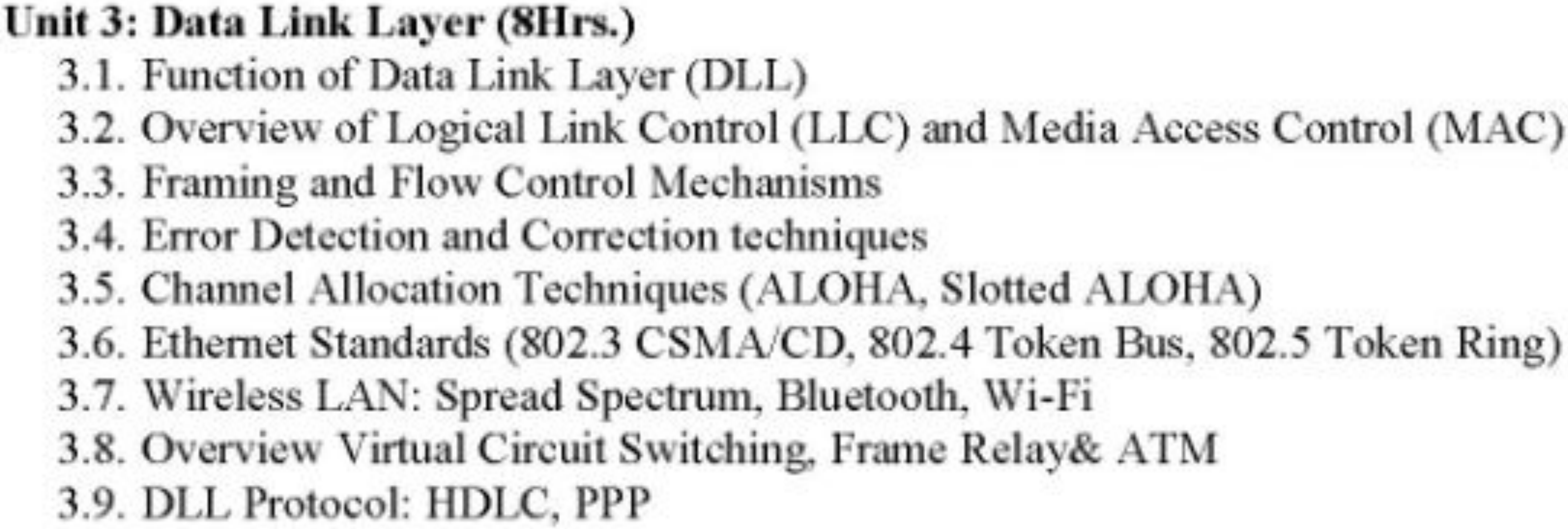
## CHAPTER 3

# DATA LINK LAYER



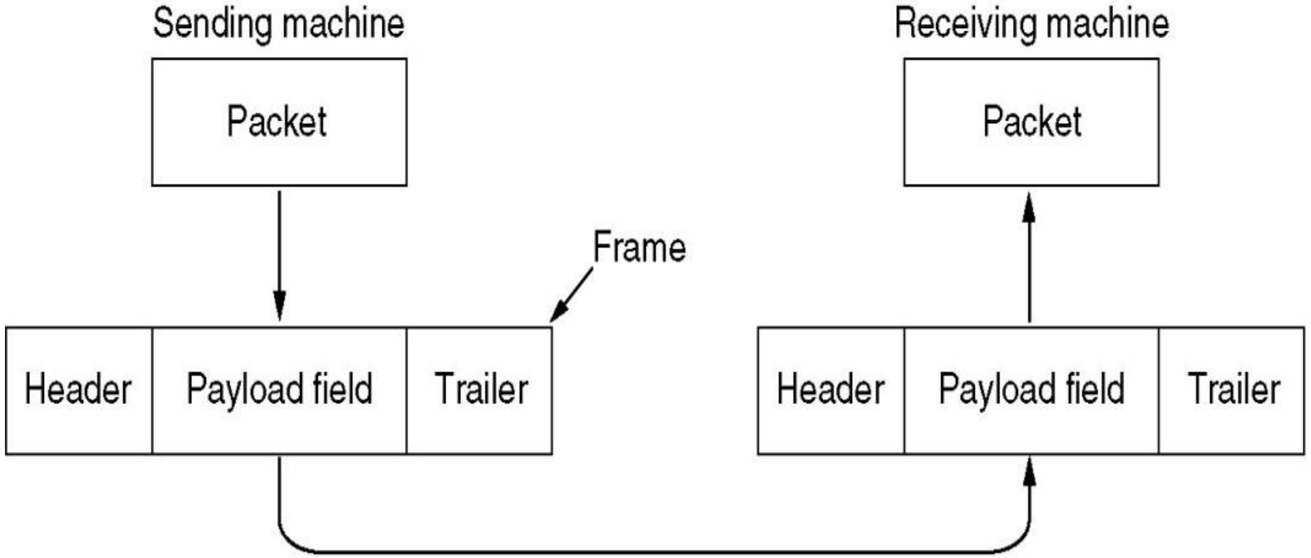
**Unit 3: Data Link Layer (8Hrs.)**

3.1. Function of Data Link Layer (DLL)
3.2. Overview of Logical Link Control (LLC) and Media Access Control (MAC)
3.3. Framing and Flow Control Mechanisms
3.4. Error Detection and Correction techniques
3.5. Channel Allocation Techniques (ALOHA, Slotted ALOHA)
3.6. Ethernet Standards (802.3 CSMA/CD, 802.4 Token Bus, 802.5 Token Ring)
3.7. Wireless LAN: Spread Spectrum, Bluetooth, Wi-Fi
3.8. Overview Virtual Circuit Switching, Frame Relay& ATM
3.9. DLL Protocol: HDLC, PPP

# Topics to Cover

# DATA LINK LAYER

- The **data link layer** is the protocol **layer** in a program that handles the moving of **data** into and out of a physical **link** in a network.
- The **data link layer** is **Layer** 2 in the Open Systems Interconnection (OSI) architecture model for a set of telecommunication protocols.
- most reliable node to node delivery of data.
- forms frames from the packets that are received from network layer and gives it to physical layer.
- second layer of OSI Layered Model.
- responsible for converting data stream to signals bit by bit
- At the receiving end, Data link layer picks up data from hardware which are in the form of electrical signals
- assembles them in a recognizable frame format, and hands over to upper layer.


Sending machine
Packet
Frame
Header
Payload field
Trailer
Receiving machine
Packet
Header
Payload field
Trailer

# 3.1 FUNCTION OF DATA LINK LAYER

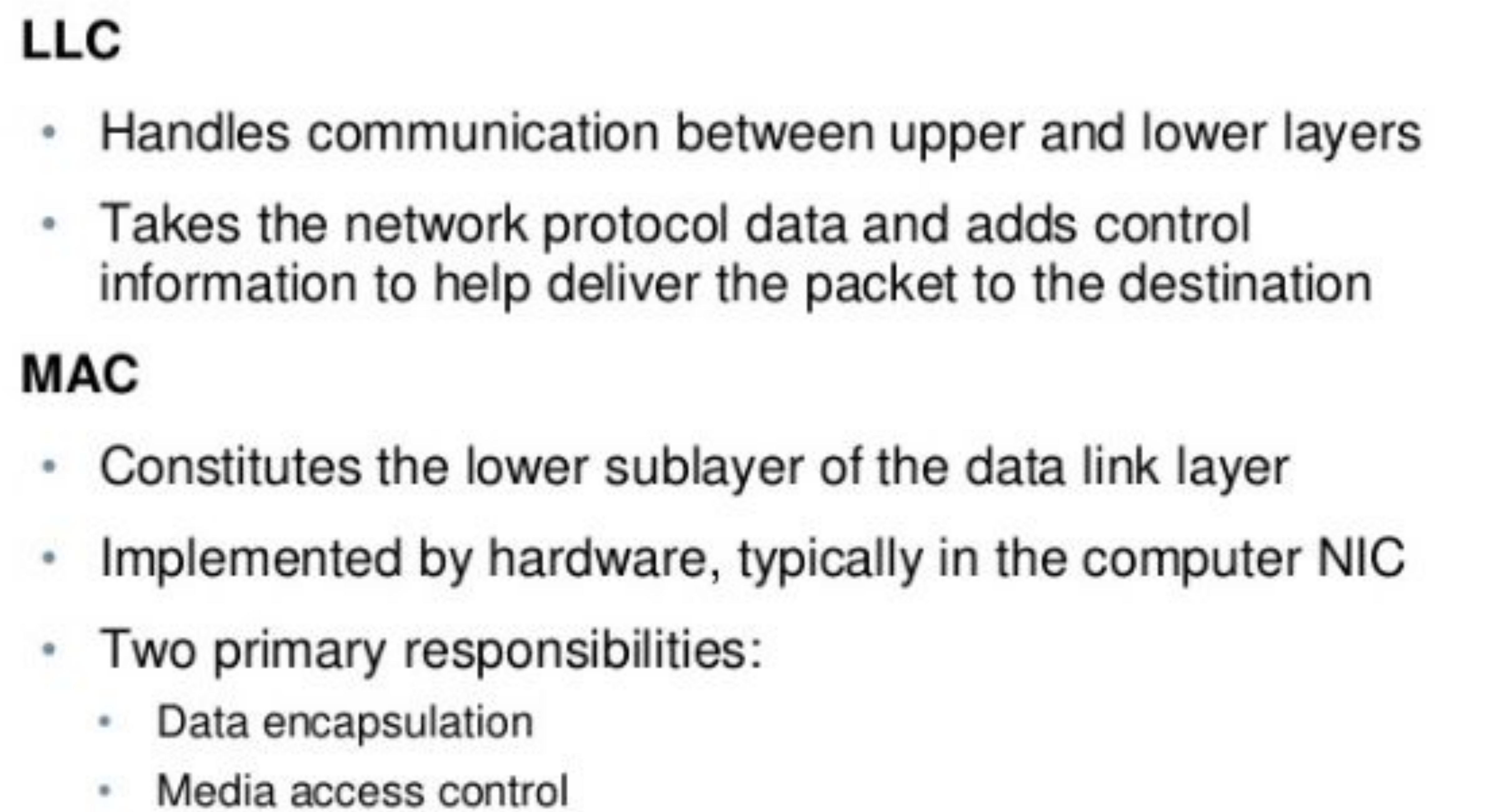
- **Framing**: The **data link layer** receives the stream of bits from the network layer divides into manageable data units called frames.
- **Physical addressing**: If frames are to be distributed to different stations on the network. To define the physical address of the sender (source address) and/or receiver (destination address) of the frame, the DLL adds a header to the frame.

- **Flow control**: If the rate at which the data are consumed by the receiver is less than the rate produced by the sender, the data link layer deals with a flow control mechanism to prevent overrun the receiver.
- **Error control**: The data link layer also deals with damaged or lost frames. By adding mechanisms to detect and retransmit lost frames increases reliability.
- A trailer added to the end of the frame to achieve error control.
- **Access control**: When more than two or two devices are connected to the common link, data link layer protocols are necessary to determine which device has control over the link at any point of time.

# 3.2 OVERVIEW OF LLC AND MAC

- Data link layer has two sub-layers:
- **Logical Link Control (LLC)**: This is the uppermost sub-layer, LLC consists of protocols running at the top of the **data link layer**, and also provides flow control, acknowledgment, and error notification.
- The LLC provides addressing and data link control.
- It specifies which methods are to be used for addressing channels over the transmission medium and for controlling the data exchanged between the generator of packet and recipient of the message.

- **Media Access Control (MAC)**: Who can access the media at any one time, determines by the MAC sublayer(e.g. CSMA/CD).
- The packet obtains from the Network layer is further divided into frames depending on the frame size of NIC(Network Interface Card).
- DLL also encases Sender's and Receiver's MAC address in the header.



## LLC

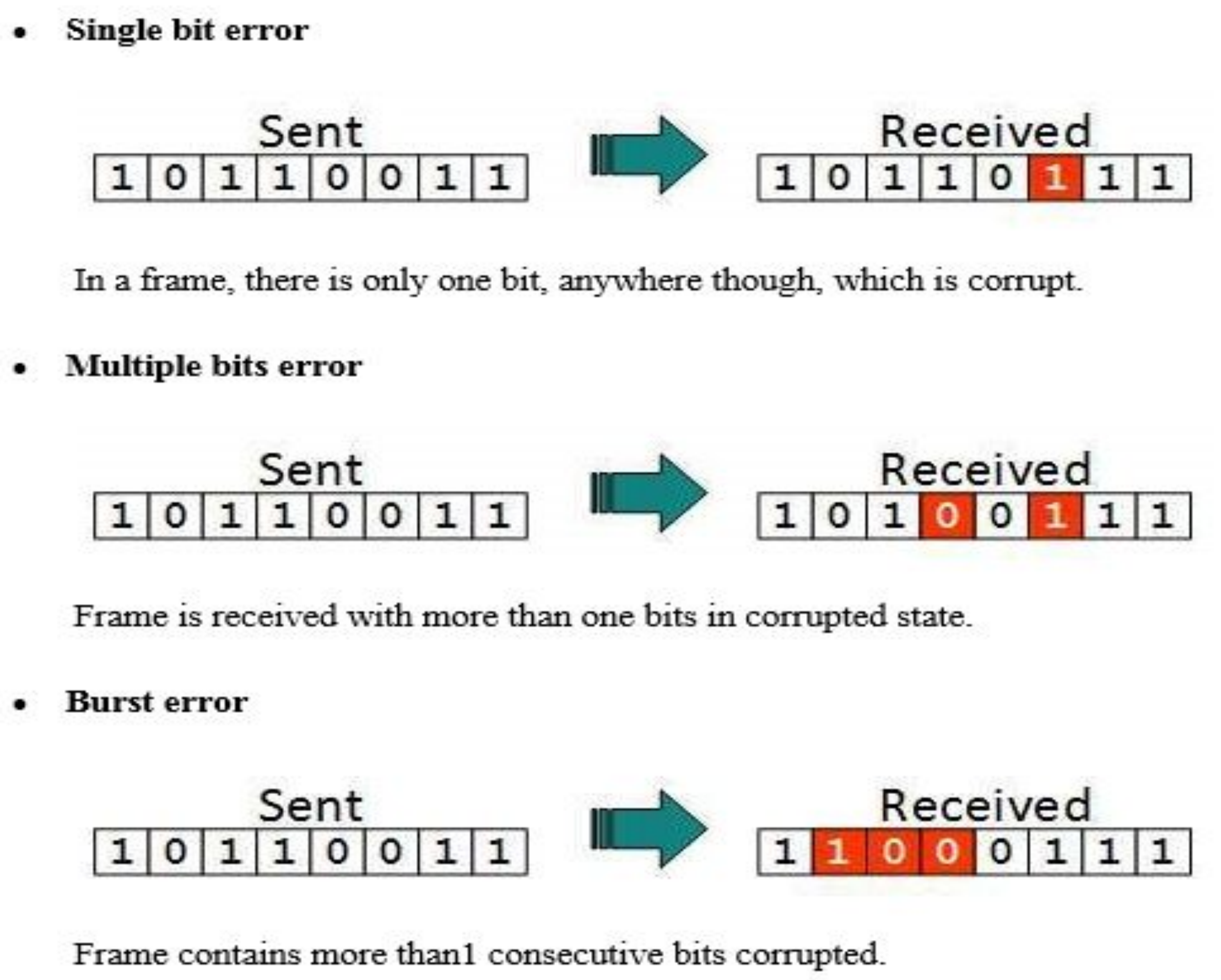
- Handles communication between upper and lower layers
- Takes the network protocol data and adds control information to help deliver the packet to the destination

## MAC

- Constitutes the lower sublayer of the data link layer
- Implemented by hardware, typically in the computer NIC
- Two primary responsibilities:
  - Data encapsulation
  - Media access control

# 3.4. ERROR DETECTION AND CORRECTION TECHNIQUES

# Types of Error



- **Single bit error**

Sent

1 0 1 1 0 0 1 1

Received

1 0 1 1 0 1 1 1

In a frame, there is only one bit, anywhere though, which is corrupt.

- **Multiple bits error**

Sent

1 0 1 1 0 0 1 1

Received

1 0 1 0 0 1 1 1

Frame is received with more than one bits in corrupted state.

- **Burst error**

Sent

1 0 1 1 0 0 1 1

Received

1 1 0 0 0 1 1 1

Frame contains more than1 consecutive bits corrupted.

# Error Detection

- Basic approach used for error detection is the use of redundancy bits, where additional bits are added to facilitate detection of errors.
- Some popular techniques for error detection are:
  1. Single Parity check
  2. Two-dimensional Parity check
  3. Checksum
  4. Cyclic redundancy check

# 1. Simple Parity check

- Blocks of data from the source are subjected to a check bit or parity bit generator form, where a parity of :
- 1 is added to the block if it contains odd number of 1’s, and
- 0 is added if it contains even number of 1’s
- This scheme makes the total number of 1’s even, that is why it is called even parity checking.

# How is the **even parity** bit generated?

## Total number of '1's should be even.

If the byte that we want to transmit is: 1 0 1 0 1 1 0 1

- **Step 1: count the number of 1's in the byte.**
  - **Answer: 5**
- **Step 2: compute the parity value.** 1 0 1 0 1 1 0 1 1
  - **Since the total number of 1's is 5, the even parity bit will have a value of 1.**
- **If the number of bits are already even, the parity bit will be '0'.**

# How is the **odd parity** bit generated?

## Total number of ‘1’s should be odd.

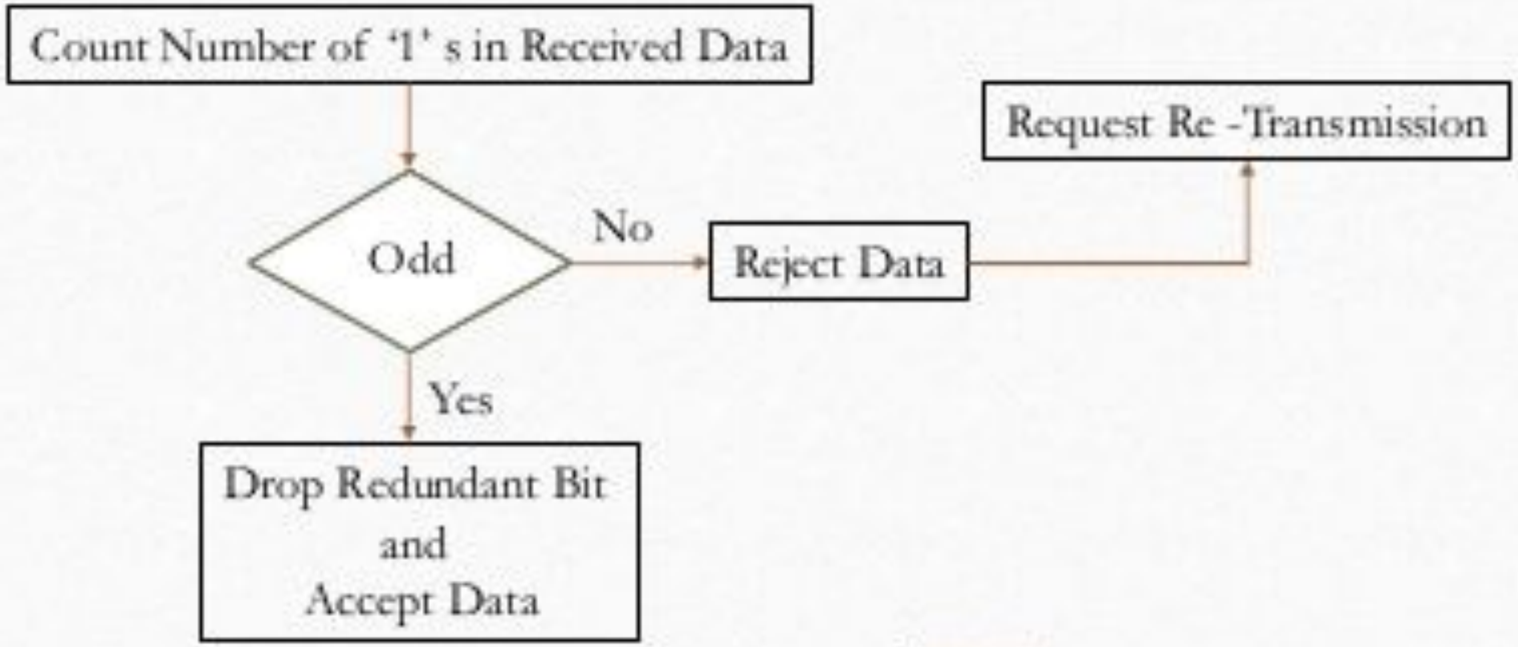
If the byte that we want to transmit is: 1 0 1 0 1 1 0 0

- **Step 1:** count the number of 1’s in the byte.
  - **Answer: 4**
- **Step 2:** compute the parity value. 1 0 1 0 1 1 0 0 1
  - **Since the total number of 1’s is 4, the odd parity bit will have a value of 1.**
- **If the number of bits are already odd, the parity bit will be ‘0’.**

# Single Parity check(Cont.)

## **Parity Detection**

- **In Odd parity Concept**



Count Number of '1' s in Received Data

Odd

No

Reject Data

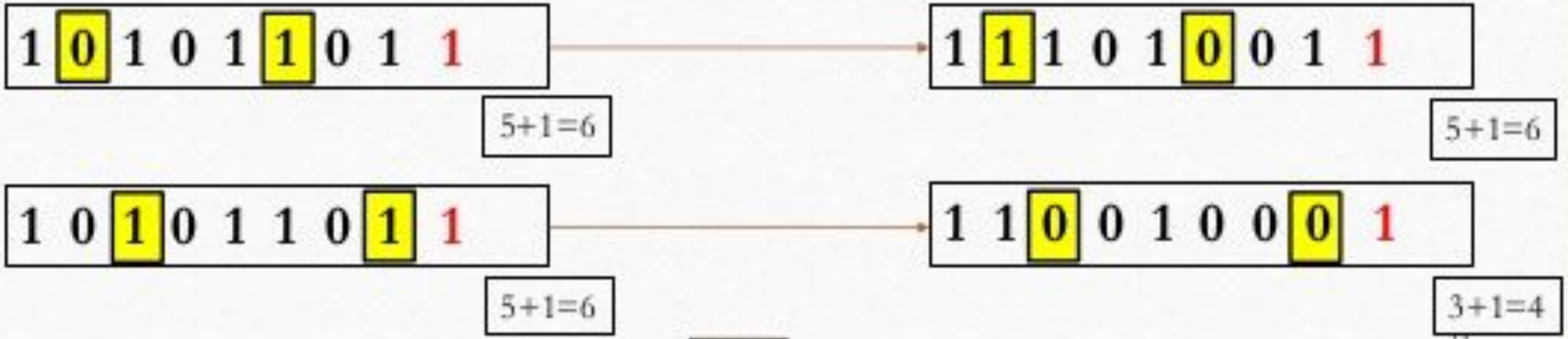
Request Re -Transmission

Yes

Drop Redundant Bit
and
Accept Data

# Drawbacks of Single Parity Check

- Only can detect single bit errors ; Single bit errors are rare.
- Can not detect errors, if 2 bits are **interchanged.**



1 0 1 0 1 1 0 1 1 → 1 1 1 0 1 0 0 1 1

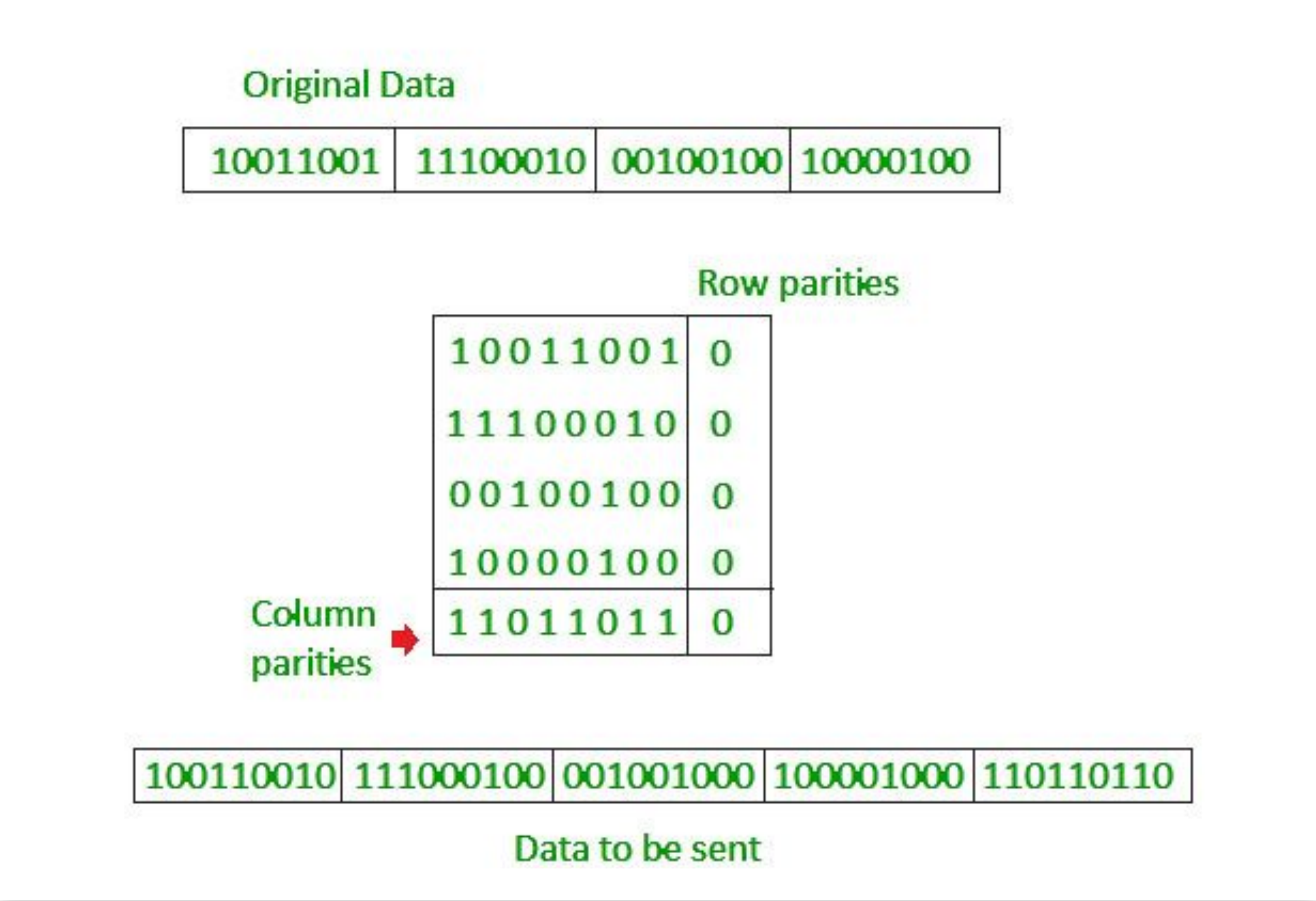
5+1=6 5+1=6

1 0 1 0 1 1 0 1 1 → 1 1 0 0 1 0 0 0 1

5+1=6 3+1=4

# 2. Two-dimensional Parity check

- Parity check bits are calculated for each row, which is equivalent to a simple parity check bit.
- Parity check bits are also calculated for all columns, then both are sent along with the data.
- At the receiving end these are compared with the parity bits calculated on the received data.



Original Data

|  |  |  |  |
|---|---|---|---|
| 10011001 | 11100010 | 00100100 | 10000100 |

Row parities

|  |  |  |
|---|---|---|
|  | 1 0 0 1 1 0 0 1 | 0 |
|  | 1 1 1 0 0 0 1 0 | 0 |
|  | 0 0 1 0 0 1 0 0 | 0 |
|  | 1 0 0 0 0 1 0 0 | 0 |
| Column parities | 1 1 0 1 1 0 1 1 | 0 |

|  |  |  |  |  |
|---|---|---|---|---|
| 100110010 | 111000100 | 001001000 | 100001000 | 110110110 |

Data to be sent

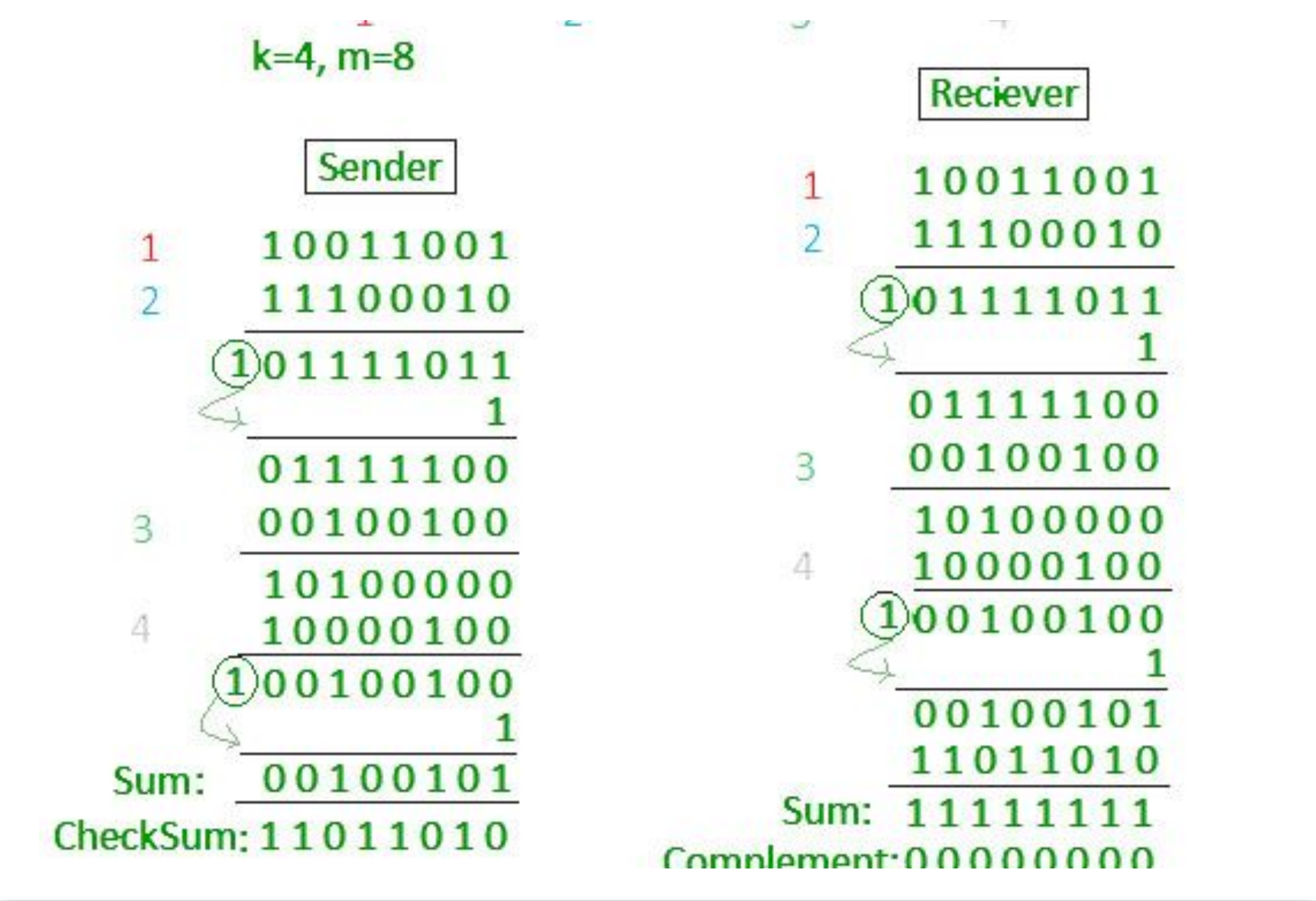
# 3. Checksum

- 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.

# Question

- Consider the data unit to be transmitted is-
  - 10011001111000100010010010000100
- Consider 8 bit checksum is used.
- K=4, m=8

| | | | |
|---|---|---|---|
| 10011001 | 11100010 | 00100100 | 10000100 |
| 1 | 2 | 3 | 4 |



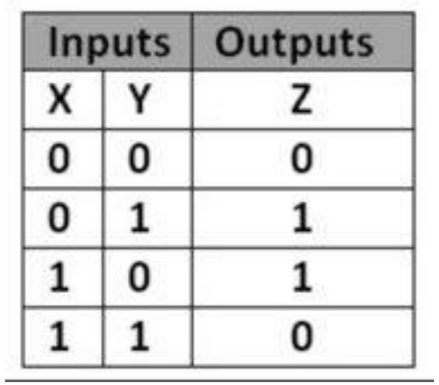
k=4, m=8

```
Sender

1          1 0 0 1 1 0 0 1
2          1 1 1 0 0 0 1 0
          ------------------
         1 0 1 1 1 1 0 1 1
                         1
          ------------------
           0 1 1 1 1 1 0 0
3          0 0 1 0 0 1 0 0
          ------------------
           1 0 1 0 0 0 0 0
4          1 0 0 0 0 1 0 0
          ------------------
         1 0 0 1 0 0 1 0 0
                         1
          ------------------
Sum:       0 0 1 0 0 1 0 1
          ------------------
CheckSum:  1 1 0 1 1 0 1 0
```

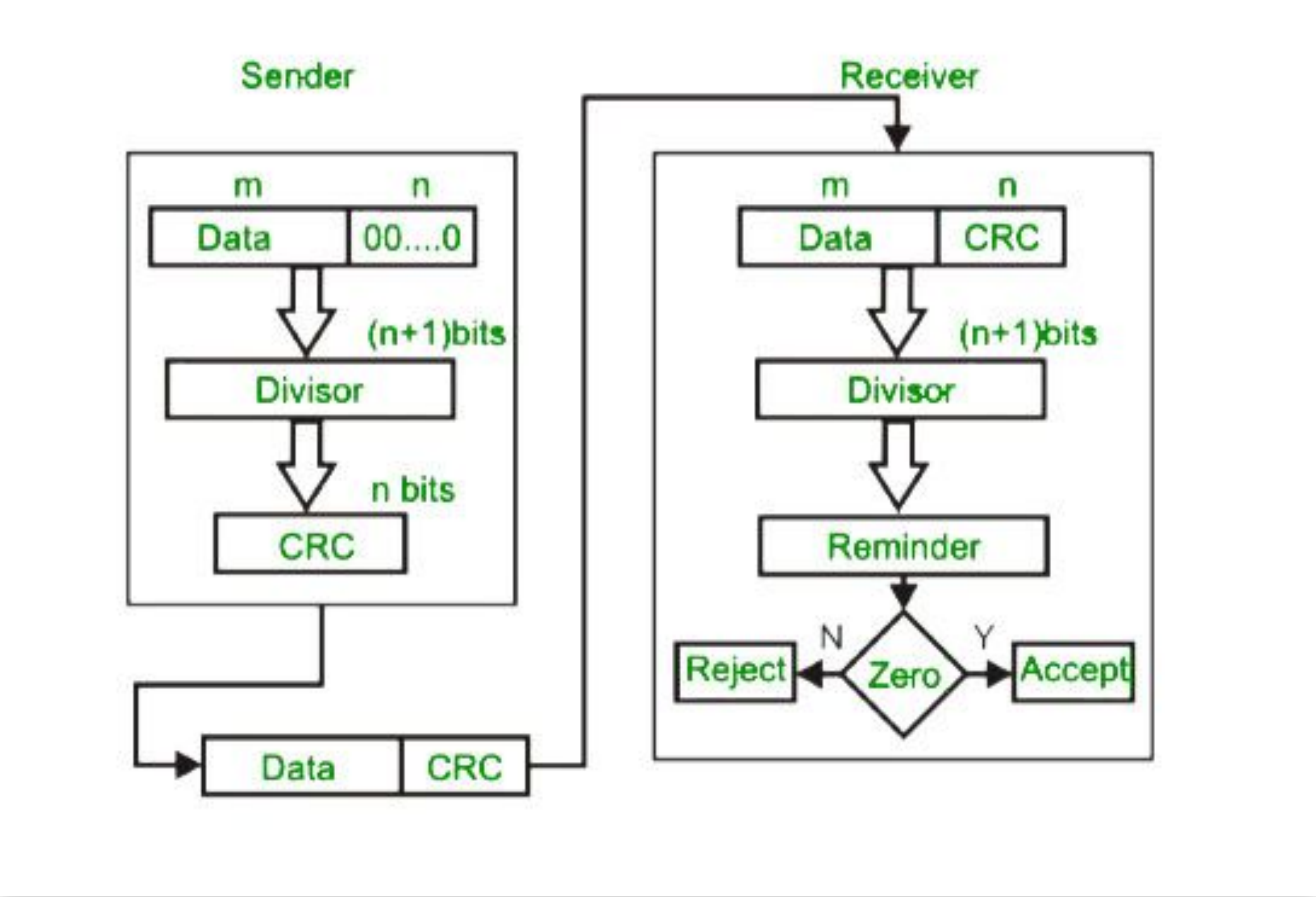
```
Reciever

1            1 0 0 1 1 0 0 1
2            1 1 1 0 0 0 1 0
            ------------------
           1 0 1 1 1 1 0 1 1
                           1
            ------------------
             0 1 1 1 1 1 0 0
3            0 0 1 0 0 1 0 0
            ------------------
             1 0 1 0 0 0 0 0
4            1 0 0 0 0 1 0 0
            ------------------
           1 0 0 1 0 0 1 0 0
                           1
            ------------------
             0 0 1 0 0 1 0 1
             1 1 0 1 1 0 1 0
            ------------------
Sum:         1 1 1 1 1 1 1 1
Complement:  0 0 0 0 0 0 0 0
```

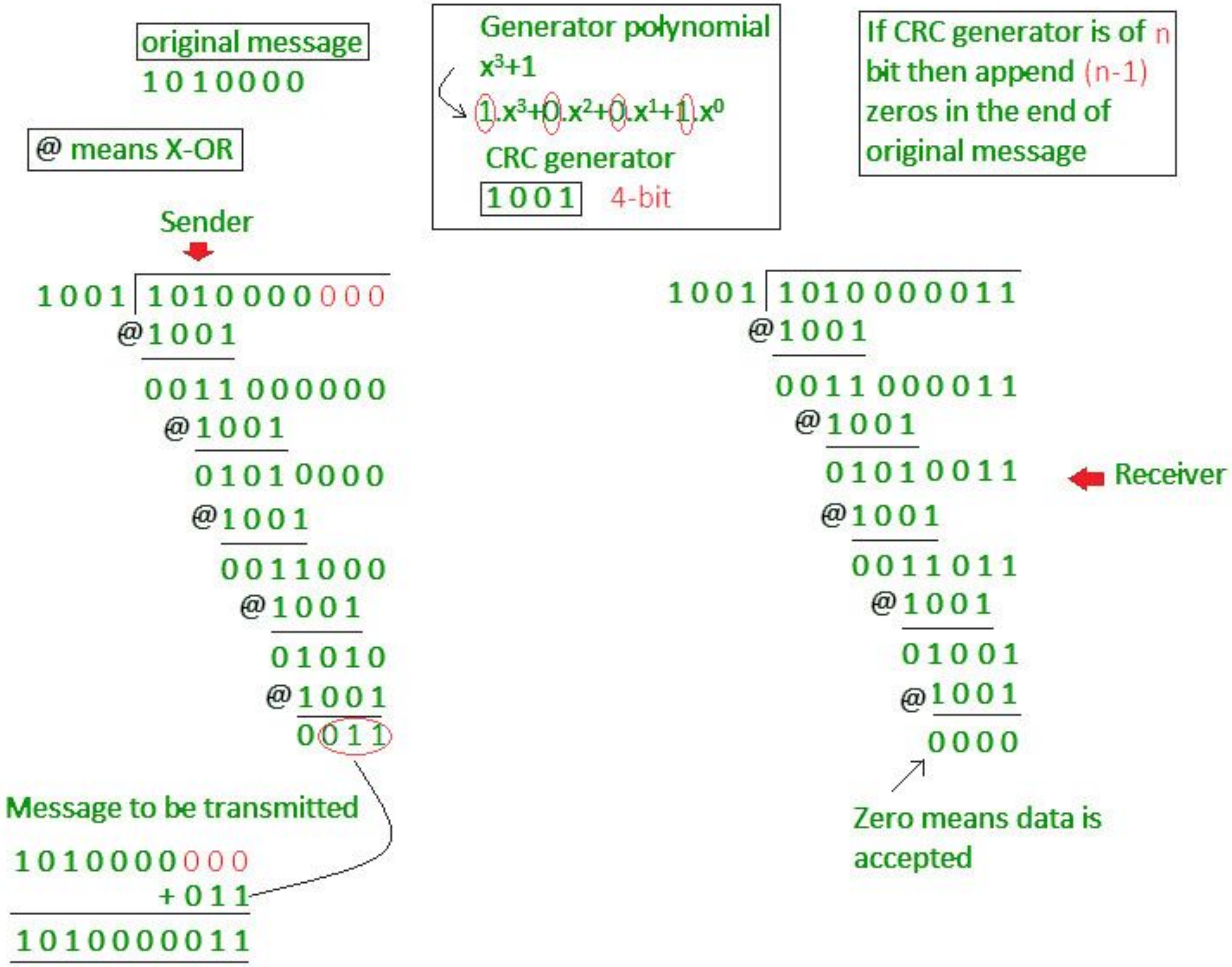
# 4. Cyclic redundancy check (CRC)

- 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.
-



| Inputs | | Outputs |
|---|---|---|
| X | Y | Z |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |


Sender
Receiver
m
n
Data
00....0
(n+1)bits
Divisor
n bits
CRC
m
n
Data
CRC
(n+1)bits
Divisor
Reminder
N
Y
Reject
Zero
Accept
Data
CRC



original message

1 0 1 0 0 0 0

@ means X-OR

Generator polynomial

$x^3+1$

$1.x^3+0.x^2+0.x^1+1.x^0$

CRC generator

1 0 0 1   4-bit

If CRC generator is of n bit then append (n-1) zeros in the end of original message

Sender

```
1 0 0 1 | 1 0 1 0 0 0 0 0 0 0
         @1 0 0 1
          0 0 1 1 0 0 0 0 0 0
            @1 0 0 1
              0 1 0 1 0 0 0 0
               @1 0 0 1
                  0 0 1 1 0 0 0
                    @1 0 0 1
                      0 1 0 1 0
                     @1 0 0 1
                      0 0 1 1
```

Message to be transmitted

```
1 0 1 0 0 0 0 0 0 0
          + 0 1 1
-------------------
1 0 1 0 0 0 0 0 1 1
```

```
1 0 0 1 | 1 0 1 0 0 0 0 0 1 1
         @1 0 0 1
          0 0 1 1 0 0 0 0 1 1
            @1 0 0 1
              0 1 0 1 0 0 1 1
               @1 0 0 1
                  0 0 1 1 0 1 1
                    @1 0 0 1
                      0 1 0 0 1
                     @1 0 0 1
                        0 0 0 0
```

Receiver

Zero means data is accepted

# Error Correction

- error correction can be done in two ways:
- **Backward Error Correction** When the receiver detects an error in the data received, it requests back the sender to retransmit the data unit.
- **Forward Error Correction** When the receiver detects some error in the data received, it executes error-correcting code, which helps it to auto-recover and to correct some kinds of errors.
- The first one, Backward Error Correction, is simple and can only be efficiently used where retransmitting is not expensive. For example, fiber optics. But in case of wireless transmission retransmitting may cost too much. In the latter case, Forward Error Correction is used.
- To correct the error in data frame, the receiver must know exactly which bit in the frame is corrupted. To locate the bit in error, redundant bits are used as parity bits for error detection.For example, we take ASCII words (7 bits data), then there could be 8 kind of information we need: first seven bits to tell us which bit is error and one more bit to tell that there is no error.
- For m data bits, r redundant bits are used. r bits can provide 2r combinations of information. In m+r bit codeword, there is possibility that the r bits themselves may get corrupted. So the number of r bits used must inform about m+r bit locations plus no-error information, i.e. m+r+1.
-

HAMMING CODE : REFER SIT

ezexplanation.com

# 3.5 CHANNEL ALLOCATION TECHNIQUE

# RANDOM ACCESS PROTOCOLS

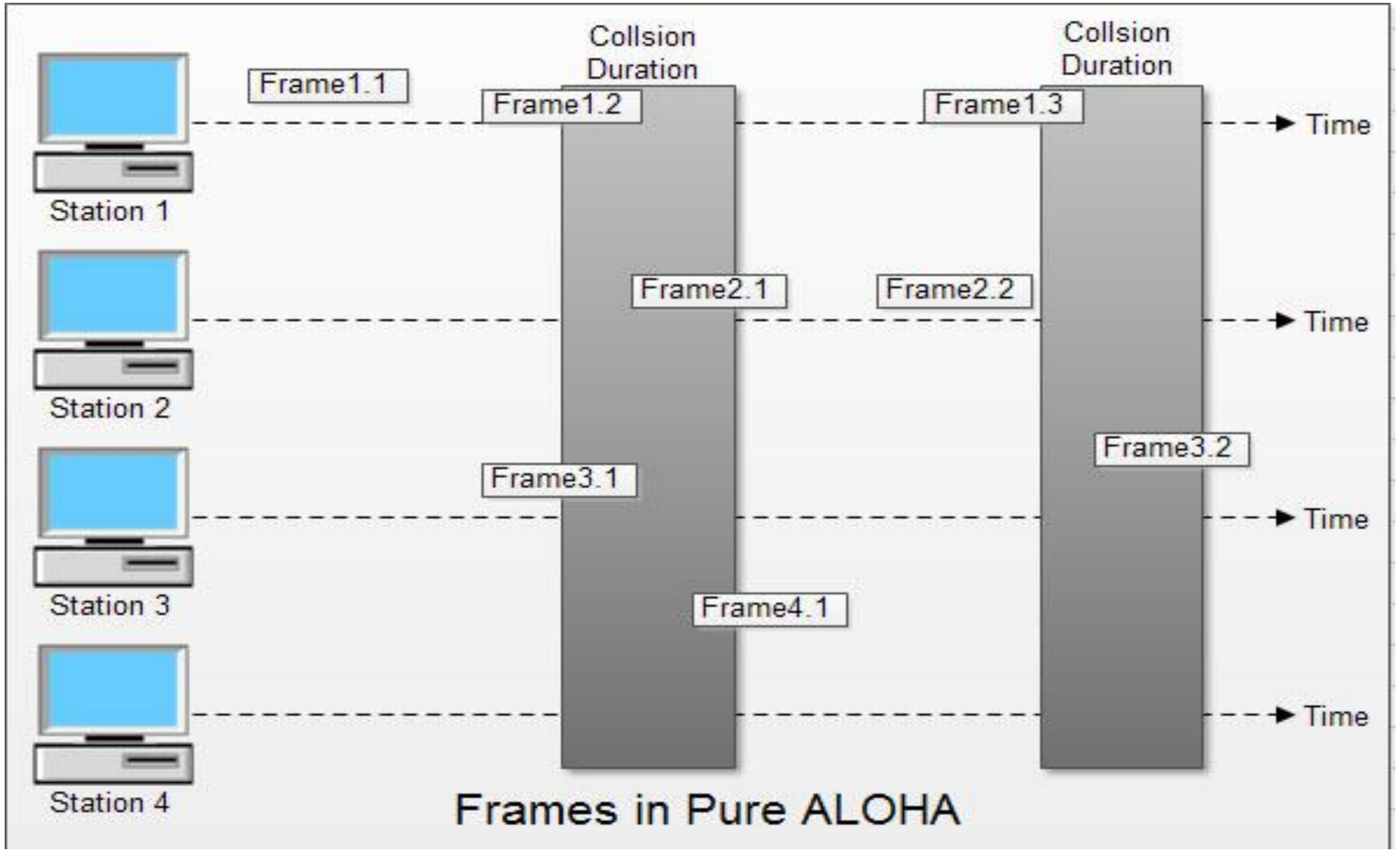
- system for **a shared communication** Networks channel.
- requires a method of **handling collisions**
- occur when **two or more systems** attempt to transmit on the **channel** at the **same time.**
- In the **ALOHA system**, a node transmits **whenever data is available** to send.
- If **another node transmits at the same time**, a collision occurs, and the frames that were transmitted are lost.

**There are two different version/types of ALOHA:**

(i) PureALOHA
(ii) Slotted ALOHA

## Pure ALOHA

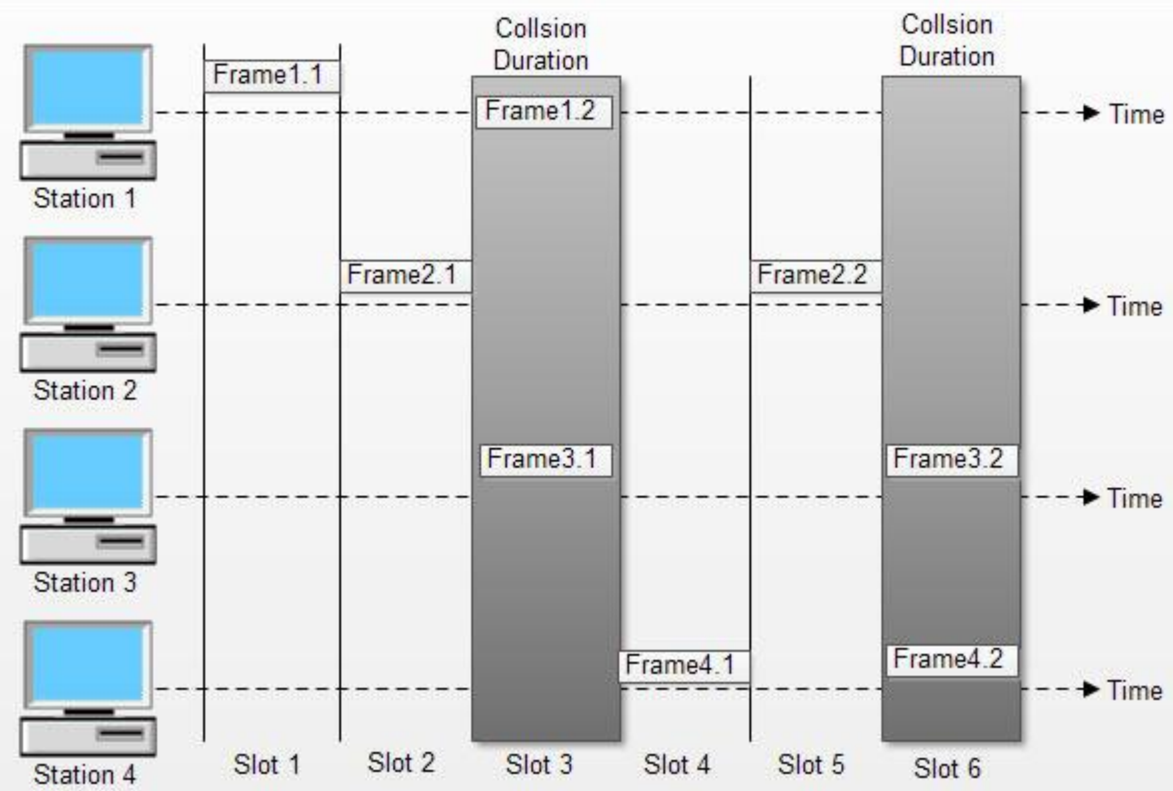
- In pure ALOHA, **whenever any station transmits a frame**, it expects the **acknowledgement** from the receiver.
- If **acknowledgement is not received** within specified time, the station assumes that the **frame has been destroyed**.
- If the **frame is destroyed** because of collision the station **waits** for a **random amount of time** and **sends it again.**
- This **waiting time must be random** otherwise same **frames will collide again and again.**
- Figure shows an example of frame collisions in pure ALOHA.



Frames in Pure ALOHA

# Slotted ALOHA

- Slotted ALOHA was **invented to improve the efficiency of pure ALOHA** as chances of collision in pure ALOHA are very high.
- In slotted ALOHA, **the time of the shared channel is divided into discrete intervals called slots.**
- The stations can send **a frame only at the beginning of the slot** and only **one frame is sent in each slot**.

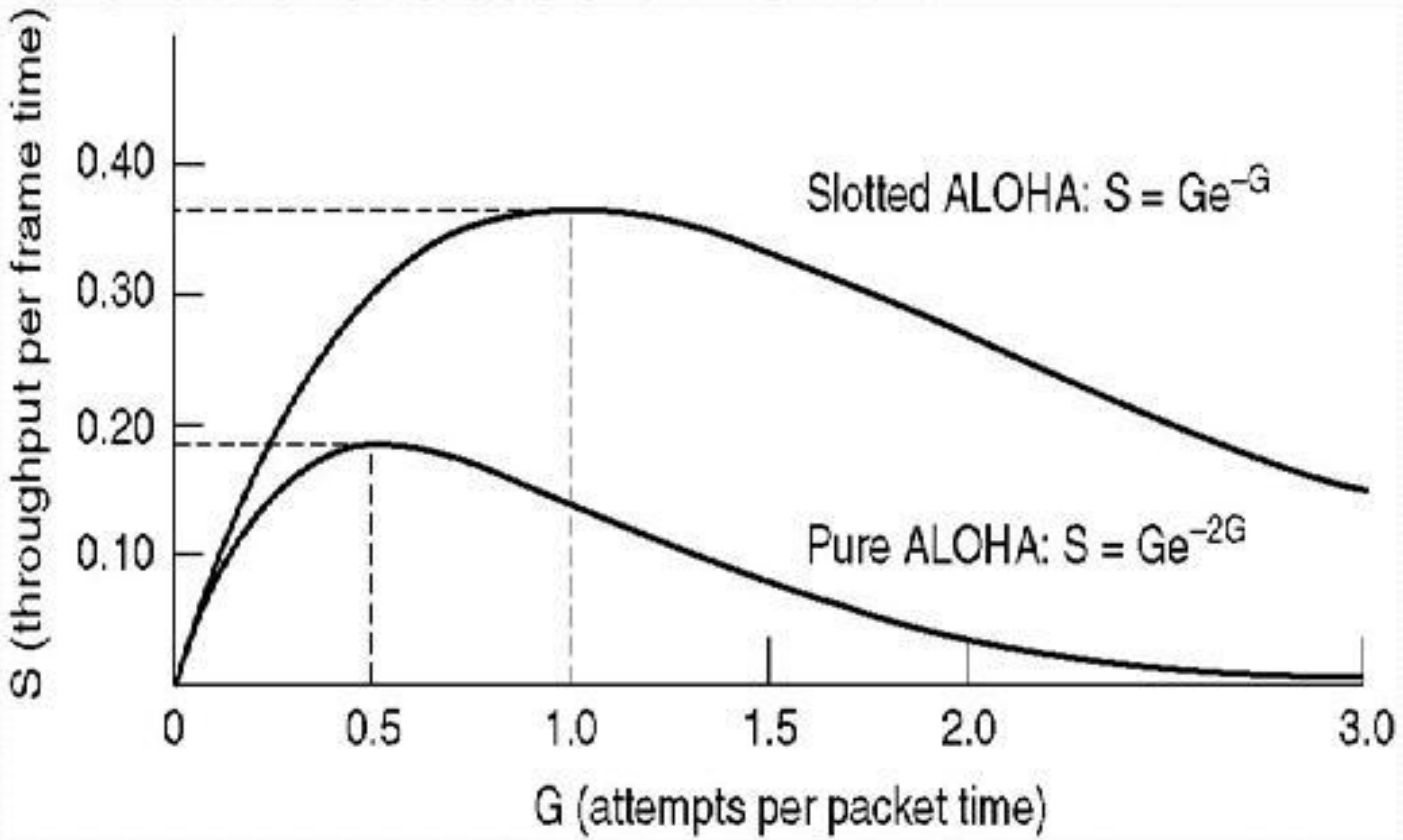


Frames in Slotted ALOHA

- In slotted ALOHA, **if any station is not able to place the frame** onto the channel at the beginning of the slot
- then the station has to **wait until the beginning of the next time slot**.
- In slotted ALOHA, there is **still a possibility of collision** if **two stations try to send at the beginning of the same time slot**
- Slotted ALOHA still has an edge over pure ALOHA as **chances of collision are reduced to one-half.**

# Key Differences Between Pure ALOHA and Slotted ALOHA

- Pure ALOHA was introduced by Norman and his associates at the university of Hawaii in 1970.
- On the other hand, Slotted ALOHA was introduced by Roberts in 1972.
- In pure ALOHA, whenever a station has data to send it transmits it without waiting whereas, in slotted ALOHA a user wait till the next time slot beings to transmit the data.
- In pure ALOHA the time is continuous whereas, in Slotted ALOHA the time is discrete and divided into slots.
- In pure ALOHA the probability of successful transmission is S=G*e^-2G. On the other hand, in slotted ALOHA the probability of successful transmission is S=G*e^-G.
- The maximum throughput occurs at G=1/2 which is 18 % whereas, the maximum throughput occurs at G=1 which is 37%.


S (throughput per frame time)
0.40
0.30
0.20
0.10
Slotted ALOHA: $S = Ge^{-G}$
Pure ALOHA: $S = Ge^{-2G}$
0
0.5
1.0
1.5
2.0
3.0
G (attempts per packet time)

# 1. CSMA, or listen with random access carrier

- CSMA (Carrier Sense Multiple Access) is to **listen to the channel before transmitting.**
- This significantly **reduces the risk of collision**, but **does not eliminate them completely.**
- If more remote stations, **a coupler does not detect the transmission** of a frame, and there may be **signal superposition**.
- it is necessary to subsequently **retransmit lost frames.**

# 2. 802.3 CSMA / CD

- **(Carrier Sense Multiple Access / Collision Detection)**
- a set of rules determining **how network devices respond when two devices attempt to use a data channel simultaneously** (called a *collision*).
- networks use CSMA/CD to **physically monitor the traffic** on the line at participating stations.
- If **no transmission** is taking place at the time, **the particular station can transmit.**

- If **two stations** attempt to **transmit simultaneously**, this **causes a collision**, which is **detected by all participating stations.**
- After a **random time interval**, the stations that collided **attempt to transmit again.**
- If **another collision occurs**, the **time intervals from which the random waiting time is selected are increased step by step.**
- This is known as **exponential back off**

# 3. CSMA / CA

- **CSMA/CA protocol is used in wireless networks because they cannot detect the collision so the only solution is collision avoidance.**
- CSMA/CA avoids the collisions using three basic techniques.
- Interframe space
- Contention window
- Acknowledgements

- **Interframe Space (IFS)**
- Whenever the **channel is found idle**, the station does not transmit immediately.
- It **waits** for a period of time called **Interframe space (IF**S).
- it may be possible that same distant station **may have already started transmitting**
- and the signal of that distant station has not yet reached other stations.

## Contention Window

- Contention window is an **amount of time divided into slots.**
- A **station that is ready to send** chooses a **random number of slots as its wait time.**
- In contention window the station **needs to sense the channel after each time slot.**

## Acknowledgement

- Despite all the precautions, **collisions may occur and destroy the data**.
- The **positive acknowledgment and the time-out timer** can help guarantee that receiver has received the frame.

# 4. IEEE 802.5 Token Ring

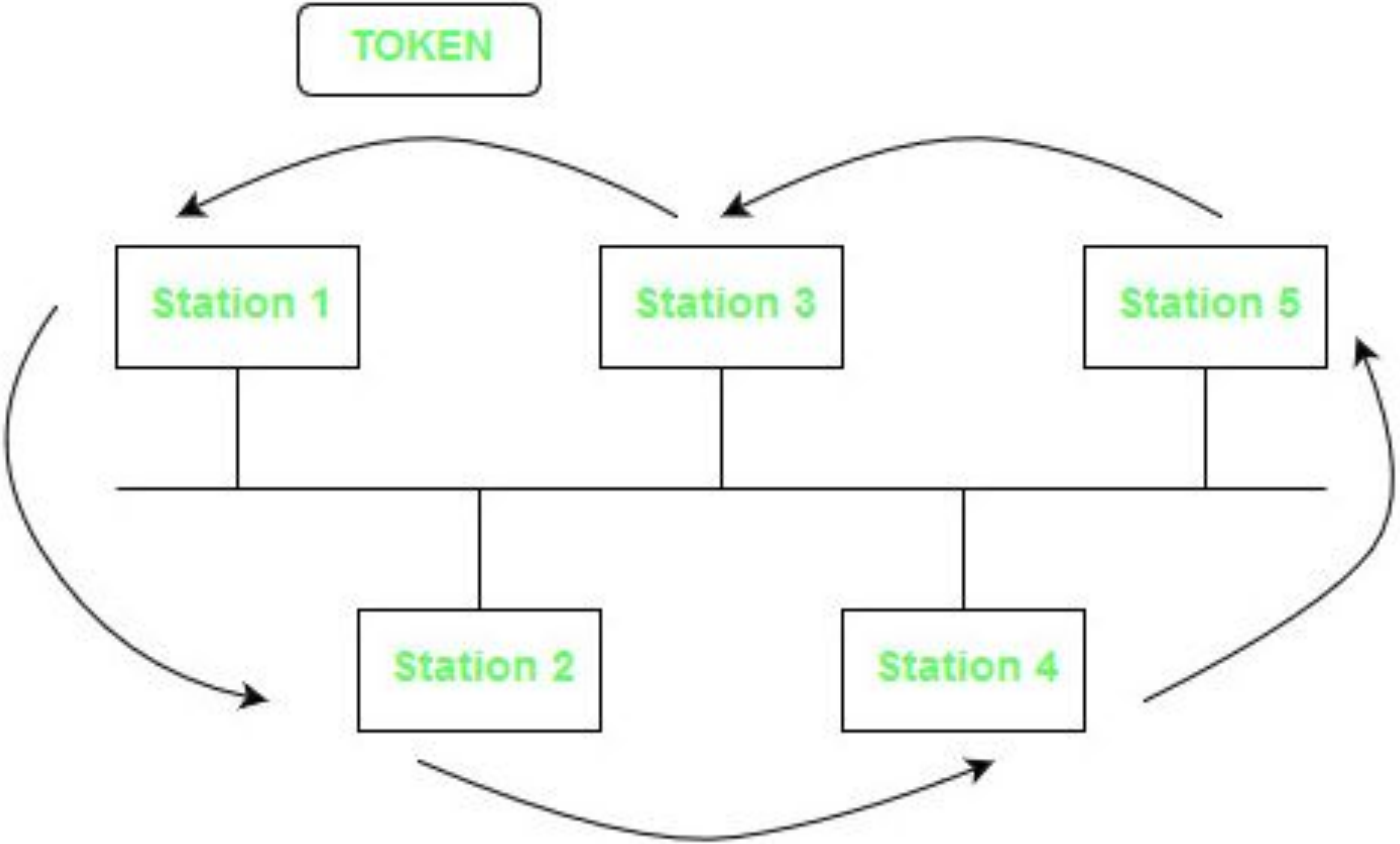
- In token ring special bit pattern, called the token, circulates around the ring whenever all stations are idle.
- When a station wants to transmit a frame, it is required to seize the token and remove it from the ring before transmitting.
- Because there is only one token, only one station can transmit at a given instant, thus solving the channel access problem the same way token bus solves it.
- A station may hold the token for the token holding time, which is **10 ms** unless an installation sets a different value.
- After all frames transmitted or the transmission of another frame would exceed the token holding time, the station regenerates the token.

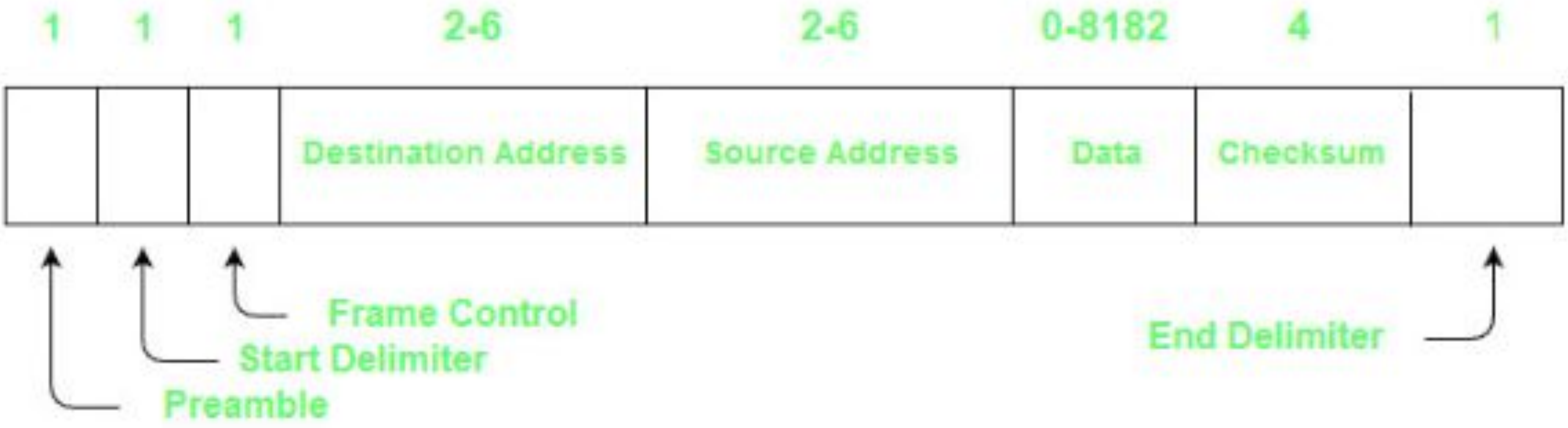
- In the token-passing method, **the stations in a network are organized in a logical ring.**
- In other words, for each station, there is **a predecessor and a successor.**
- The predecessor is the station which is logically before the station in the ring;
- the successor is the station which is after the station in the ring.
- The **current station** is the one that is **accessing the channel now.**
- The right to this access has been passed from the predecessor to the current station.

- The **right will be passed to the successor** when the current station has no more data to send.
- In this method, **a special packet called a token circulates through the ring**.
- The **possession of the token** gives the station the **right to access the channel** and send its data.
- When a station has some data to send, it waits until it receives the token from its predecessor
- It then holds the token and sends its data. When the station has no more data to send,
- it releases the token, passing it to the next logical station in the ring. The station cannot send data until it receives the token again in the next round.

# 5. 802.4 TOKEN BUS

- **Token Bus (IEEE 802.4)** is a popular standard for the token passing LANs.
- In a token bus LAN, the physical media is a bus or a tree and a logical ring is created using coaxial cable.
- The token is passed from one user to other in a sequence (clockwise or anticlockwise).
- Each station knows the address of the station to its “left” and “right” as per the sequence in the logical ring.
- A station can only transmit data when it has the token. The working of token bus is somewhat similar to Token Ring.


TOKEN
Station 1
Station 3
Station 5
Station 2
Station 4



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 2-6 | 2-6 | 0-8182 | 4 | 1 |
| | | | Destination Address | Source Address | Data | Checksum | |

# Frame Format:

- **Preamble –** It is used for bit synchronization. It is 1 byte field.
- **Start Delimiter –** These bits marks the beginning of frame. It is 1 byte field.
- **Frame Control –** This field specifies the type of frame – data frame and control frames. It is 1 byte field.
- **Destination Address –** This field contains the destination address. It is 2 to 6 bytes field.
- **Source Address –** This field contains the source address. It is 2 to 6 bytes field.
- **Data –** If 2 byte addresses are used than the field may be upto 8182 bytes and 8174 bytes in case of 6 byte addresses.
- **Checksum –** This field contains the checksum bits which is used to detect errors in the transmitted data. It is 4 bytes field.
- **End Delimiter –** This field marks the end of frame. It is 1 byte field.

# 3.7. WIRELESS LAN

# Bluetooth

- Bluetooth is a wireless technology standard for exchanging data over short distances
- using short-wavelength UHF radio waves from 2.4 to 2.485 GHz
- Invented by telecom vendor Ericsson in 1994, it was originally conceived as a wireless alternative to RS-232 data cables.
- It can connect several devices, overcoming problems of synchronization.
- Bluetooth is managed by the Bluetooth Special Interest Group (SIG), which has more than 20,000 member companies in the areas of telecommunication, computing, networking, and consumer electronics.
- Bluetooth was standardized as IEEE 802.15.1, but the standard is no longer maintained.
- To be marketed as a Bluetooth device, it must be qualified to standards defined by the SIG.
- A network of patents is required to implement the technology, which is licensed only for that qualifying device.

# 3.8 OVERVIEW

# Virtual Circuit Switching

- Virtual circuit switching is a packet switching methodology whereby a path is established between the source and the final destination through which all the packets will be routed during a call.
- This path is called a virtual circuit because to the user, the connection appears to be a dedicated physical circuit.
- However, other communications may also be sharing the parts of the same path.

- Before the data transfer begins, the source and destination identify a suitable path for the virtual circuit.
- All intermediate nodes between the two points put an entry of the routing in their routing table for the call.
- Additional parameters, such as the maximum packet size, are also exchanged between the source and the destination during call setup.
- The virtual circuit is cleared after the data transfer is completed.

- Virtual circuit packet switching is connection orientated.

Advantages of virtual circuit switching are:

- Packets are delivered in order, since they all take the same route;
- The overhead in the packets is smaller,
- The connection is more reliable
- Billing is easier,

- Examples of virtual circuit switching are X.25 and Frame Relay.

# Frame Relay

- In the 1980s, the X.25 networks were largely replaced by a new kind of network called frame relay.
- The essence of frame relay is that it is a connection-oriented network with no error control and no flow control.
- Because it was connection-oriented, packets were delivered in order (if they were delivered at all).
- The properties of frame relay are in-order delivery, no error control, and no flow control
- Its most important application is interconnecting LANs at multiple company offices.

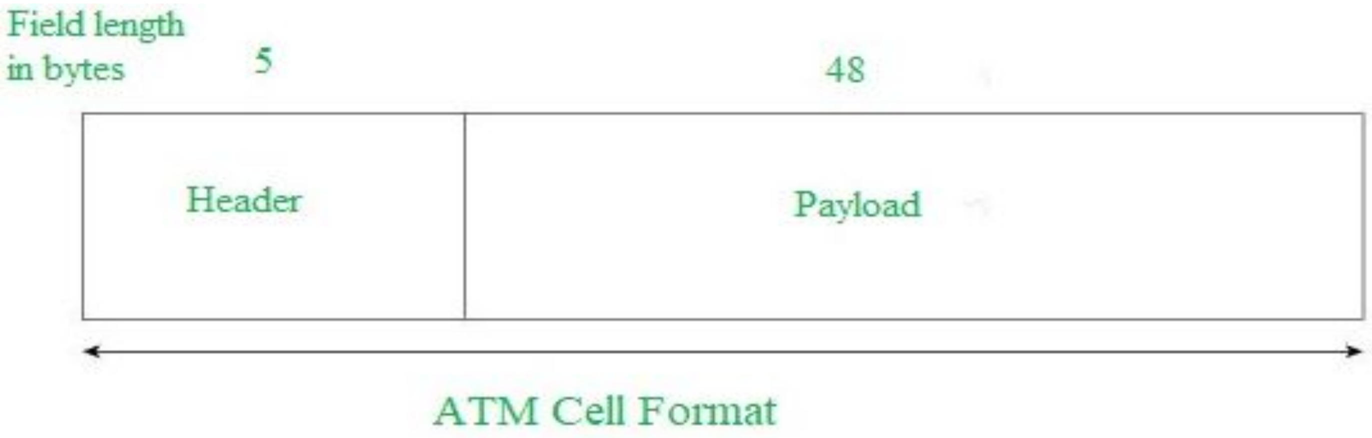
- Frame relay is a packet-switching telecommunication service designed for cost-efficient data transmission for intermittent traffic between local area networks (LANs)
- Frame relay is based on the older X.25 packet-switching technology that was designed for transmitting analog data such as voice conversations.
- Unlike X.25, which was designed for analog signals, frame relay is a fast packet technology, which means that the protocol does not attempt to correct errors.
- When an error is detected in a frame, it is simply dropped (that is, thrown away).

- It uses a technology called fast packet in which error checking does not occur in any intermediate node of the transmission but done at the ends.
- It makes it more efficient than X.25, and a higher process speed achieved (it can transmit over 2,044 Mbps).
- The frame relay networks are designed to operate efficiently at the user’s data rates upto 2 Mbps.

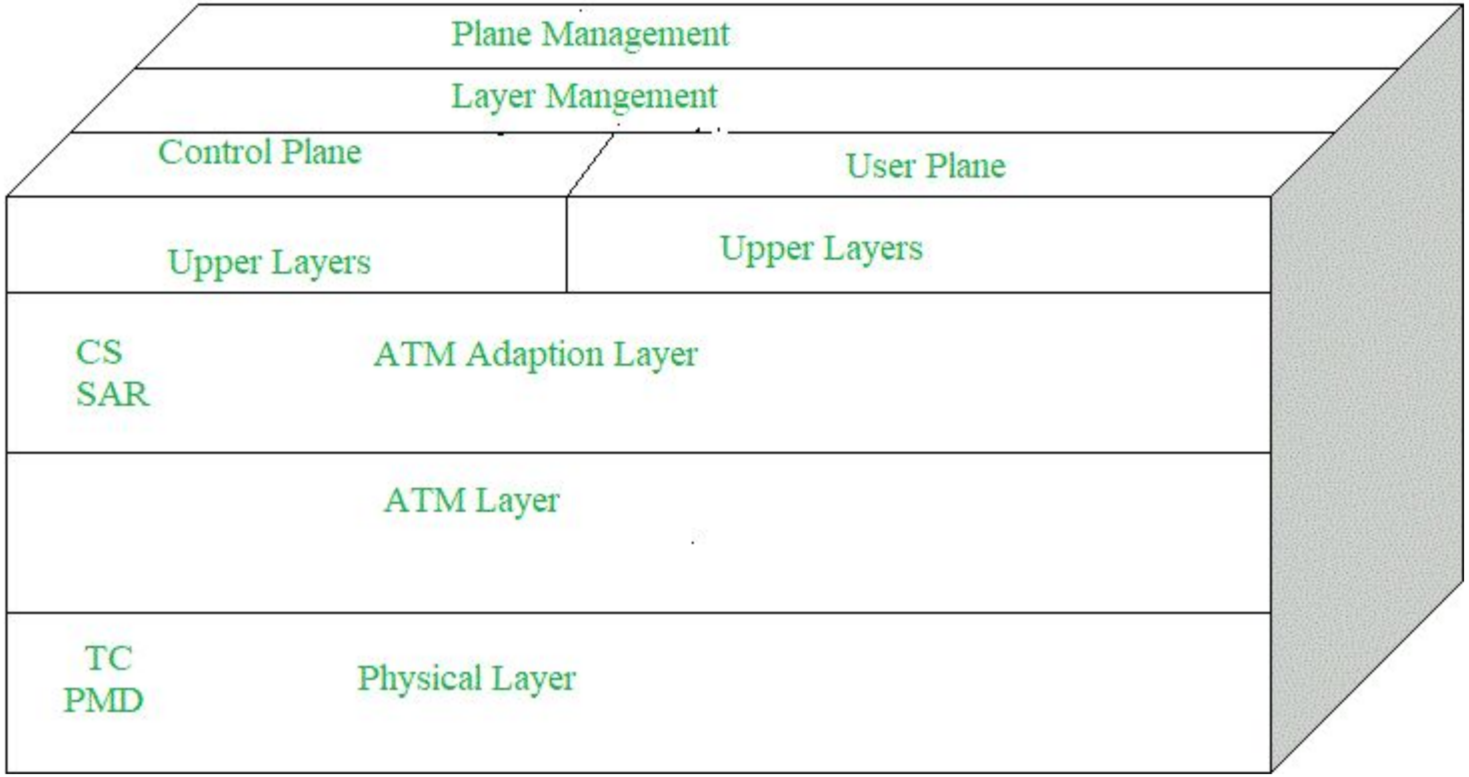
# ATM

- Asynchronous Transfer Mode (ATM):
  It is an International Telecommunication Union- Telecommunications Standards Section (ITU-T) efficient for call relay and it transmits all information including multiple service types such as data, video or voice which is conveyed in small fixed size packets called cells.
- Cells are transmitted asynchronously and the network is connection oriented.

- **ATM Cell Format –**
  **As information is transmitted in ATM in the form of fixed size units called cells**. As known already each cell is 53 bytes long which consists of 5 bytes header and 48 bytes payload.



# ATM Layers:


Plane Management
Layer Mangement
Control Plane
User Plane
Upper Layers
Upper Layers
CS
SAR
ATM Adaption Layer
ATM Layer
TC
PMD
Physical Layer

- ATM Adaption Layer (AAL) –
  It is meant for isolating higher layer protocols from details of ATM processes and prepares for conversion of user data into cells and segments it into 48-byte cell payloads.
- AAL protocol excepts transmission from upper layer services and help them in mapping applications, e.g., voice, data to ATM cells.
- Physical Layer –
  It manages the medium-dependent transmission
- Main functions are as follows:
  - It converts cells into a bit stream.
  - It controls the transmission and receipt of bits in the physical medium.
  - Looks for the packaging of cells into appropriate type of frames.

- ATM Layer –
  It handles transmission, switching, congestion control, cell header processing, sequential delivery, etc.

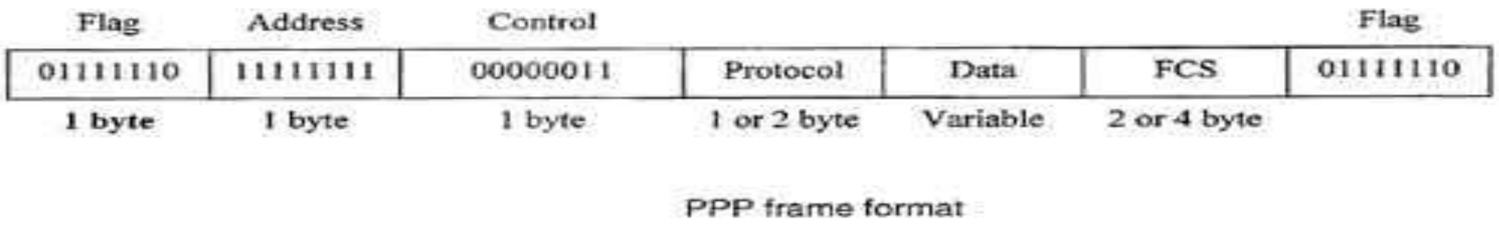
# 3.9. DLL PROTOCOL

# PPP – the Point-to-Point Protocol

- PPP is most commonly used data link protocol.
- It is used to **connect the one point to another.** This protocol offers several facilities:
- PPP **defines the format of the frame** to be exchanged between the devices.
- It defines link control protocol (LCP) for:-
- (a) Establishing the link between two devices.
- (b) Maintaining this established link.
- (c) Configuring this link.
- (d) Terminating this link after the transfer.

- It defines **how network layer data are encapsulated in data link frame.**
- PPP provides **error detection.**
- It also defines how two devices can authenticate each other.

# PPP Frame Format



| Flag | Address | Control |  |  |  | Flag |
|---|---|---|---|---|---|---|
| 01111110 | 11111111 | 00000011 | Protocol | Data | FCS | 01111110 |
| 1 byte | 1 byte | 1 byte | 1 or 2 byte | Variable | 2 or 4 byte |  |

PPP frame format

**Flag field**: Flag field marks the beginning and end of the PPP frame. Flag byte is 01111110. (1 byte).

**Address field**: This field is of 1 byte and is always 11111111. This address is the broadcast address *i.e.* all the stations accept this frame.

**Control field**: This field is also of 1 byte. The value is always 00000011 to show that the frame does not contain any sequence numbers and there is no flow control or error control.
**Protocol field**: This field specifies the kind of packet in the data field *i.e.* what is being carried in data field.

**Data field** It carries user data or other information.

**FCS field**: The frame checks sequence. It is either of 2 bytes or 4 bytes. It contains the checksum.

# HDLC

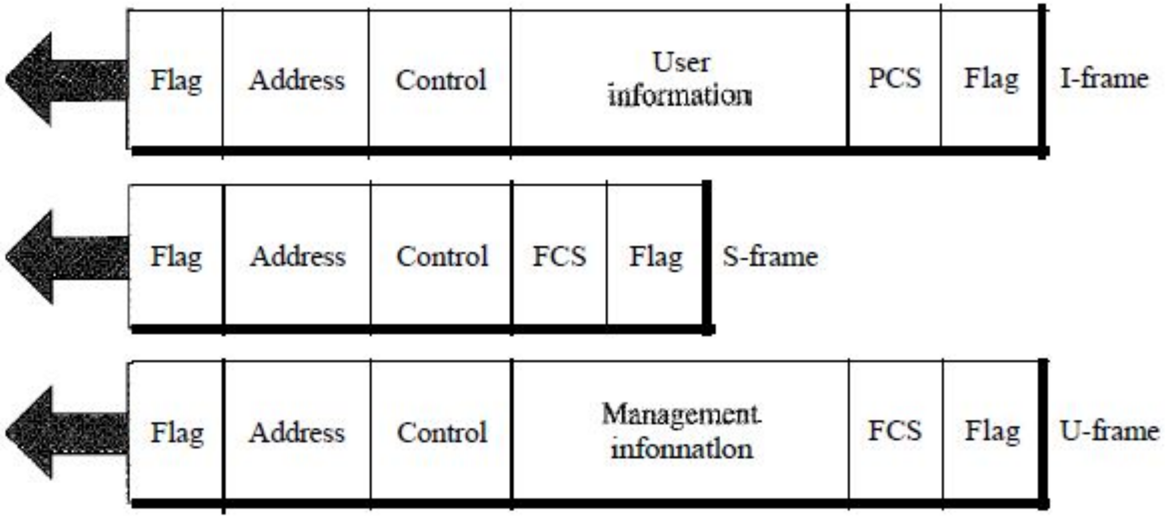
- high-level data link control (HDLC)
- ensures the error-free transmission of data to the proper destinations and controls the data transmission speed.

- A high-level data link control defines rules for transmitting data between network points.
- Data in an HDLC is organized into units called frames and is sent across networks to specified destinations.
- This is done using a frame delimiter or flag, which contains unique sequence of bits

- There are three types of HDLC frames:
- Information frames/User data (I-frames)
- Supervisory frames/Control data (S-frames)
- Unnumbered frames (U-frames)

- The common fields within an HDLC frame are:
- Flag
- Address
- Control information
- Frame check sequence

Figure 11.27 *HDLC frames*



- o Flag field.

identifies both the beginning and the end of a frame

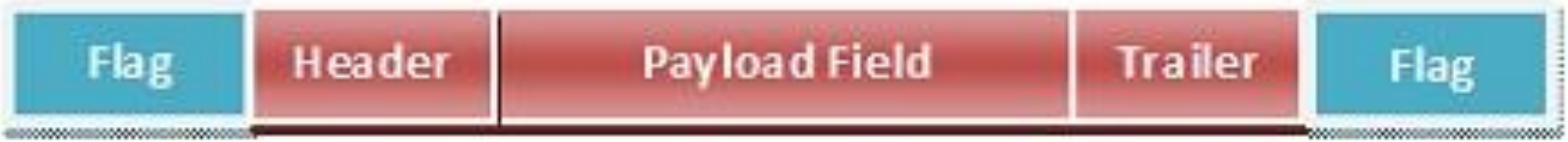
- o Address field. The second field of an HDLC frame contains the address of the secondary station
- o Control field. The control field is a 1- or 2-byte segment of the frame used for flow and error control.
- o Information field. The information field contains the user's
- o FCS field. The frame check sequence (FCS) is the HDLC error detection field.

# 3.3 FRAMING

- 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.

# 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.


Flag
Header
Payload Field
Trailer
Flag

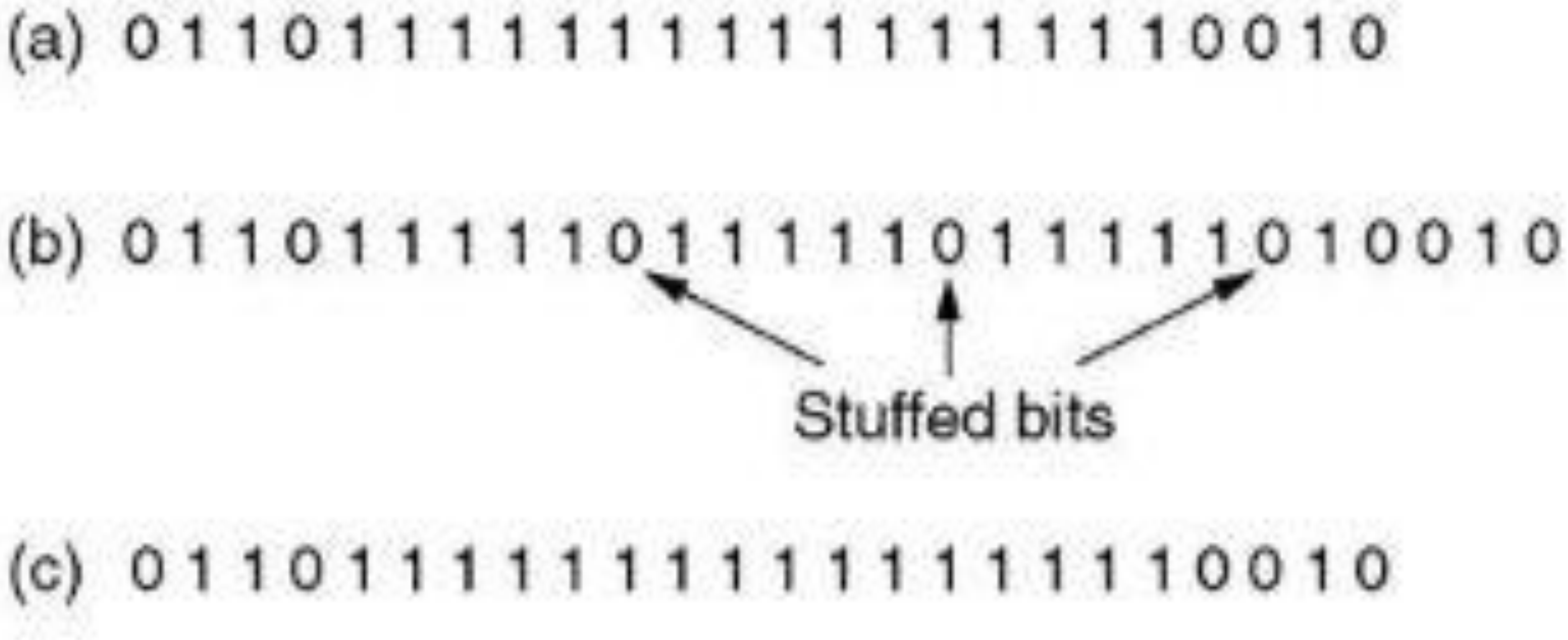
# METHODS OF FRAMING

- Bit stuffing
- Flag byte with Byte Stuffing
- Character Count

# Bit stuffing:

- Allows frame to contain arbitrary number of bits and arbitrary character size.
- The frames are separated by separating flag.
- Each frame begins and ends with a special bit pattern, 01111110 called a flag byte.
- When five consecutive l's are encountered in the data, it automatically stuffs a '0' bit into outgoing bit stream.
- In this method, frames contain an arbitrary number of bits and allow character codes with an arbitrary number of bits per character.
- In this case, each frame starts and ends with a special bit pattern, 01111110.

- In the data a 0 bit is automatically stuffed into the outgoing bit stream whenever the sender's data link layer finds five consecutive 1s.
- This bit stuffing is similar to byte stuffing, in which an escape byte is stuffed into the outgoing character stream before a flag byte in the data.
- When the receiver sees five consecutive incoming i bits, followed by a o bit, it automatically destuffs (i.e., deletes) the 0 bit.

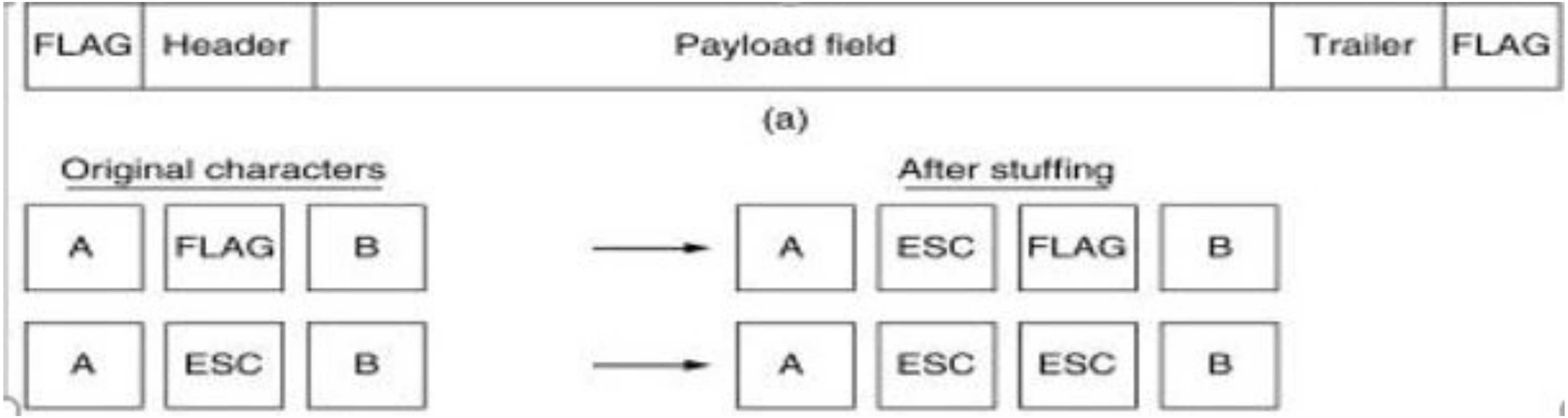


(a) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0
(b) 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0
Stuffed bits
(c) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

**Fig1: Bit stuffing**

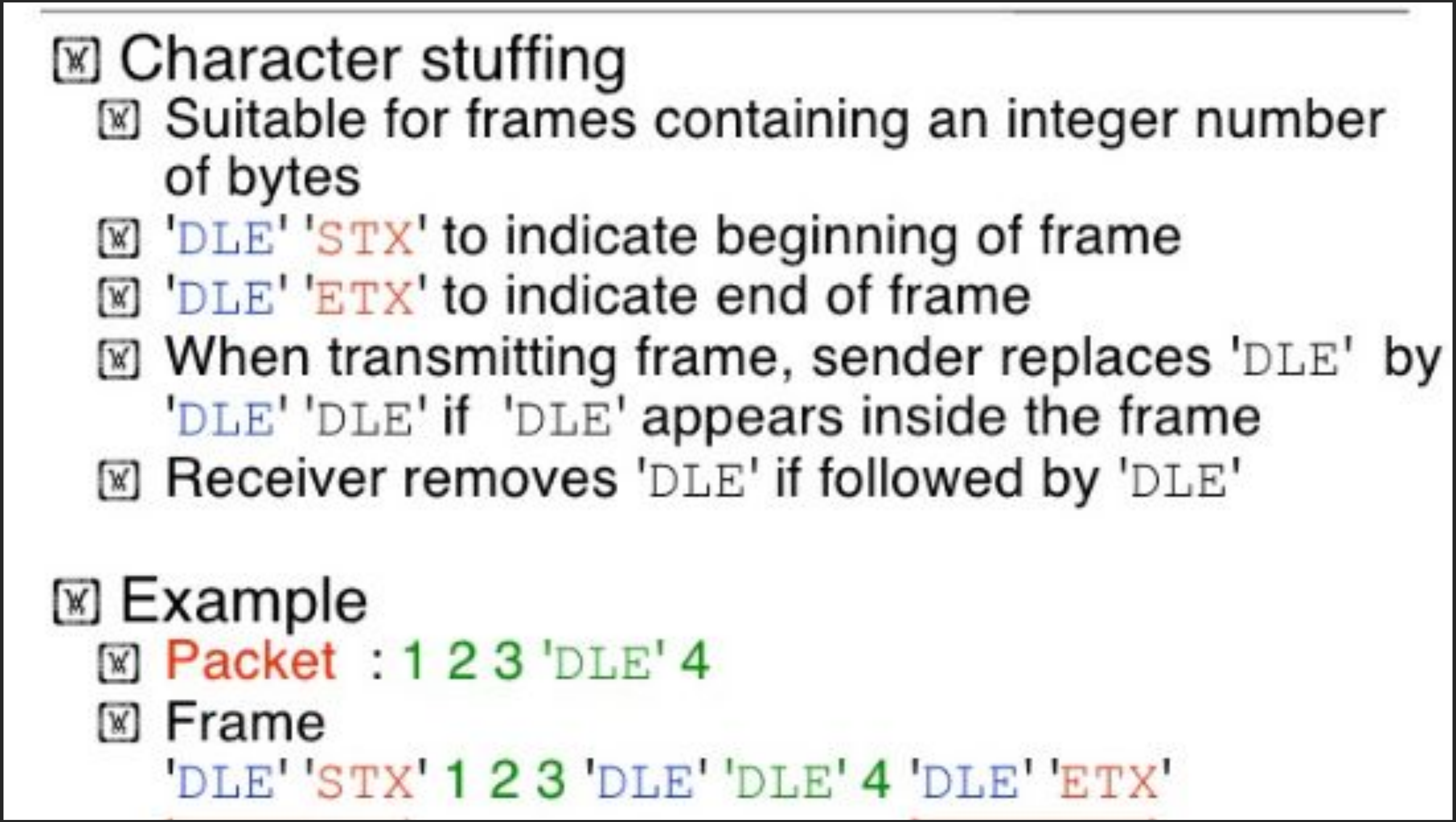
# Byte stuffing:

- In this method, start and end of frame are recognized with the help of flag bytes. Each frames starts with and ends with a flag byte.
- Two consecutive flag bytes indicate the end of one frame and start of the next one.
- The flag bytes used in the figure  used is named as “ESC” flag byte.
- Four example of byte sequences before and after stuffing:


FLAG
Header
Payload field
Trailer
FLAG
(a)
Original characters
After stuffing
A
FLAG
B
A
ESC
FLAG
B
A
ESC
B
A
ESC
ESC
B

# Character Count

- Each frame starts with the ASCII character sequence DLE STX and ends with the sequence DLE ETX.(where DLE is Data Link Escape, STX is Start of Text and ETX is End of Text.)



- Character stuffing
  - Suitable for frames containing an integer number of bytes
  - 'DLE' 'STX' to indicate beginning of frame
  - 'DLE' 'ETX' to indicate end of frame
  - When transmitting frame, sender replaces 'DLE' by 'DLE' 'DLE' if 'DLE' appears inside the frame
  - Receiver removes 'DLE' if followed by 'DLE'

- Example
  - Packet : 1 2 3 'DLE' 4
  - Frame
    'DLE' 'STX' 1 2 3 'DLE' 'DLE' 4 'DLE' 'ETX'

# 3.3 FLOW CONTROL

- It is a set of procedures that tells the sender how much data it can transmit before the data overwhelms the receiver.
- The receiving device has limited speed and limited memory to store the data.
- Therefore, the receiving device must be able to inform the sending device to stop the transmission temporarily before the limits are reached.
- It requires a buffer, a block of memory for storing the information until they are processed.
- **Two methods have been developed to control the flow of data:**
- Stop-and-wait
- Sliding window

# SIMPLEST

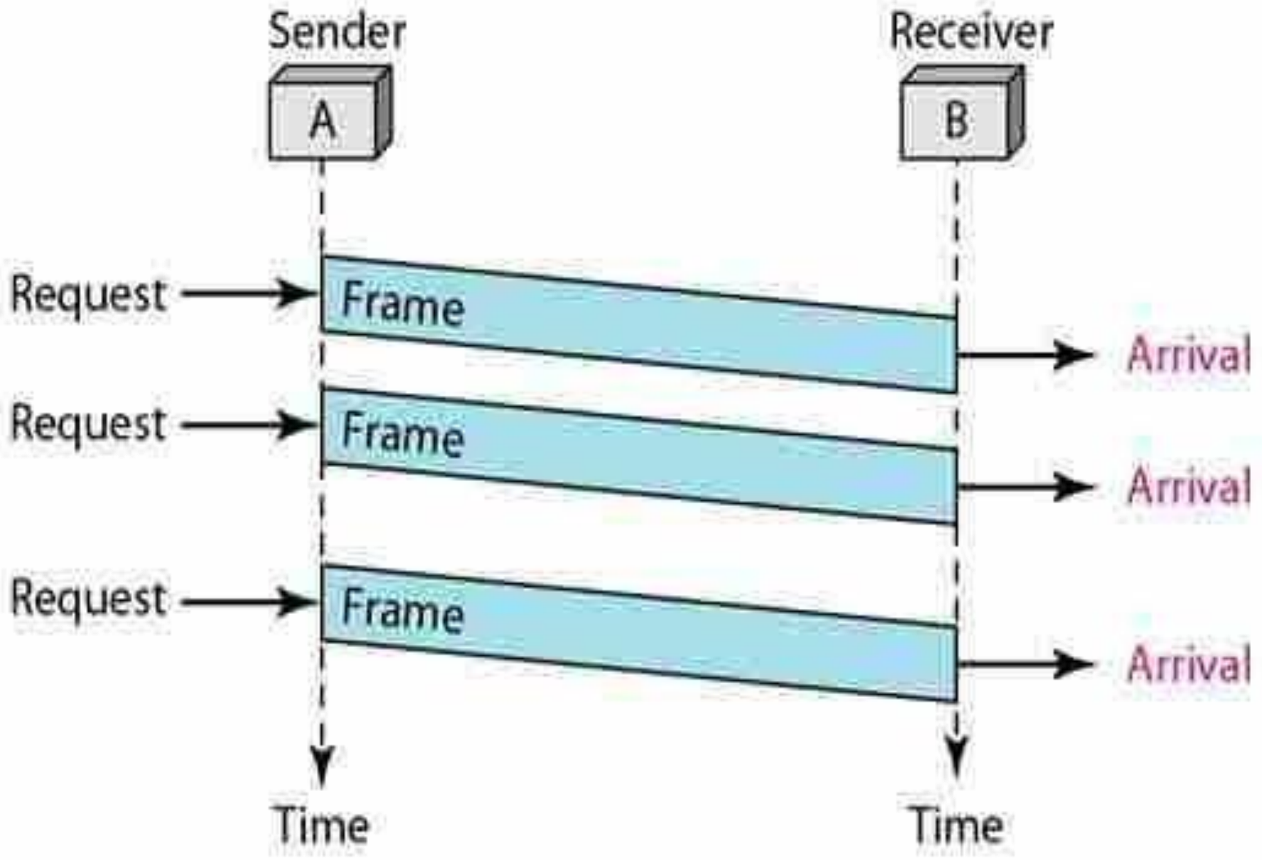
Simplest Protocol is one that has no flow or error control and it is a unidirectional protocol in which data frames are traveling in only one direction-from the sender to receiver.

We assume that the receiver can immediately 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.

The following figure shows an example of communication using this protocol.

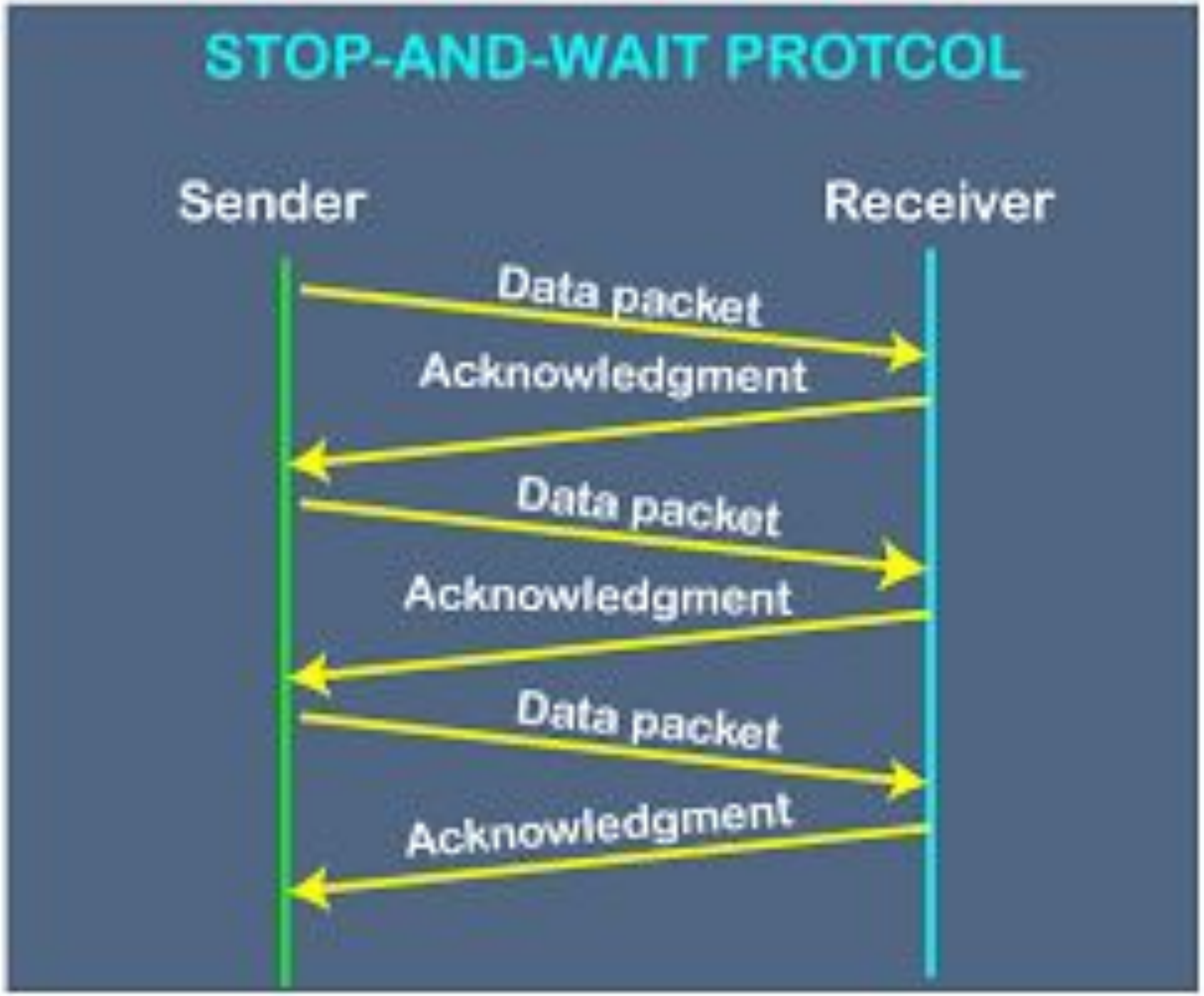
It is very simple. The sender sends a sequence of frames without even thinking about the receiver.


Sender
Receiver
A
B
Request
Frame
Arrival
Request
Frame
Arrival
Request
Frame
Arrival
Time
Time

# STOP AND WAIT

The sender sends a frame and waits for a response from the receiver. When ACK(acknowledged) will arrive from the receiver side then send the next frame and so on.

1. The sender node sends a data packet to the receiver node.
2. Then, waits for the feedback of the transmitted packet.
3. As soon as the receiver node receives a data packet it starts processing it.
4. Then, the receiver node sends the feedback to the sender node (about the received data packet).
5. After receiving the feedback, if the feedback is positive then the sender node sends the next data packet otherwise resends the damaged packet.

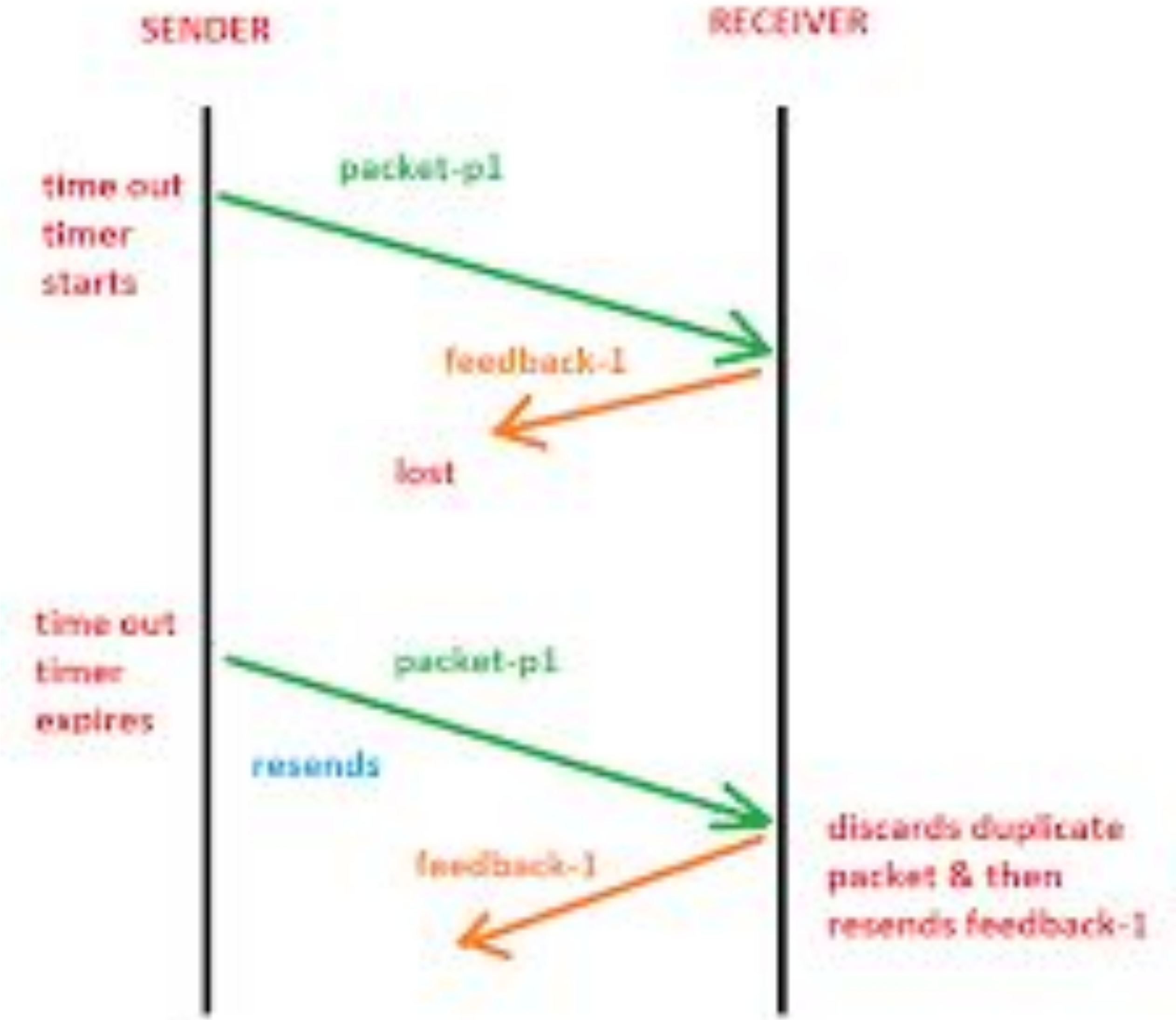

STOP-AND-WAIT PROTCOL
Sender
Receiver
Data packet
Acknowledgment
Data packet
Acknowledgment
Data packet
Acknowledgment

# STOP AND WAIT ARQ

Stop & Wait ARQ assumes that the communication channel is noisy (previously Stop & Wait assumed that the communication channel is not noisy). Stop & Wait ARQ also assumes that errors may occur in the data while transmission.

The following transition may occur in Stop-and-Wait ARQ:

- The sender maintains a timeout counter.
- When a frame is sent, the sender starts the timeout counter.
- If acknowledgement of a frame comes in time, the sender transmits the next frame in the queue.
- If acknowledgement does not come in time, the sender assumes that either the frame or its acknowledgement is lost in transit. Sender retransmits the frame and starts the timeout counter.
- If a negative acknowledgement is received, the sender retransmits the frame.


SENDER
RECEIVER
time out timer starts
packet-p1
feedback-1
lost
time out timer expires
packet-p1
resends
discards duplicate packet & then resends feedback-1
feedback-1

# Sliding Window Flow Control :

- This method is required where reliable in-order delivery of packets or frames is very much needed like in data link layer.
- It is point to point protocol that assumes that none of the other entity tries to communicate until current data or frame transfer gets completed.
- In this method, sender transmits or sends various frames or packets before receiving any acknowledgment.
- In this method, both the sender and receiver agree upon total number of data frames after which acknowledgment is needed to be transmitted.
- Data Link Layer requires and uses this method that simply allows sender to have more than one unacknowledged packet “in-flight” at a time.
- This increases and improves network throughput.

# GO BACK N

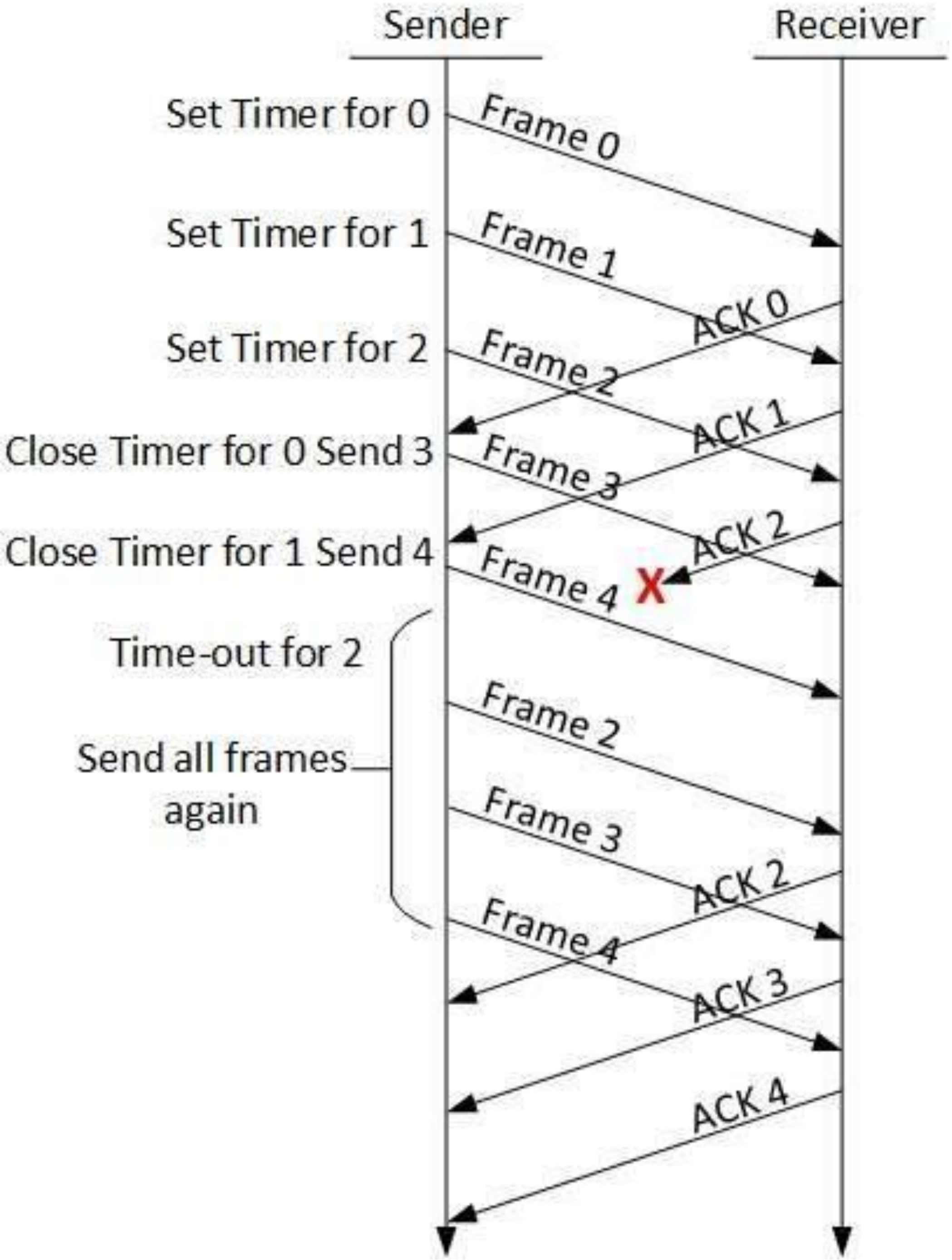
Stop and wait ARQ mechanism does not utilize the resources at their best.When the acknowledgement is received, the sender sits idle and does nothing. In Go-Back-N ARQ method, both sender and receiver maintain a window.

The sending-window size enables the sender to send multiple frames without receiving the acknowledgement of the previous ones.

The receiving-window enables the receiver to receive multiple frames and acknowledge them. The receiver keeps track of the incoming frame's sequence number.

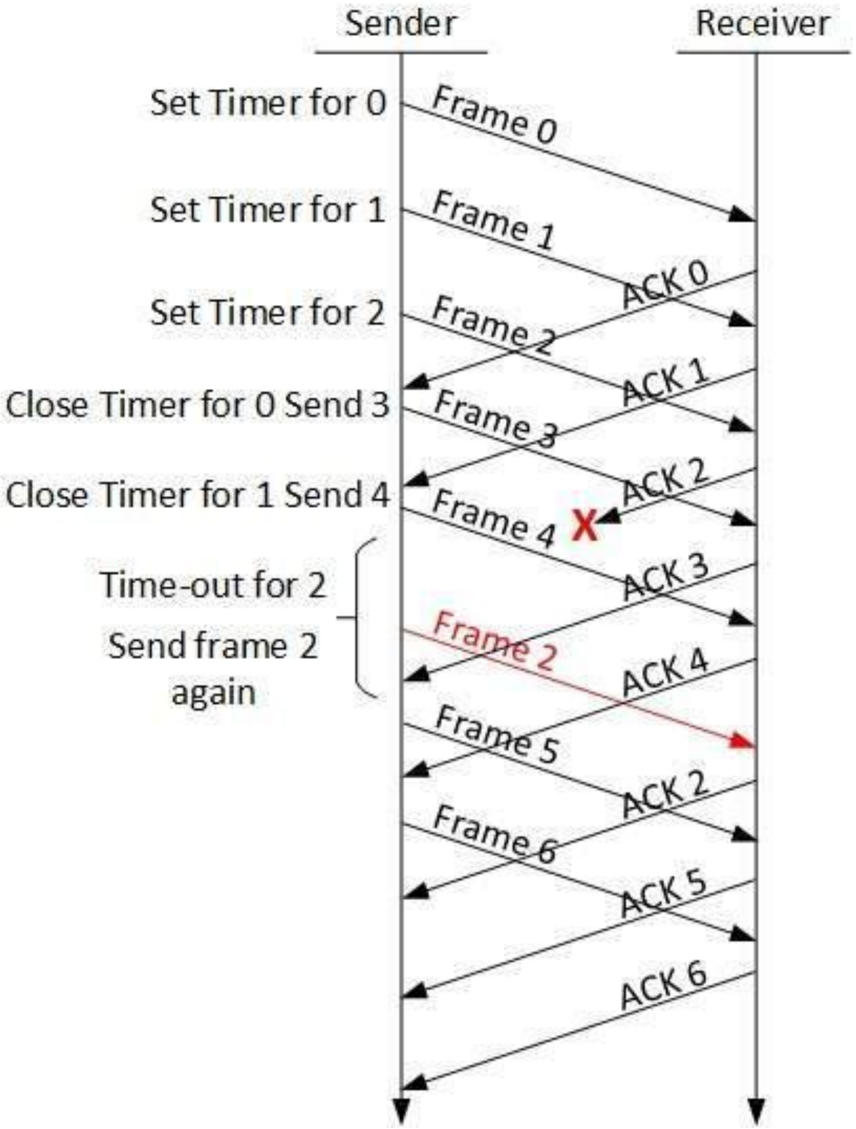
When the sender sends all the frames in the window, it checks up to what sequence number it has received positive acknowledgement.

If all frames are positively acknowledged, the sender sends the next set of frames. If the sender finds that it has received NACK or has not received any ACK for a particular frame, it retransmits all the frames after which it does not receive any positive ACK.


Sender
Receiver
Set Timer for 0
Frame 0
Set Timer for 1
Frame 1
ACK 0
Set Timer for 2
Frame 2
ACK 1
Close Timer for 0 Send 3
Frame 3
ACK 2
Close Timer for 1 Send 4
Frame 4
X
Time-out for 2
Frame 2
Send all frames again
Frame 3
ACK 2
Frame 4
ACK 3
ACK 4

# SELECTIVE REPEAT

In Go-back-N ARQ, it is assumed that the receiver does not have any buffer space for its window size and has to process each frame as it comes. This enforces the sender to retransmit all the frames which are not acknowledged. In Selective-Repeat ARQ, 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 in this case, sends only the packet for which NACK is received.


Sender
Receiver
Set Timer for 0
Frame 0
Set Timer for 1
Frame 1
ACK 0
Set Timer for 2
Frame 2
ACK 1
Close Timer for 0 Send 3
Frame 3
ACK 2
Close Timer for 1 Send 4
Frame 4
X
ACK 3
Time-out for 2
Send frame 2 again
Frame 2
ACK 4
Frame 5
ACK 2
Frame 6
ACK 5
ACK 6