# MODULE 3: DATA LINK LAYER

*(NOTE:YEH MODULE SE SUMS BHI AYENGE)*

## Q.1) **FRAMING METHODS IN DATA LINK LAYER**

**Framing in the Data Link Layer**

The bits to be transmitted at first broken into discrete frame at the data link Layer.

In order to guarantee that the bit stream is error free, the checksum of each frame is computed.

When a frame is received, the data link there computes the checksum.

If it is different from the checksum present in the frame, then the data link layer knows that an error has occurred.

It then discard the bad frame and sends back a request for re-transmission.

Breaking the bit stream into frames is called as framing.

### Framing Methods

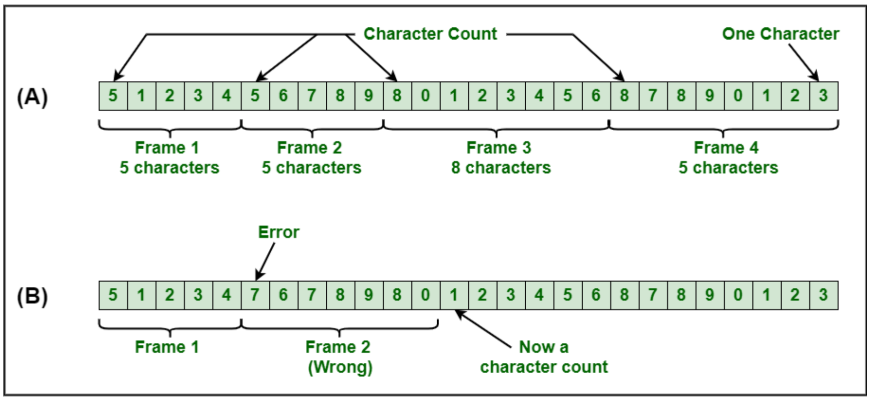
1. **Character Count**
2. **Character Stuffing (Byte Stuffing)**
3. **Bit Stuffing**
4. **Physical Layer Coding Violations**

Each method helps maintain data integrity and synchronization, enabling error-free and structured communication across network points.

1. **Character Count :**This method is rarely used and is generally required to count total number of characters that are present in frame. This is be done by using field in header. Character count method ensures data link layer at

the receiver or destination about total number of characters that follow, and about where the frame ends.

There is disadvantage also of using this method i.e., if anyhow character count is disturbed or distorted by an error occurring during transmission, then destination or receiver might lose synchronization. The destination or receiver might also be not able to locate or identify beginning of next frame.

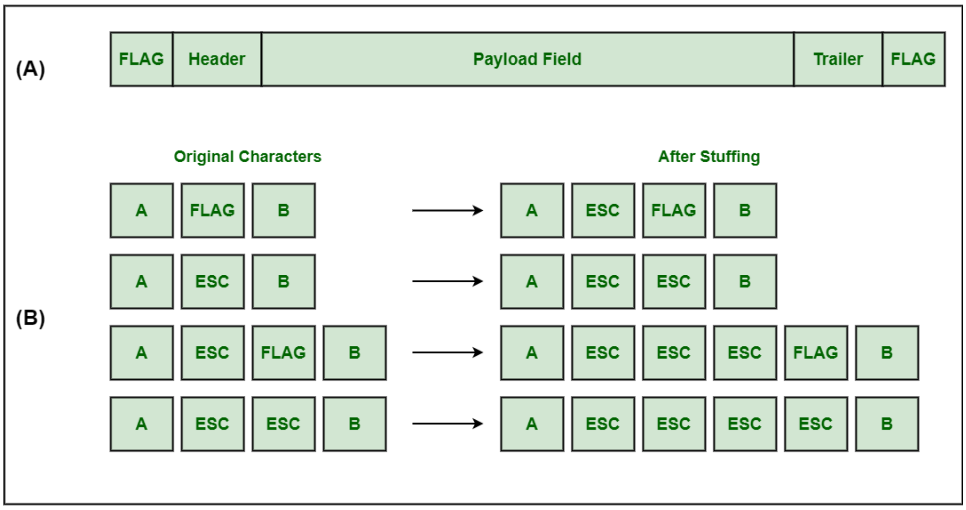



**A Character Stream**

(A) Without Errors
(B) With one Error

2. **Character Stuffing:**Character stuffing is also known as byte stuffing or character-oriented framing and is same as that of bit stuffing but byte stuffing actually operates on bytes whereas bit stuffing operates on bits. In byte stuffing, special byte that is basically known as ESC (Escape Character) that has predefined pattern is generally added to data section of the data stream or frame when there is message or character that has same pattern as that of flag byte.

   But receiver removes this ESC and keeps data part that causes some problems or issues. In simple words, we can say that character stuffing is addition of 1 additional byte if there is presence of ESC or flag in text.




**A Character Stuffing**
(A) A frame delimited by flag bytes
(B) Four examples of byte sequences before and after byte stuffing

3. **Bit Stuffing :**Bit stuffing is also known as bit-oriented framing or bit-oriented approach. In bit stuffing, extra bits are being added by network protocol designers to data streams. It is generally insertion or addition of extra bits into transmission unit or message to be transmitted as simple way to provide and give signaling information and data to receiver and to avoid or ignore appearance of unintended or unnecessary control sequences.

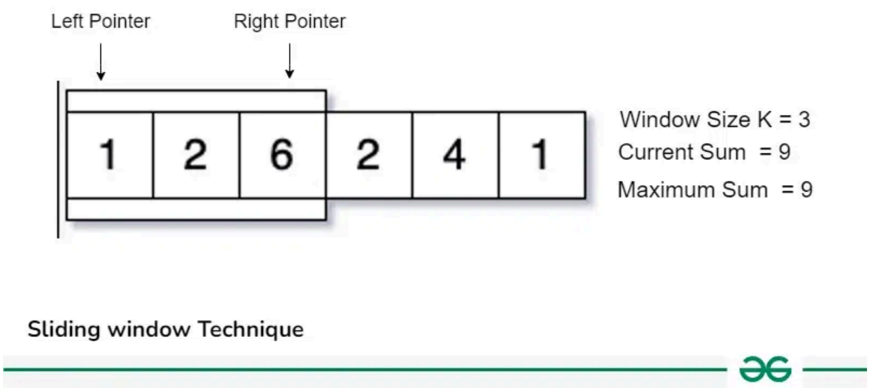
   It is type of protocol management simply performed to break up bit pattern that results in transmission to go out of synchronization. Bit stuffing is very essential part of transmission process in network and communication protocol. It is also required in USB.

4. **Physical Layer Coding Violations :**Encoding violation is method that is used only for network in which encoding on physical medium includes some sort of redundancy i.e., use of more than one graphical or visual structure to simply encode or represent one variable of data.

# Q.2) WRITE A NOTE ON SLIDING WINDOW

**Sliding Window Protocol**

The Sliding Window protocol is a data link layer technique used for efficient and reliable data transmission, especially in networks with varying speeds and potential delays, like the internet. It manages the flow of data between a sender and a receiver, ensuring that packets are transmitted without overwhelming the receiver and that data integrity is maintained.




## Key Components

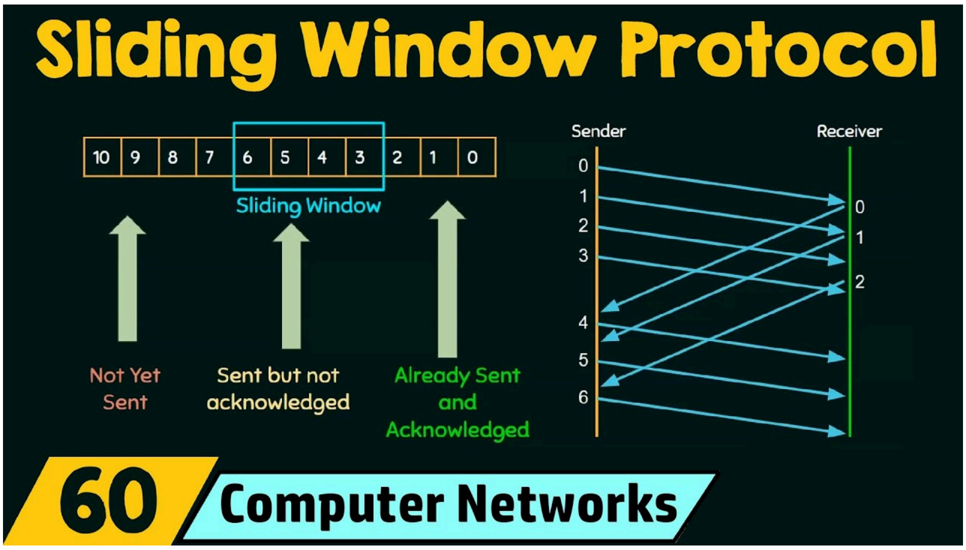
1. **Window Size**:
   - A set number of frames the sender can transmit without waiting for an acknowledgment.
   - Both sender and receiver maintain a "window" that slides as frames are sent and acknowledged, hence the name.
2. **Acknowledgments**:
   - The receiver sends back acknowledgments to the sender for each received frame. If the sender receives an acknowledgment, the window slides forward, allowing the sender to transmit more frames.
3. **Error Control**:
   - If a frame is lost or damaged, the receiver can request retransmission, ensuring data is accurately received in sequence.

## Types of Sliding Window Protocols

1. **Stop-and-Wait**:
   - The simplest form, where the sender sends one frame and waits for an acknowledgment before sending the next. This limits transmission speed and is suitable for low-delay networks.
2. **Go-Back-N**:
   - The sender can send multiple frames up to the window size. If an error is detected, the sender goes back and retransmits from the erroneous

frame onward. This is faster than Stop-and-Wait but may involve redundant retransmissions.

3. **Selective Repeat**:
    - The sender only retransmits the specific frames that failed, rather than the entire sequence from the error onward. This approach maximizes efficiency and minimizes retransmissions.




## Advantages

- **Efficient Use of Bandwidth**: Multiple frames can be sent before requiring acknowledgment, maximizing throughput.
- **Error Recovery**: Ensures data integrity by retransmitting lost or damaged frames.
- **Flow Control**: Prevents data overload at the receiver by adjusting the transmission rate.

## Applications

Sliding Window protocols are widely used in TCP/IP networks, data streaming, file transfers, and other applications where reliable, orderly data transmission is essential. It's particularly effective in high-speed and high-latency networks, where maintaining efficient data flow and minimizing retransmissions are crucial for performance.

*(NOTE: EK EXAMPLE SUM KARKE EXPLAIN KR SKTE in sliding window: go-back-n)*

# Q.3) EXPLAIN CSMA PROTOCOLS. EXPLAIN HOW COLLISION ARE HANDLED IN CSMA/CD.

⇒

The CSMA protocol operates on the principle of carrier sensing.

In this protocol, a station listens to see the presence of transmission (carrier) on the cable and decides to act accordingly.

**Non-Persistent CSMA:**

In this scheme, if a station wants to transmit a frame and it finds that the channel is busy, then it will wait for fixed interval of time.

After this time, it again checks the status of the channel, and if the channel is free, it will transmit.

**1-Persistent CSMA:**

In this scheme, the station which wants to transmit continuously monitors the channel until it is idol and then transmit it immediately.

Disadvantage of the strategy is that if two patients are waiting, then they will transfer simultaneously and collagen will take place. This then required re-transmission.

**P-Persistent CSMA:**

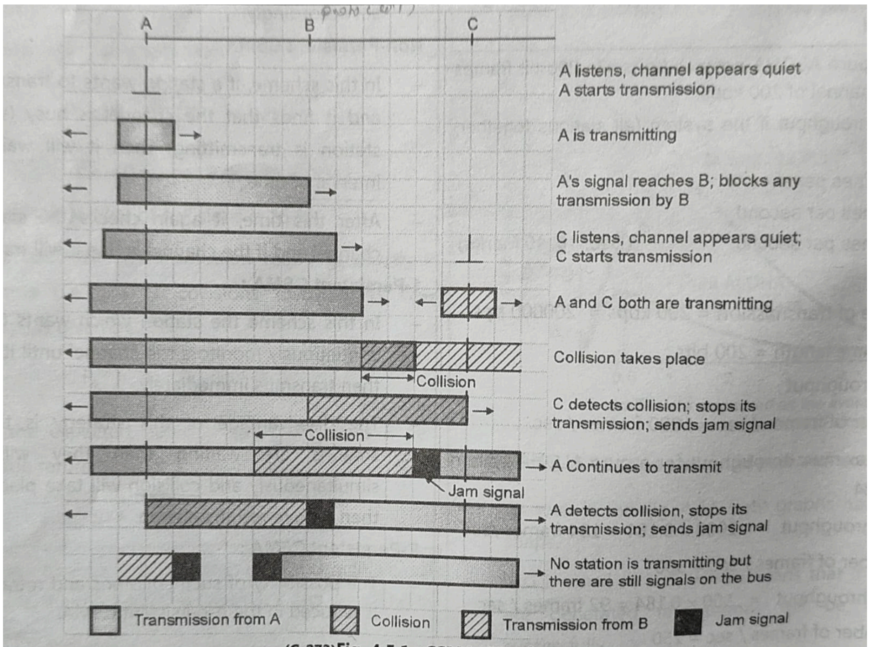
The possibility of such collision and transmission is reduced in the P-Persistent CSMA.

In this scheme, all the waiting stations are not allowed to transmit simultaneously as soon as the channel becomes idol.

A station is assumed to be transmitting with a probability “p”.

For example, if p=1/6, and if 6 stations are waiting, then on an average, only one station will transfer and other will wait.

## CARRIER SENSE MULTIPLE ACCESS/COLLISON DETECTION (CSMA/CD):



(G-273)Fig. 4.5.1 : CSMA/CD scheme

The CSMA/CD specification have been standardised by IEEE 802.3 standardisation.

It is a widely used MAC protocol.

Media access control:

-The problem in CSA explain earlier is that a transmitting station contains to transmit its frame, even though a collision occurs

-The channel time is unnecessarily wasted due to this. In CSMA/CD, if a station receives other transmission when it is transmitting, then a collision can be detected as soon as it occurs and the transmission time can be saved.

-As soon as a collision is detected, the transmitting station releases a jam signal.

-Station are not supposed to transport immediately after the collision has occurred.

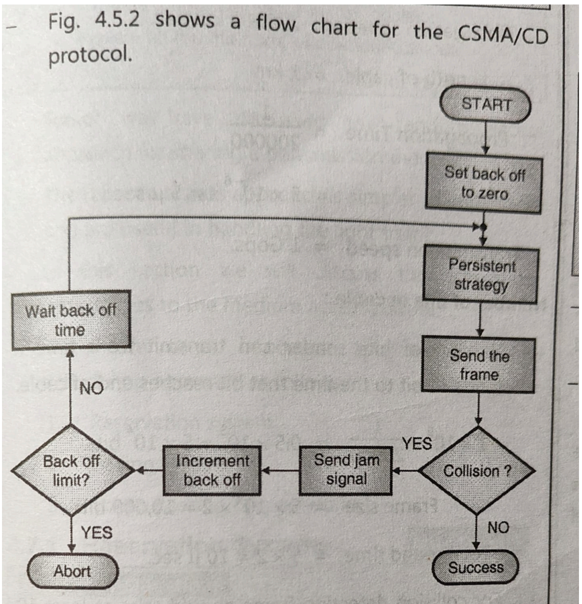
-Otherwise, there is a possibility that the same friends would collide again.

-After some "back off" delay time the stations will retry the transmission

-If again the collision takes place then the back off time is increased progressively.

-A careful design can achieve efficiencies of more than 90% using CSMA/CD.

**CSMA/CD PROCEDURE:**



(G-276)Fig. 4.5.2 : CSMA/CD procedure

- The station that has a ready frame at the back off parameter to 0.
- Then it senses the line using one of the persistent strategies.
- It then send the frame if there is no collision for a period corresponding to one complete frame, then the transmission is successful.
- Otherwise, the station sends the jam signal to inform the other station about the collision.
- The station then increments the back off time and wait for a random back off time and send the frame again.
- If the back of has reached its limit, then the station about the transmission.

# Q.4) CHANNEL APPLICATION PROBLEM.

## Channel Application Problem

**Scenario**: Consider a shared communication channel in a Local Area Network (LAN) where multiple devices (computers) are trying to send data simultaneously. We will analyze a problem involving data transmission using the CSMA (Carrier Sense Multiple Access) protocol, focusing on collision management.

## Problem Statement

Suppose we have a network with four devices (A, B, C, and D) that need to send packets of data to a central server. The devices utilize the CSMA protocol for channel access.

- Each device has a packet of size 1,000 bytes to send.
- The channel bandwidth is 1 Mbps (1,000,000 bits per second).
- The round-trip time for an acknowledgment (ACK) is 40 ms.
- Devices will wait for a random backoff time after a collision.

## Questions to Address

1. **What is the time taken for each device to successfully transmit its packet, considering potential collisions?**
2. **How does the backoff mechanism affect overall transmission time?**
3. **What would happen if all devices attempted to send data simultaneously without the CSMA protocol?**

## Solution Steps

1. **Calculate Transmission Time Without Collisions**:
    - The time taken to transmit 1,000 bytes (8,000 bits) is calculated as:
    $$
    \text{Transmission Time} = \frac{\text{Packet Size (bits)}}{\text{Channel Bandwidth (bps)}} = \frac{8000 \text{ bits}}{1,000,000 \text{ bps}} = 0.008 \text{ seconds} = 8 \text{ ms}
    $$
    - Each device would take approximately 8 ms to transmit its packet if there are no collisions.
2. **Collision Scenarios**:

- Assume that all four devices attempt to send their packets simultaneously.
- A collision occurs when two or more devices transmit at the same time.

3. **Backoff Mechanism**:
    - After a collision, devices will wait for a random backoff time before attempting to retransmit.
    - If we assume a random backoff range of 0 to 4 time slots (where each slot is 20 ms), the backoff time could be between 0 ms and 80 ms.
4. **Successive Retransmissions**:
    - If collisions continue to occur, the average time taken for a successful transmission increases.
    - For example, if Device A waits for 20 ms, Device B waits for 40 ms, and Devices C and D wait longer, Device A might transmit successfully first.
5. **Total Transmission Time Calculation**:
    - For each successful transmission after backoff, you would add the transmission time (8 ms) and the random backoff time.
    - The overall time taken would be the cumulative time for all successful transmissions plus the time spent in backoff for each device.

## Example Calculation:

- Assume the following successful transmission after backoff:
    - Device A: Backoff of 20 ms, Transmission = 8 ms, Total = 28 ms
    - Device B: Backoff of 40 ms, Transmission = 8 ms, Total = 48 ms
    - Device C: Backoff of 60 ms, Transmission = 8 ms, Total = 68 ms
    - Device D: Backoff of 80 ms, Transmission = 8 ms, Total = 88 ms

## Conclusion

- **Effectiveness of CSMA**: The CSMA protocol helps manage access to the shared channel, minimizing collisions and optimizing transmission times despite the inherent delays caused by backoff.
- **Collisions Impact**: Without CSMA, if all devices transmitted simultaneously, there would be continual collisions, resulting in significant delays and

inefficient use of the channel.

This scenario illustrates the importance of using protocols like CSMA to manage access in shared communication channels effectively, ensuring orderly and efficient data transmission.