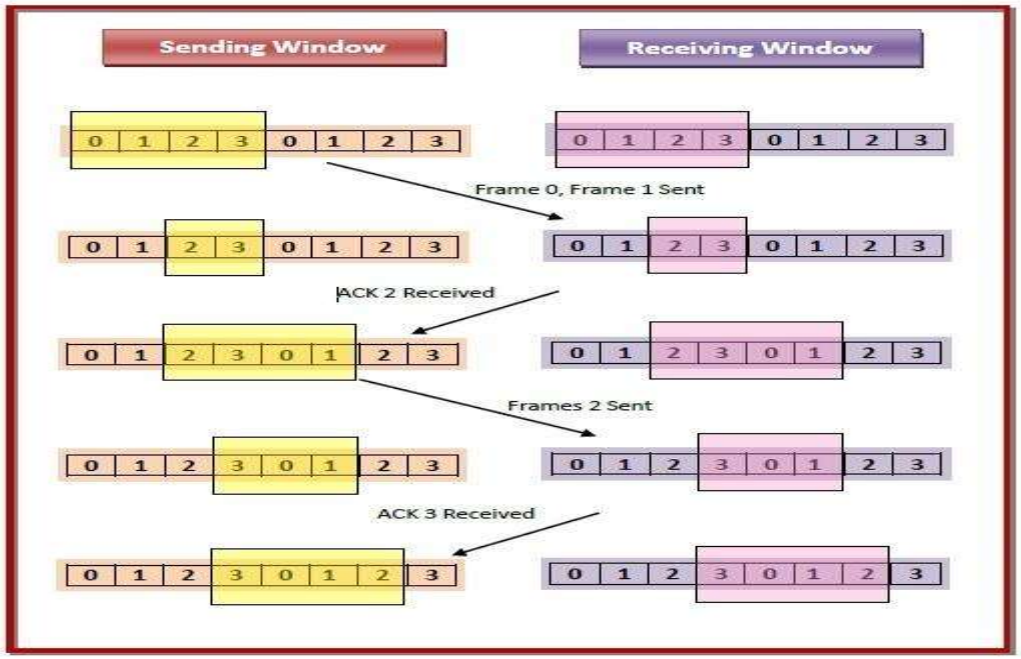
The working principle of this protocol can be described as follows

Both the sender and the receiver has finite sized buffers called windows. The sender and the receiver agrees 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.

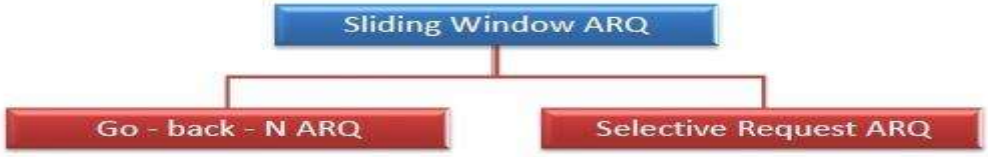
Example

Suppose that we have sender window and receiver window each of size 4. So the sequence numbering of both the windows will be 0,1,2,3,0,1,2 and so on. The following diagram shows the positions of the windows after sending the frames and receiving acknowledgments.



Types of Sliding Window Protocols

The Sliding Window ARQ (Automatic Repeat reQuest) protocols are of two categories –



Go – Back – *N* ARQ

Go – Back – *N* ARQ provides for sending multiple frames before receiving the acknowledgment for the first frame. It uses the concept of sliding window, and so is also called sliding window protocol. The frames are sequentially numbered and a finite number of frames are sent. If the acknowledgment of a frame is not received within the time period, all frames starting from that frame are retransmitted.

Selective Repeat ARQ

This protocol also provides for sending multiple frames before receiving the acknowledgment for the first frame. However, here only the erroneous or lost frames are retransmitted, while the good frames are received and buffered.

MAC Layer

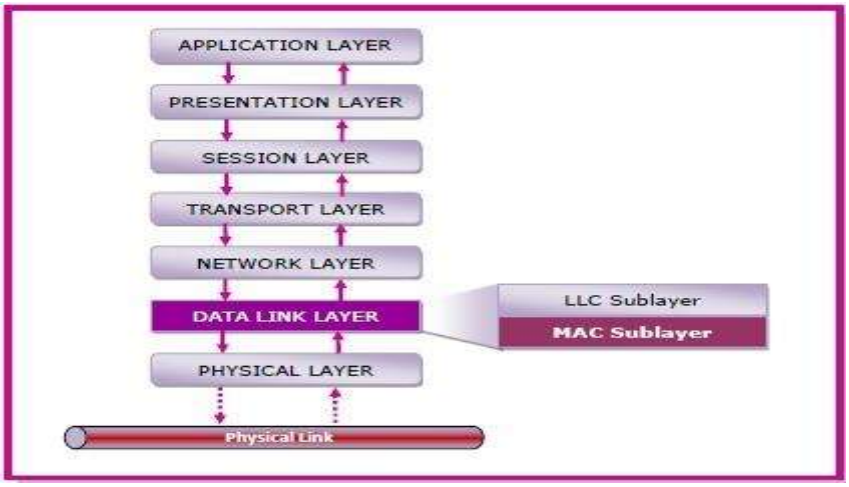
The **medium access control** (MAC) is a sublayer of the **data link layer** of the **open system interconnections (OSI) reference model** for data transmission. It is responsible for flow control and multiplexing for transmission medium. It controls the transmission of data packets via remotely shared channels. It sends data over the network interface card.

MAC Layer in the OSI Model

The Open System Interconnections (OSI) model is a layered networking framework that conceptualizes how communications should be done between heterogeneous systems. The data link layer is the second lowest layer. It is divided into two sublayers −

- **The logical link control (LLC) sublayer**
- **The medium access control (MAC) sublayer**

The following diagram depicts the position of the MAC layer –



Functions of MAC Layer

- It provides an abstraction of the physical layer to the LLC and upper layers of the OSI network.
- It is responsible for encapsulating frames so that they are suitable for transmission via the physical medium.
- It resolves the addressing of source station as well as the destination station, or groups of destination stations.
- It performs multiple access resolutions when more than one data frame is to be transmitted. It determines the channel access methods for transmission.
- It also performs collision resolution and initiating retransmission in case of collisions.
- It generates the frame check sequences and thus contributes to protection against transmission errors.

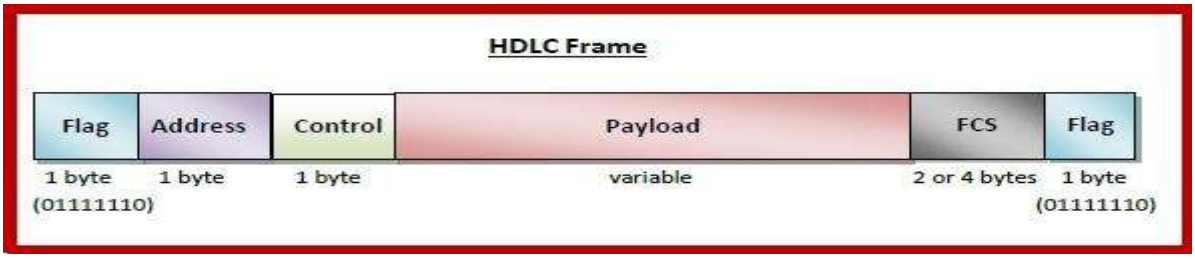
High-Level Data Link Control (HDLC)

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.

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 −

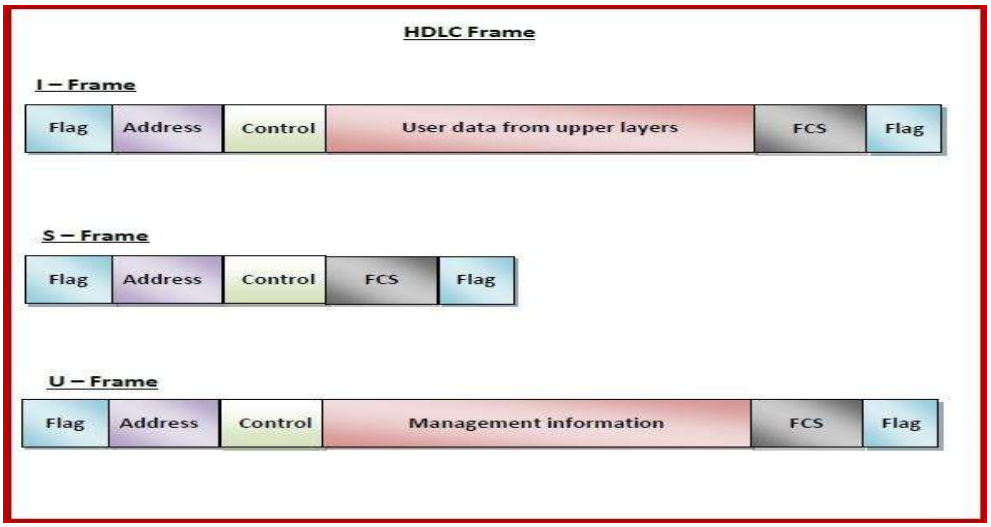
- **Flag** − It is an 8-bit sequence that marks the beginning and the end of the frame. The bit pattern of the flag is 01111110.
- **Address** − It contains the address of the receiver. If the frame is sent by the primary station, it contains the address(es) of the secondary station(s). If it is sent 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 −

1. **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.
2. **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.
3. **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.



Carrier Sense Multiple Access (CSMA)

Carrier Sense Multiple Access (CSMA) is a method used in computer networks to manage how devices share a communication channel to transfer the data between two devices. In this protocol, each device first sense the channel before sending the data. If the channel is busy, the device waits until it is free. This helps reduce collisions. CSMA is commonly used in technologies like Ethernet and Wi-Fi.

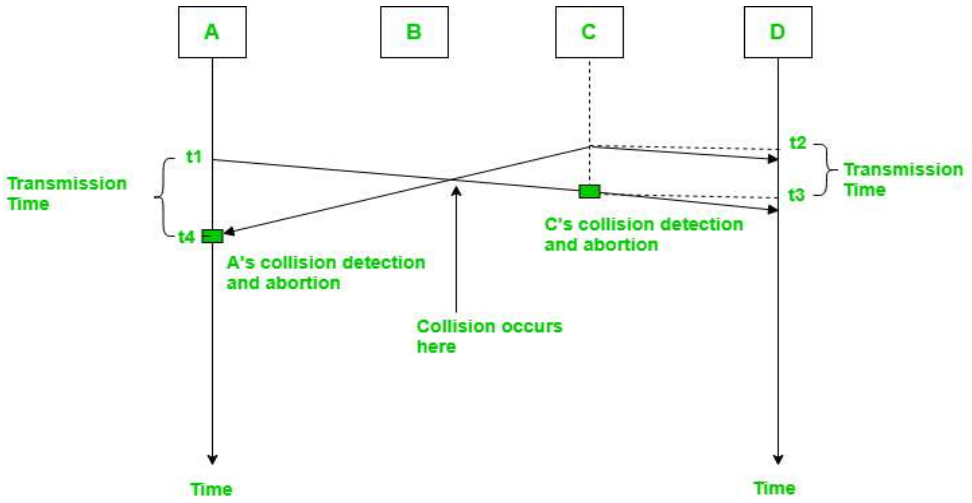
Types of CSMA Protocol

There are two main types of **Carrier Sense Multiple Access (CSMA)** protocols, each designed to handle how devices manage potential data collisions on a shared communication channel. These types differ based on how they respond to the detection of a busy network:

1. CSMA/CD
2. CSMA/CA

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

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



In the diagram, *starts* sending 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

Types of CSMA Access Modes

There are 4 types of access modes available in CSMA. It is also referred as 4 different types of CSMA protocols which decide the time to start sending data across shared media.

1. **1-Persistent:** It senses the shared channel first and delivers the data right away if the channel is idle. If not, it must wait and ***continuously*** track for the channel to become idle and then broadcast the frame without condition as soon as it does. It is an aggressive transmission algorithm.

2. **Non-Persistent:** It first assesses the channel before transmitting data; if the channel is idle, the node transmits data right away. If not, the station must wait for an arbitrary amount of time (***not continuously***), and when it discovers the channel is empty, it sends the frames.
3. **P-Persistent:** It consists of the 1-Persistent and Non-Persistent modes combined. Each node observes the channel in the 1Persistent mode, and if the channel is idle, it sends a frame with a P probability. If the data is not transferred, the frame restarts with the following time slot after waiting for a (q = 1-p probability) random period.