UNIT 4- DATA LINK LAYER

- **CONTENTS**
- Introduction; Media Access Control (Contention, Controlled Access, Relative Performance); Error
- Control (Sources of Errors, Error Prevention, Error Detection, Error Correction via
- Retransmission, Forward Error Correction, Error Control in Practice); Data Link Protocols
- (Asynchronous Transmission, Synchronous Transmission); Transmission Efficiency.

•**Role of the Data-Link Layer**
•Transfers data across individual network links.
•Connects two directly linked devices, like neighboring computers or routers.

•**What is a Link?**
•Communication channels between two adjacent devices.
•Examples include Ethernet cables and Wi-Fi connections.

•**Common Data-Link Layer Protocols**
•**Ethernet**
•**Token Ring**
•**FDDI (Fiber Distributed Data Interface)**
•**PPP (Point-to-Point Protocol)**

•**Data Movement Across a Network**
•Data travels from the source to the destination through multiple links.
•The Data-Link Layer ensures smooth transfer over each individual link.
•**Key Takeaway**
•The Data-Link Layer acts as a bridge, ensuring data moves efficiently and accurately across each segment of
•a network path.

- **Data-Link Layer Functions**

1.Framing and Link Access:

1. **Framing:** Encapsulates each network-layer packet into a frame with a defined structure, including headers.
2. **Link Access:** Determines how frames are transmitted over the link using a channel access protocol.

2.Reliable Delivery:

1. Ensures each packet is delivered across the link without errors.
2. Uses **acknowledgments** and **retransmissions** to achieve this.

3.Flow Control:

1. Prevents the sender from overwhelming the receiver.
2. Manages the rate of data transmission to ensure smooth communication.

4.Error Detection:

1. Identifies errors in the transmitted frames.
2. The sender includes error-detection bits, and the receiver checks for errors upon receipt.

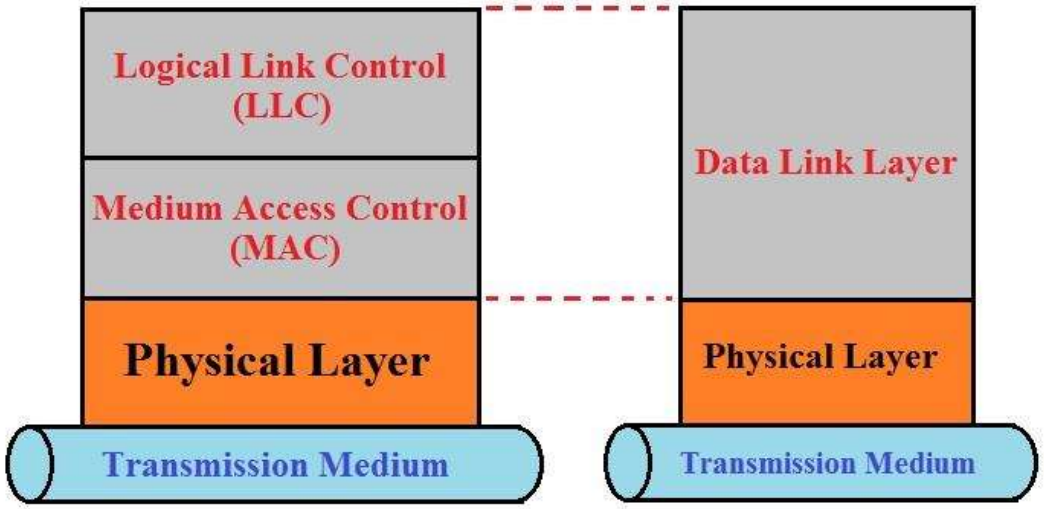
5.Error Correction:

1. Not only detects errors but also identifies and corrects them within the frame.

6.Half-Duplex and Full-Duplex:

1. **Full-Duplex:** Both nodes can send and receive data simultaneously.
2. **Half-Duplex:** A node can either send or receive data at one time, but not both.

- Overview of Logical Link Control (LLC) and Media Access Control (MAC)
- The data link layer is made up of two sublayers:
- LLC (Logical Link Control) Layer
- MAC(Media Access Control)Layer
- Both of these two sublayers are responsible for different functions for the data link layer.
- LLC interacts with the network layer above and the lower sub-layer, termed as MAC, that interacts
- with the physical layer below, as shown in the diagram given below:



- **LLC (Logical Link Control) Layer**
- LLC is responsible for handling multiple Layer3 protocols (multiplexing/demultiplexing) and link services like reliability and flow control,
- The functional overview of LLC are:
- This sublayer multiplexes protocols running a top the data link layer, and optionally provides flow control, acknowledgment, and error recovery.
- The LLC (Logical Link Control) handles the addressing and control of data at the data link layer. It determines how devices on a network are identified and manages the flow of data between the sender and receiver to ensure smooth communication.

- **MAC (Media Access Control) Layer**

1. Access Control:

1. Controls how devices access and share the transmission medium.

2. Data Packet Movement:

1. Moves data packets between Network Interface Cards (NICs) across a shared medium.

3. Physical Addressing:

1. Handles MAC addresses, ensuring devices can be uniquely identified on the network.

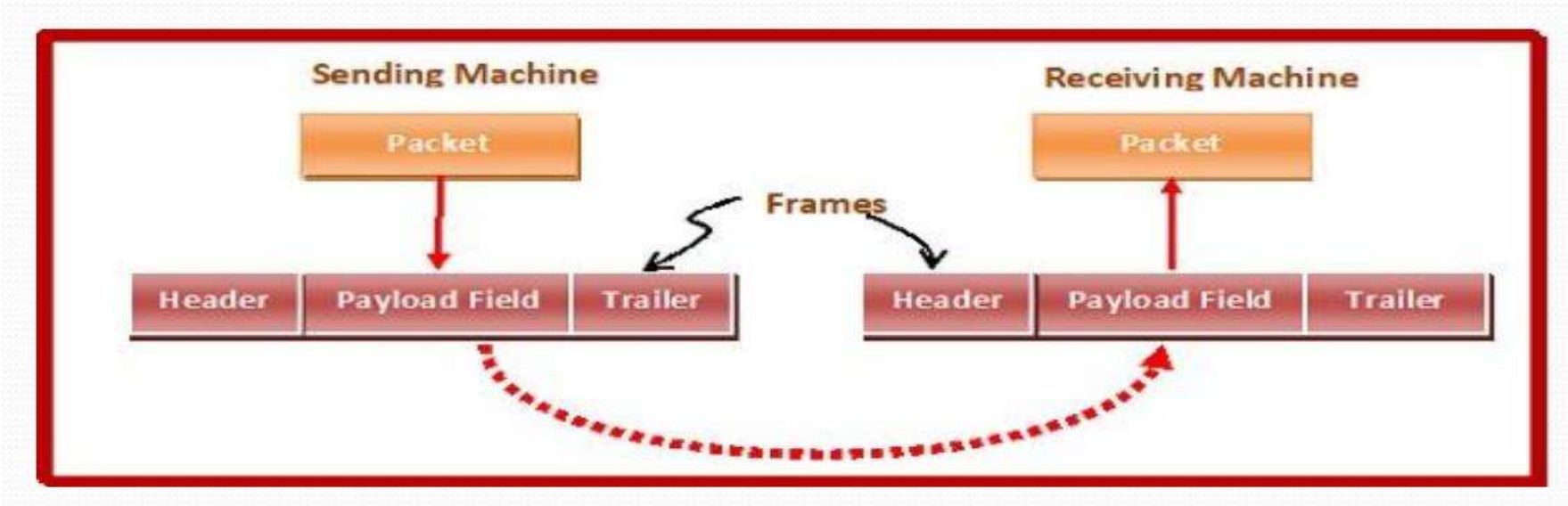
4. Collision Prevention:

1. Prevents multiple devices from transmitting simultaneously, avoiding data collisions.

5. Common MAC Methods:

1. **CSMA/CD (Carrier Sense Multiple Access/Collision Detection):** Used in Ethernet networks to detect and avoid collisions.
2. **CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance):** Used in networks like AppleTalk to avoid collisions before they happen.
3. **Token Passing:** Used in Token Ring and FDDI networks, where a token is passed around to control which device can send data.

- **Framing and Flow control Mechanism**
- Framing:
- In the physical layer, data transmission involves synchronized 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
- 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**

1. Fixed-Sized Framing:

1. Frames are all the same size.
2. The frame size itself is used to identify the start and end of each frame.
3. **Example:** ATM cells in conference call (used in Asynchronous Transfer Mode networks).

2. Variable-Sized Framing:

1. Frames can be different sizes.
2. Special markers or bits are used to indicate where one frame ends and the next begins.

Common Use: Local Area Networks (LANs).

- **Media Access Control**

1. What is Media Access Control?

1. It manages when computers can transmit data on a shared communication circuit.

2. When is it Needed?

1. **Not Needed:** In point-to-point full-duplex setups (only two devices can send/receive data anytime).
2. **Needed:** When multiple computers share the same communication circuit (like in half-duplex or multipoint setups). It prevents data collisions by ensuring only one device transmits at a time.

3. Fundamental Approaches:

1. **Contention:** Devices compete for the chance to transmit data. If two devices transmit at the same time, there's a collision, and the devices must try again.
2. **Controlled Access:** A system is in place to determine which device can transmit at a given time, preventing collisions.

- **Contention**

1. What is Contention?

1. Computers wait until the communication circuit is free (no other computers are transmitting) and then transmit their data.
2. Commonly used in Ethernet Local Area Networks (LANs).

2. Analogy:

1. Imagine a group conversation with friends. Everyone listens, and if no one is talking, you can speak.
2. If you want to say something, you wait for the current speaker to finish and then jump in.
3. Usually, the first person to speak right after the previous speaker is done gets to talk.
4. Occasionally, two people might start speaking at the same time, causing a "verbal collision," and then they need to sort out who talks next.

- **Controlled Access**

1. What is Controlled Access?

1. One device controls the communication circuit and decides which device can transmit data at a given time.

2. Common Techniques:

1. **Access Requests:** Devices that want to transmit send a request to the controlling device (like a wireless access point).
    1. The controlling device grants permission to one device at a time.
    2. Other devices wait until the current transmission is complete, then use a contention technique to request access.

3. Analogy:

1. Think of a classroom where the teacher controls who speaks.
2. Students raise their hands to request to speak.
3. The teacher (like the controlling device) decides who gets to speak next.
4. When the student finishes, the teacher takes control again and lets another student speak.

- **Polling**

1. What is Polling?

1. Polling is a process where the controlling device (like a wireless access point) sends a signal to each client computer, asking if it has data to send.
2. If the client has data, it sends it. If not, the controller moves on to the next client.

2. Types of Polling:

1. Roll-Call Polling:

1. The controller goes through a list of clients one by one, polling each in turn.
2. The sequence can be modified to give certain clients higher priority, polling them more frequently.
3. Example: Client 1 is polled more often in a sequence like 1, 2, 3, 1, 4, 5, 1, 6, 7.

2. Hub Polling (Token Passing):

1. One device starts the poll and passes it to the next device on the network.
2. Each device passes the poll along until it returns to the first device, which starts the process over again.

- **Relative Performance: Controlled Access vs. Contention**

1.Key Factor: Throughput

1. The main concern is which approach allows the most user data to be transmitted effectively.

2.Contention: Best for Small Networks with Low Usage

1. Works well when there are few computers and low network traffic.
2. Each computer can transmit data whenever needed, with minimal waiting.
3. Low risk of collisions, making it efficient in small networks.

3.Controlled Access: Best for Large Networks with High Usage

1. Ideal for networks with many computers and high traffic.
2. Reduces collisions, which are costly in large networks due to wasted capacity and the need for retransmissions.
3. Although it may slow down response time slightly, it makes better use of the network's capacity.

4.Finding the Balance:

1. The crossover point between using contention and controlled access is typically around 20 computers.
2. For networks with fewer than 20 computers, contention may be more efficient.
3. For networks with more than 20 computers, controlled access is usually better.

- **Error Control in Networks**
- **1. Importance of Error Control:**
- **Accuracy:** Networks need to ensure that data sent from one device to another arrives intact and unaltered. This is crucial for maintaining data integrity and reliability, especially in applications where accuracy is paramount, such as text data transmission.
- **2. Sources of Error:**
- **Data Corruption:** During transmission, data can become corrupted due to various factors, including electrical interference, signal degradation, or even hardware issues.
- **Types of Errors:** Errors can include bit flips (where bits change from 0 to 1 or vice versa) and lost or duplicated frames.
- **3. Error Detection and Correction:**
- **Error Detection:** The process involves identifying that an error has occurred during transmission. Common techniques include checksums, cyclic redundancy checks (CRC), and parity bits.
- **Error Correction:** This involves not only detecting errors but also correcting them. Techniques such as Automatic Repeat reQuest (ARQ), Forward Error Correction (FEC), and error-correcting codes like Hamming code are employed.

- **4. Handling Errors:**
- **Data-Link Layer:** At this layer, if a frame is corrupted, many link-layer protocols discard the frame and rely on upper-layer protocols to handle retransmissions.
- **Multimedia Applications:** Some applications, especially those involving real-time data like audio and video, may implement mechanisms to correct errors in the corrupted frame, often using techniques like interpolation or error concealment.
- **5. Error Tolerance:**
- **Text vs. Multimedia:** Different types of data have varying tolerance for errors. Text requires high accuracy, whereas multimedia content can often tolerate a small level of error due to its inherent redundancy and the nature of human perception.

Sources of Errors and Prevention

	Source of Error	What causes it	How to prevent it
More important	Line Outages	Faulty equipment, Storms, Accidents (circuit fails)	
	White Noise (hiss) (Gaussian Noise)	Movement of electrons (thermal energy)	Increase signal strength (increase SNR)
	Impulse Noise (Spikes)	Sudden increases in electricity (e.g., lightning, power surges)	Shield or move the wires
	Cross-talk	Multiplexer guard bands are too small or wires too close together	Increase the guard bands, or move or shield the wires
	Echo	Poor connections (causing signal to be reflected back to the source)	Fix the connections, or tune equipment
	Attenuation	Gradual decrease in signal over distance (weakening of a signal)	Use repeaters or amplifiers
mostly on analog	Intermodulation Noise	Signals from several circuits combine	Move or shield the wires
	Jitter	Analog signals change (small changes in amp., freq., and phase)	Tune equipment
	Harmonic Distortion	Amplifier changes phase (does not correctly amplify its input signal)	Tune equipment

- **Types of Noise and Errors in Data Communication**

1.Line Outages:

1. **Cause**: Storms, equipment failure, or short circuits.
2. **Impact**: Catastrophic errors or incomplete transmission.

2.White Noise:

1. **Cause**: Thermal agitation of electrons.
2. **Impact**: Always present but usually harmless unless it overwhelms the signal.

3.Impulse Noise:

1. **Cause**: Voltage changes, lightning, or poor circuit connections.
2. **Impact**: Primary cause of errors, appearing as clicks or spikes.

4.Cross-talk:

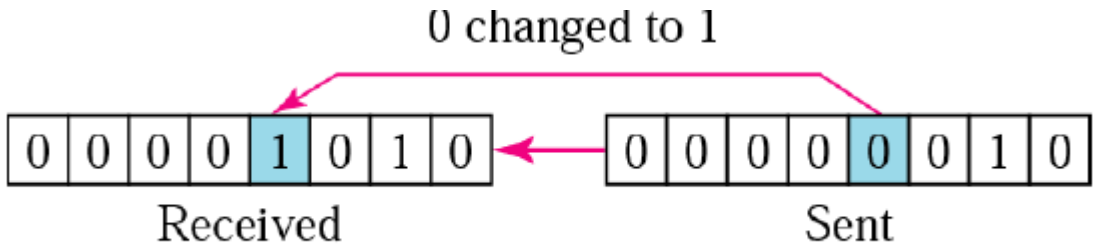
1. **Cause**: Signal leakage between circuits.
2. **Impact**: Unintended signals heard during transmission, like other conversations on a phone call.

5.Echoes:

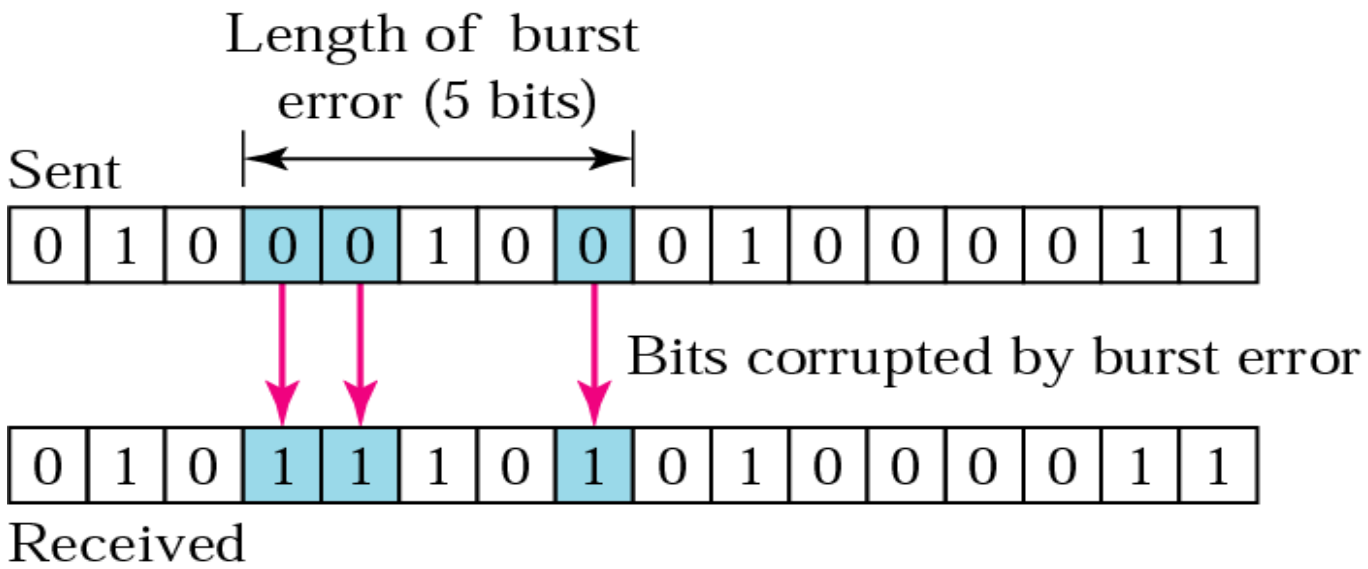
1. **Cause**: Signal reflection due to poor connections.
2. **Impact**: Can cause errors if strong enough.

1. **Attenuation**:
    1. **Cause**: Signal weakening over distance.
    2. **Impact**: Loss of signal strength, leading to interpretation errors.
2. **Intermodulation Noise**:
    1. **Cause**: Combination of signals from different circuits.
    2. **Impact**: Creates a new, unwanted signal.
3. **Jitter**:
    1. **Cause**: Variations in signal timing.
    2. **Impact**: Fluctuations in sound or data quality, like volume changes during a call.
4. **Harmonic Distortion**:
    1. **Cause**: Faulty amplifiers.
    2. **Impact**: Incorrect signal representation.
5. **Phase Hits**:
    1. **Cause**: Temporary phase shifts in the signal.
    2. **Impact**: Short-term errors that may correct themselves.

- **Types of Errors**
- **Single-Bit Error : In a single-bit error, only one bit in the data unit has changed.**



Burst Error: A burst error means that 2 or more bits in the data unit have changed.



- **Error Prevention**

1.Shielding:

1. **Purpose**: Prevents impulse noise, cross-talk, and intermodulation noise.
2. **Method**: Covers wires with insulating material. More shielding means better protection but higher cost.

2.Cable Management:

1. **Purpose**: Reduces noise by distancing cables from noise sources.
2. **Method**: Keep cables away from power lines, lights, and heavy machinery.

3.Multiplexing Adjustments:

1. **Purpose**: Reduces cross-talk and intermodulation noise.
2. **Method**: Use TDM instead of FDM, or adjust guardbands in FDM.

4.Equipment Maintenance:

1. **Purpose**: Reduces echoes and white noise.
2. **Method**: Regularly tune and check connections, especially in fiber-optic cables.

5.Amplifiers & Repeaters:

1. **Purpose**: Counteracts attenuation over long distances.
2. **Method**: Boosts signal strength at intervals, though noise is also amplified.

- **Error Detection**

Why Errors Occur:

1. **Causes**: Noise, cross-talk, and other interferences can corrupt data during transmission.
2. **Impact**: Applications may fail if they receive incorrect data.

1. Role of the Data-Link Layer:

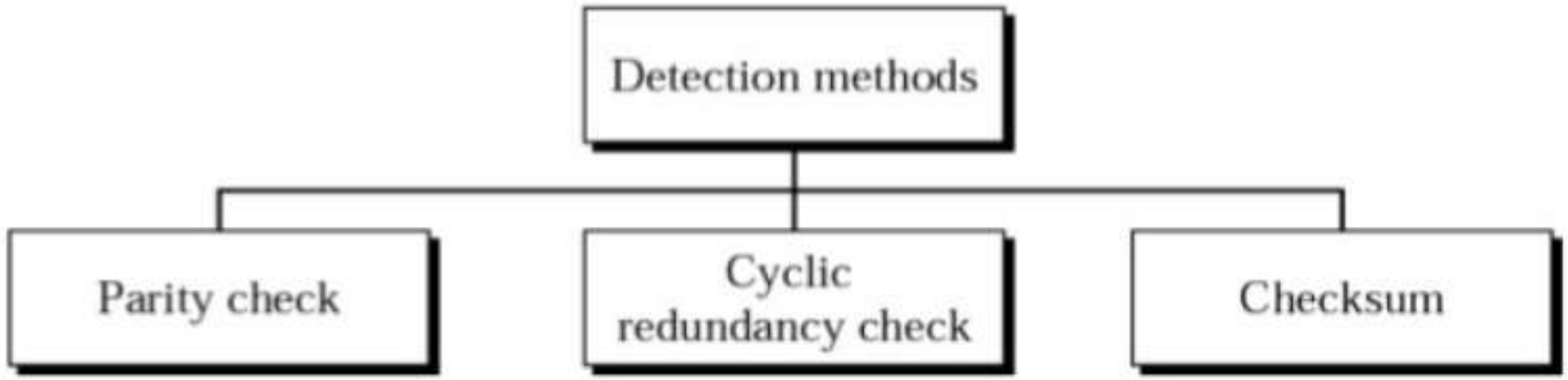
1. **Function**: Uses error control mechanisms to ensure data accuracy during transmission.

2. Error-Detecting Codes:

1. **Purpose**: Additional data is added to messages to detect any transmission errors.
2. **Method**: Extra bits, known as redundancy bits, are included to help identify errors.

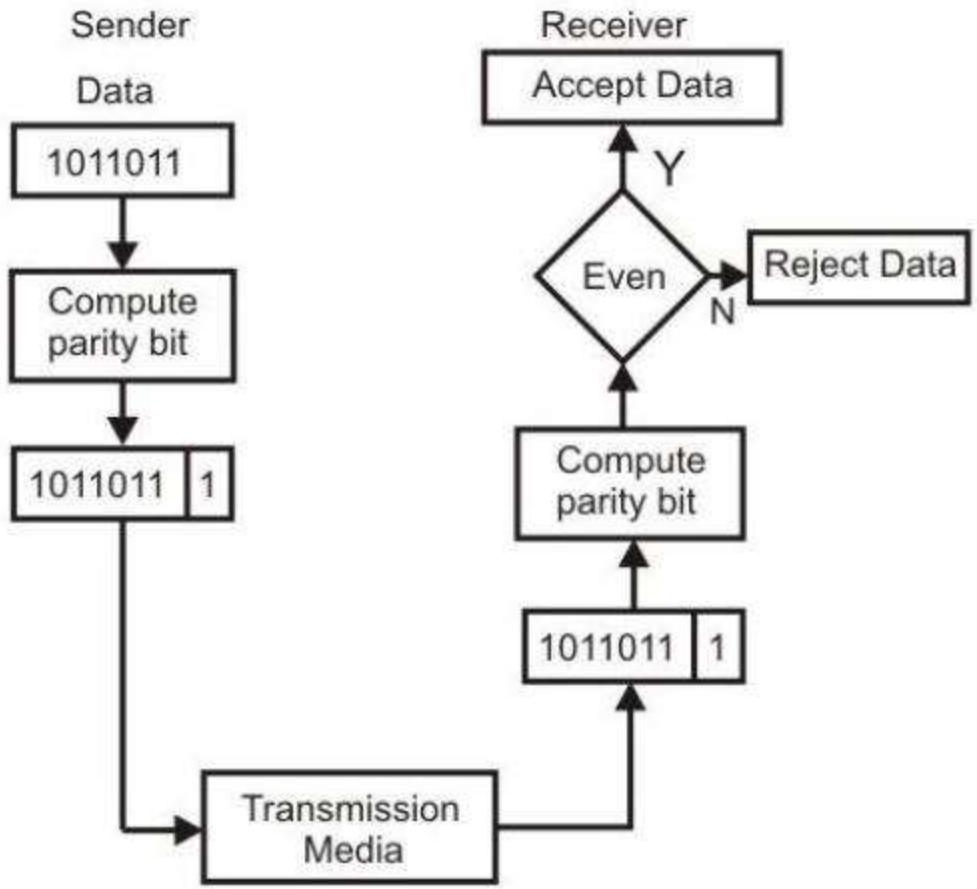
3. Redundancy Bits:

1. **Concept**: Extra bits added to the original data allow for error detection without altering the actual message content.


Detection methods
Parity check
Cyclic redundancy check
Checksum

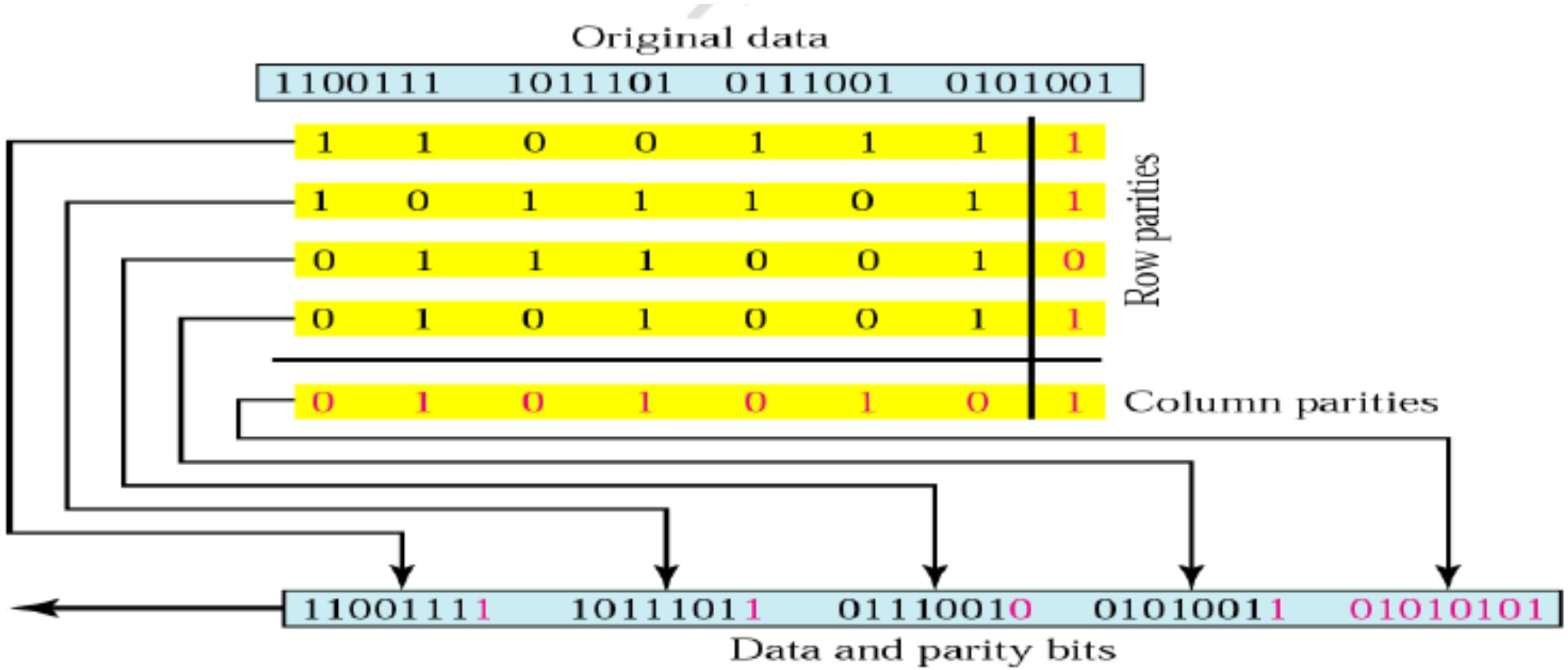
- **Parity Check**
- **Definition**: Parity check adds a parity bit to data to ensure the total number of 1s is either even (for even parity) or odd (for odd parity).
- **Simple Parity Check**
- **Process**:
  - **Sender Side**: A parity bit is added based on the data's 1s count.
    - If the data has an odd number of 1s, a parity bit of 1 is added for even parity (or 0 for odd parity).
  - **Receiver Side**: The received data's parity is checked to detect any errors.
- **Error Detection**:
  - **Single-bit errors**: All single-bit errors are detected.
  - **Burst errors**: Only detected if the total number of errors is odd.

- **Simple Parity Check Process**
- Sender Side:
- Input Data: The sender starts with a binary data string (e.g., 1011011).
- Compute Parity Bit: The sender counts the number of 1s in the data. In this case, there are five 1s (odd number). For even parity, 1 is added to make the total number of 1s even, resulting in 10110111.
- Transmission: The data with the parity bit (10110111) is sent through the transmission media



- Receiver Side:Compute Parity Bit: The receiver checks the received data (10110111) and computes the parity **Error Detection: If** the parity is even, the data is accepted.
- If the parity is not even (odd in this case), the data is rejected as it indicates an error occurred during transmission.

- **Longitudinal (2-D) Parity Check**
- **Concept**: In a two-dimensional (2-D) parity check, data is organized into a grid of rows and columns.
- **Process**:
  - **Row Parity**: Parity bits are calculated for each row of data, ensuring that each row has either an even or odd number of 1s.
  - **Column Parity**: Parity bits are then calculated for each column in the same way.
  - **Redundant Row**: A final row of parity bits (the redundant row) is added based on the parity of each column.
- **Error Detection**:
  - This method can detect and correct more complex errors than simple parity.
  - It can identify single-bit errors and detect if multiple bits within the block have been altered.



(Fig: Two-dimensional parity)

- **Checksum**
- It is based on the redundancy. Many computer network send a checksum along with each packet to help the receiver detect errors.
- To compute a checksum, the sender treats the data as a sequence of binary integers and form the sum of all units in the message which is called checksum.
- **Checksum Generator:**
- In the sender side the data unit is divided into equal segment of n bits.
- The segments are added.
- The total is then complemented and added to the end of the original data unit as redundancy bit called checksum bit.
- The extended data is transmitted across the network.

- **Checksum Checker:**
- In the receiver side the data unit is divided into k sections each of n bits.
- All sections are added to get the sum.
- The total is then complemented (1's complement).
- If the result is zero (0) the data is accepted otherwise they are rejected.

- Adding Binary
- 0 + 0 = 0
- 0 + 1 = 1
- 1 + 0 = 1
- 1 + 1 = 10 (which is 0 with a carry of 1 to the next higher bit)
- 1 + 1 + 1 = 11 (which is 1 with a carry of 1 to the next higher bit)

Example: Suppose the following block of 16 bits is to be sent using a checksum of 8 bits.

10101001 00111001

The numbers are added using one's complement

```
10101001
00111001
--------
```

Sum 11100010

Checksum **00011101**

The pattern sent is 10101001 00111001 **00011101**

Now suppose the receiver receives the pattern sent in Example and there is no error.

10101001 00111001 00011101

When the receiver adds the three sections, it will get all 1s, which, after complementing, is all 0s and shows that there is no error.

```
10101001
00111001
00011101
```

Sum 11111111

Complement **00000000** means that the pattern is OK.

Example:

Now suppose there is a burst error of length 5 that affects 4 bits.

10101**111 11**111001 00011101

When the receiver adds the three sections, it gets

	10101111	
	11111001	
	00011101	
Partial Sum	**1** 11000101	
Carry	**1**	
Sum	11000110	
Complement	**00111001**	**the pattern is corrupted.**

k=4, m=8: This implies that there are 4 data segments, each 8 bits long.

- The binary data segments are:
  - 10110011
  - 10101011
  - 10111001
  - 11011010
- The segments are added together using binary addition. Any carry beyond 8 bits is wrapped around and added to the sum.
- After adding all the segments, the sum is 10001111.
- The checksum is obtained by taking the complement (bitwise NOT) of the sum, which is 01110000.
- **Example (b): Checksum Verification**:
- The same data is received along with the checksum.
- The receiver adds the received data segments together, including the checksum.
- The sum obtained is 11111111.
- The complement of this sum is 00000000.
- Since the complement is all zeros, the conclusion is that the data is accepted, meaning no errors were detected in the transmission.

- **Cyclic Redundancy Check (CRC)**
- This Cyclic Redundancy Check is the most powerful and easy to implement technique.
- Unlike checksum scheme, which is based on addition, CRC is based on binary division i.e. modulo 2 division (X-OR)
- Can detect all odd errors, single bit error etc
- Burst error of length equal to polynomial degree.

- We will need to find CRC bit, i.e redundant bit
- We cant find the error in actual data so we will add redundant bit to the actual data.
- Formula: m+r
- M= number of bits in the message
- R= redundant bits
- We will need to append bits

- Lets say there is string 1010101010 and the generator polynomial is

X^4+X^3+1,

Now we have to append extra zero to the strings. i.e 1010101010000

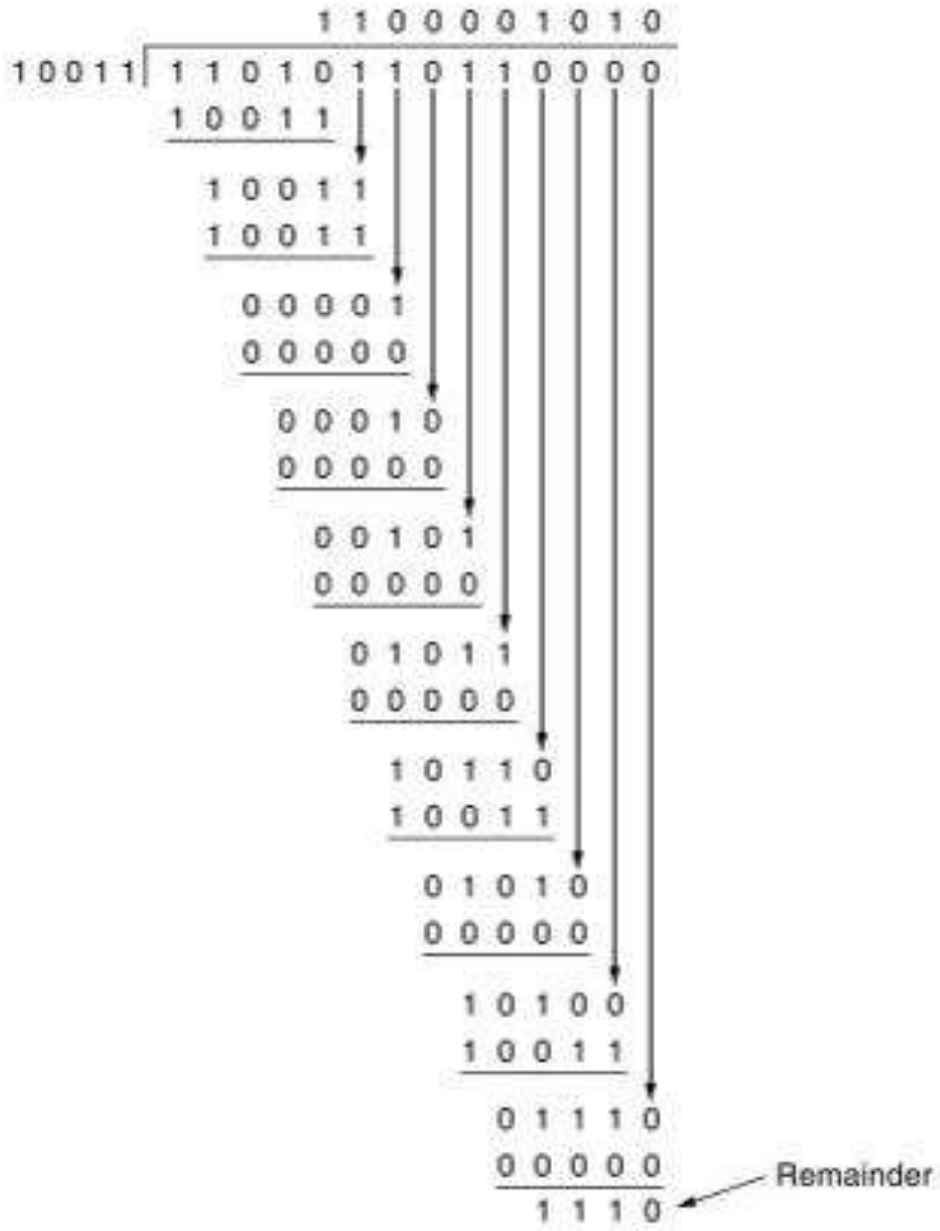
Converting polynomial to bits: 11001

If all are given in bits then (5-1), number of bits in divisor -1

Same number = 0

Different number = 1

- Problem 1
- A bit stream 1101011011 is transmitted using the standard CRC method. The generator polynomial is x^4+x+1. What is the actual bit string transmitted?
- Solution:
- The generator polynomial G(x) = x^4 + x + 1 is encoded as 10011.
- Clearly, the generator polynomial consists of 5 bits.
- So, a string of 4 zeroes (5-1) is appended to the bit stream to be transmitted.
- The resulting bit stream is 11010110110000.



From here, CRC = 1110.
Now,

- The code word to be transmitted is obtained by replacing the last 4 zeroes of 1101011011**0000 with the CRC.**
- Thus, the code word transmitted to the receiver = 1101011011**1110**

Error Correction

- Error correction codes are used to detect and correct the errors when data is transmitted from the sender to the receiver.
- Error correction is the additional ability to reconstruct the original, error-free data.
- There are two basic ways :
- Automatic Repeat Request(ARQ)
- Forward Error Correction(FEC)

- **Automatic Repeat Request (ARQ)**
- ARQ is an error-control method for data transmission that uses acknowledgments and timeouts to ensure reliable communication.
- **Process**:
  - **Data Transmission**: The sender transmits a data packet and waits for an acknowledgment (ACK) from the receiver.
  - **Acknowledgment**: If the receiver successfully receives the packet, it sends an ACK back to the sender.
  - **Timeout and Retransmission**: If the sender does not receive an ACK within a certain time (timeout), it assumes the packet was lost or corrupted and retransmits the packet.
  - **Negative Acknowledgment (NAK)**: Some ARQ systems also use NAKs, where the receiver sends a NAK if it detects an error, prompting immediate retransmission.

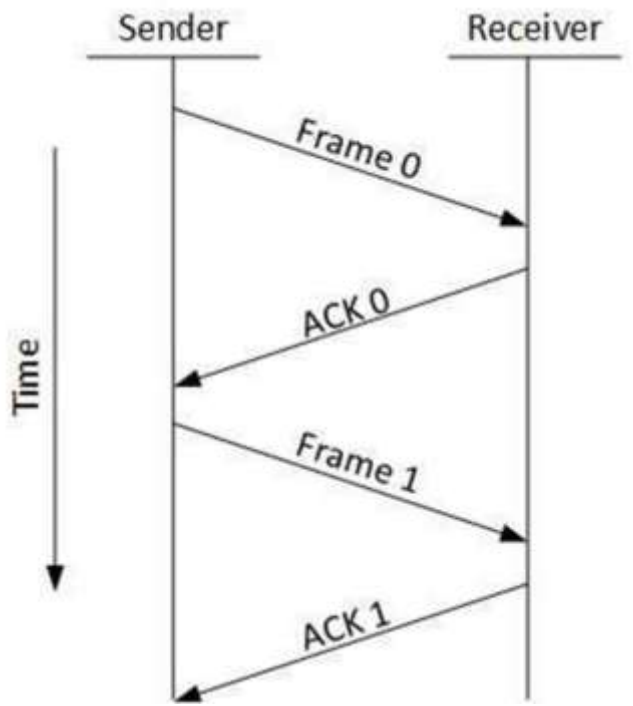
- **Forward Error Correction (FEC)**
- FEC is an error-control method that enables the receiver to detect and correct errors without needing retransmission.
- **Process**:
  - **Redundancy**: The sender adds redundant data (error-correcting codes) to the original data before transmission.
  - **Error Detection and Correction**: The receiver uses the redundant data to detect and correct any errors that occurred during transmission.
  - **No Retransmission**: Unlike ARQ, FEC does not require retransmission of data; the receiver corrects errors on its own.

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 memory to store data. Therefore receiving device must be able to inform the sending device to stop transmission temporarily before limits are reached.
- There are two methods
- Stop and wait
- Sliding window(continuous)

- Stop and wait flow control
- **Single Frame Transmission**: The sender transmits one data frame at a time and then waits for an acknowledgment (ACK) from the receiver before sending the next frame.
- Acknowledgment Required: The sender must receive an ACK from the receiver to confirm that the frame was received successfully. Without the ACK, the sender does not send the next frame.
- **Timeout and Retransmission**: If the ACK is not received within a certain period (timeout), the sender retransmits the frame, assuming it was lost or corrupted.
- **Simple but Inefficient**: While Stop and Wait is easy to implement and ensures reliable transmission, it can be inefficient, especially over long-distance or high-latency networks, as the sender remains idle while waiting for the ACK.

- **Advantages**
- Each frame is transmitted only after first frame is acknowledged.
- Data frame is not lost.
- **Disadvantages**
- Inefficient, only one frame can be in transmission at a time.
- he time spent for waiting acknowledgment between each frame add significant amount to total transmission time

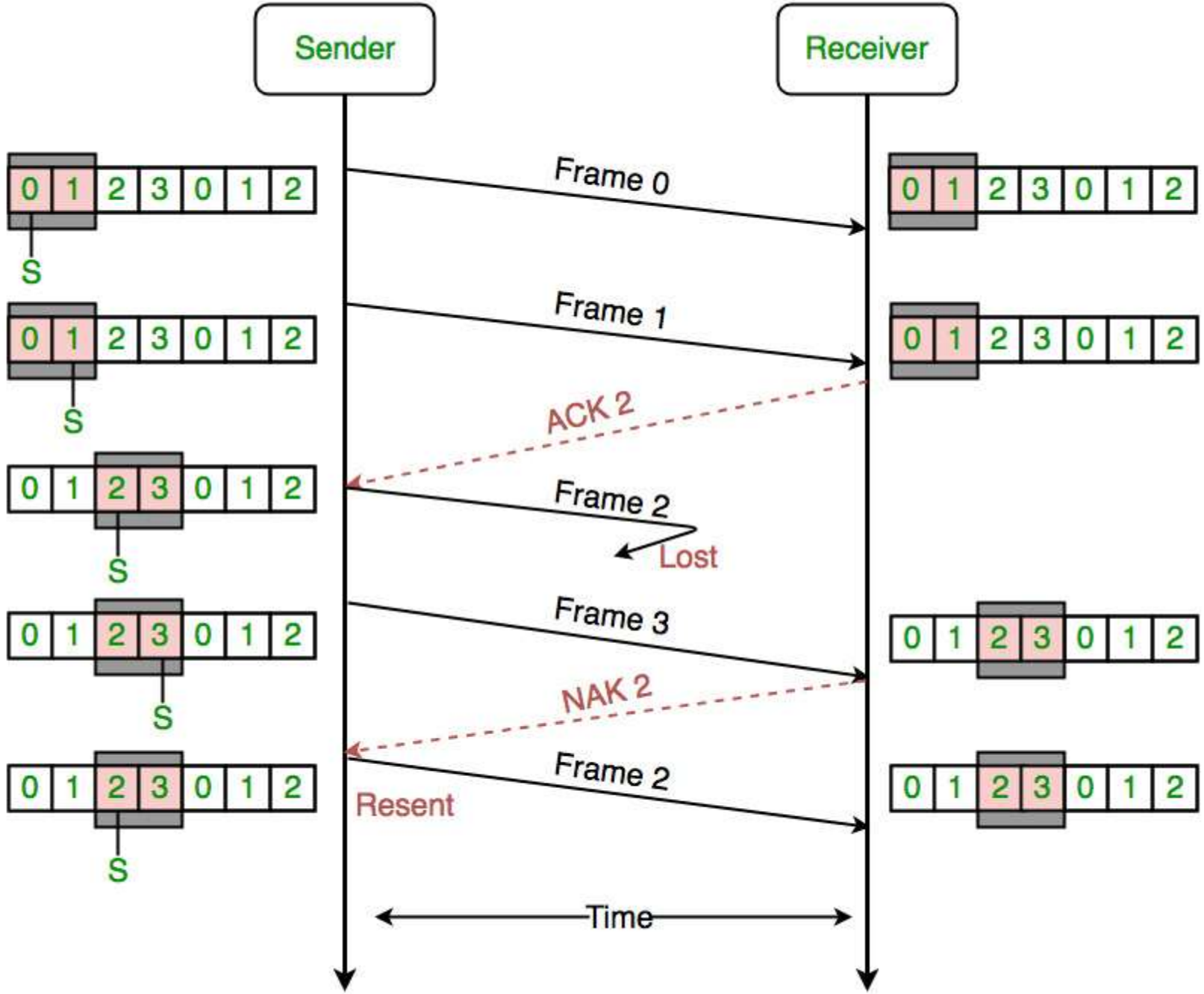


- The **Sliding Window Protocol** is a more advanced flow control mechanism that improves the efficiency of data transmission by allowing multiple frames to be sent before requiring an acknowledgment. It is used in various network protocols, including TCP.
- **Key Concepts:**

1. **Window Size**: The "window" represents the number of frames the sender can transmit without waiting for an acknowledgment. Both sender and receiver maintain a window of acceptable frames.
2. **Acknowledgments (ACKs)**: Unlike the Stop-and-Wait protocol, the Sliding Window Protocol allows the sender to send several frames before needing an ACK. The receiver can acknowledge multiple frames with a single ACK

- **Sliding Mechanism**:
- As frames are sent and acknowledged, the window "slides" forward, allowing more frames to be sent.
- The sender can send new frames within the window, and once the receiver acknowledges a frame, the window moves forward to allow additional frames to be sent.
- **How It Works:**

1. **Sender sends multiple frames within the window size**.
2. **Receiver processes the frames** and sends an acknowledgment (ACK) for one or more frames.
3. **Sender receives the ACK** and slides the window forward, allowing new frames to be sent.


Sender
Receiver
0 1 2 3 0 1 2
S
Frame 0
Frame 1
ACK 2
Frame 2
Lost
Frame 3
NAK 2
Frame 2
Resent
Time

- What is piggybacking?
- The link between receiver and transmitter is full duplex and usually both transmitter and receiver stations send data to each other. So, instead of sending separate acknowledgement packets, a portion (few bits) of the data frames can be used for acknowledgement. This phenomenon is known as piggybacking. It helps in better channel utilization

- Error Control:
- **Error Control** in data communication involves mechanisms to detect and correct errors during frame transmission. Errors include:
- **Lost Frame**: A frame fails to reach its destination, possibly due to severe noise.
- **Damaged Frame**: A frame arrives with some bits altered.
- **Key Techniques for Error Control**:
- **Error Detection**: Identifies errors in frames.
- **Positive Acknowledgment**: Confirms successful receipt of error-free frames.
- **Retransmission after Timeout**: Retransmits frames not acknowledged within a set time.
- **Negative Acknowledgment and Retransmission**: Retransmits frames that are acknowledged as erroneous.

- Collectively, these mechanisms are all referred to as automatic repeat request
- (ARQ); the effect of ARQ is to turn an unreliable data link into a reliable one.

Three versions of ARQ have been standardized:

- Stop-and-wait ARQ
- Go-back-N ARQ
- Selective-reject ARQ
- All of these forms are based on the use of the flow control techniques discussed.

- Stop and wait ARQ
- Is a technique used to retransmit the data in case of damaged or lost frames. Works on the principal that the sender will not transmit the next frame until it receives the acknowledgement of last frame.
- **How It Works:**

1. Alternating Frame Numbers:

1. Each frame is assigned a unique label or number, typically alternating between 0 and 1.
2. This means the first frame sent is labeled 0, the second is labeled 1, the third is labeled 0 again, and so on.

2. Acknowledgment Correspondence:

1. **ACK0**: Indicates that frame 0 was successfully received and that the receiver is ready to accept frame 1 next.
2. **ACK1**: Indicates that frame 1 was successfully received and that the receiver is ready to accept frame 0 next.

- **Example:**

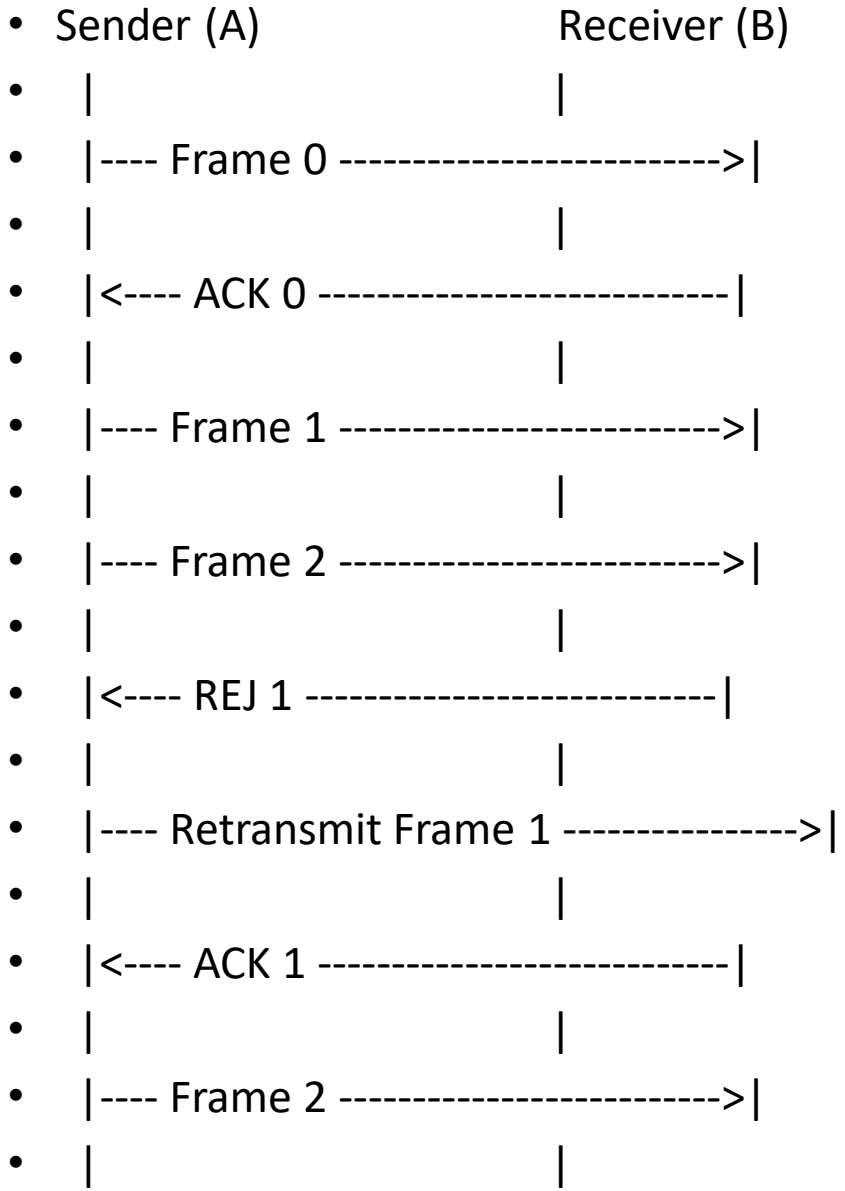
1. **Sender**: Sends frame 0.
2. **Receiver**: Receives frame 0 and sends ACK1.
3. **Sender**: Receives ACK1 and sends frame 1.
4. **Receiver**: Receives frame 1 and sends ACK0.
5. **Sender**: Receives ACK0 and sends frame 0.

- If a frame is lost or an acknowledgment is damaged:
- **Sender**: Resends the last frame (frame 0 or 1).
- **Receiver**: Identifies and discards any duplicate frames based on their labels.

- **Go-Back-N ARQ**
- **Overview:**
- **Improved Efficiency**: Unlike Stop-and-Wait, Go-Back-N ARQ allows the sender to transmit multiple frames before waiting for an acknowledgment (ACK).
- **Sliding Window**: Utilizes a sliding window technique to manage unacknowledged frames and improve line efficiency.
- **How It Works:**

1. **Frame Transmission**:
    1. **Sender**: Transmits multiple frames sequentially without waiting for individual ACKs.
    2. **Window Size**: Determines the number of frames that can be sent before receiving an ACK.
2. **Acknowledgments**:
    1. **Acknowledgment (ACK)**: Receiver sends ACKs for correctly received frames.
    2. **Negative Acknowledgment (REJ)**: If an error is detected in a frame, the receiver sends a REJ for that frame.

1. **Error Handling**:
    1. **Frame Rejection**: On receiving a REJ, the sender retransmits the erroneous frame and all subsequent frames.
    2. **Discarding Frames**: The receiver discards the erroneous frame and all frames received after it until the erroneous frame is correctly received.



Example Scenario:

- **Station A** (Sender) sends frames to **Station B** (Receiver).
- After sending frame i, Station A starts a timer.
- If frame i is successfully received, Station B sends an ACK for frame i.
- If an error is detected in frame i, Station B sends a REJ.
- Station A then retransmits frame i and all subsequent frames.

- **Error Handling in Go-Back-N ARQ**

1. **Damaged Frame**:
    1. Error occurs in frame 'i'.
    2. **Receiver Action**: Rejects frame 'i' and all subsequent frames.
    3. **Transmitter Action**: Retransmits from frame 'i'.
2. **Lost Frame**:
    1. Frame 'i' is lost.
    2. **Scenario 1**:
        1. Transmitter sends frame 'i+1'.
        2. **Receiver Action**: Detects out-of-sequence frame, rejects, and awaits frame 'i'.
    3. **Scenario 2**:
        1. Transmitter times out and sends an ACK request with the P bit set.
        2. **Receiver Action**: Responds with ACK for frame 'i'.
        3. **Transmitter Action**: Retransmits from frame 'i'.
3. **Damaged ACK**:
    1. Receiver gets frame 'i' but ACK for 'i+1' is lost.
    2. **Transmitter Action**: Times out and sends an ACK request with the P bit set.
    3. **Receiver Action**: Responds with ACK for frame 'i+1'.
    4. This can repeat until the issue is resolved, or a reset is initiated.
4. **Damaged REJ**:
    1. REJ for a damaged frame is lost.
    2. **Handled Like Lost Frame**: Transmitter times out and retransmits the affected frames.

- **Selective Reject ARQ (Selective Retransmission)**
- **Overview:**
- Only the frames that receive a negative acknowledgment (SREJ) or time out are retransmitted.
- More efficient than Go-Back-N ARQ as it reduces unnecessary retransmissions.
- **Advantages:**
- **Efficiency**: Minimizes retransmissions, making it more efficient than Go-Back-N ARQ.
- **Disadvantages:**
- **Complexity**: Requires both sender and receiver to manage more complex logic.
- **Buffer Requirement**: Receiver needs a large buffer to store out-of-sequence frames until the correct one is received.
- **Use Case:**
- Ideal for satellite links due to long propagation delays.
- This summary highlights the key points of Selective Reject ARQ for your presentation.

- **Selective Reject ARQ (Selective Retransmission)**
- **Overview:**
- Only the frames that receive a negative acknowledgment (SREJ) or time out are retransmitted.
- More efficient than Go-Back-N ARQ as it reduces unnecessary retransmissions.
- **Advantages:**
- **Efficiency**: Minimizes retransmissions, making it more efficient than Go-Back-N ARQ.
- **Disadvantages:**
- **Complexity**: Requires both sender and receiver to manage more complex logic.
- **Buffer Requirement**: Receiver needs a large buffer to store out-of-sequence frames until the correct one is received.
- **Use Case:**
- Ideal for satellite links due to long propagation delays.
- This summary highlights the key points of Selective Reject ARQ for your presentation.

- Data Link Protocols
- In this section, we outline several commonly used data link layer protocols, which are summarized in Figure. Here we focus on message delineation, which indicates where a message starts and stops, and the various parts or fields within the message. For example, you must clearly indicate which part of a message or packet of data is the error-control portion; otherwise, the receiver cannot use it properly to determine if an error has occurred. The data link layer performs this function by adding a PDU to the packet it receives from the network layer. This PDU is called a frame.

Protocol	Size	Error Detection	Retransmission	Media Access
Asynchronous transmission	1	Parity	Continuous ARQ	Full Duplex
Synchronous protocols				
SDLC	*	16-bit CRC	Continuous ARQ	Controlled Access
HDLC	*	16-bit CRC	Continuous ARQ	Controlled Access
Ethernet	*	32-bit CRC	Stop-and-wait ARQ	Contention
PPP	*	16-bit CRC	Continuous ARQ	Full Duplex

*Varies depending on the message length.

ARQ = Automatic Repeat reQuest; CRC = cyclical redundancy check; HDLC = high-level data link control; PPP = Point-to-Point Protocol; SDLC = synchronous data link control.

Asynchronous Transmission vs. Synchronous Transmission

Feature	Asynchronous Transmission	Synchronous Transmission
Timing	No timing; data sent independently of a clock	Uses a clock signal to sync sender and receiver
Data Grouping	Data sent in small groups (bytes), each with start/stop bits	Data sent in large blocks without start/stop bits
Overhead	Higher due to start/stop bits with each byte	Lower since blocks are sent with fewer extra bits
Speed	Slower, due to added start/stop bits	Faster, with continuous data flow
Use Case	Simple, low-speed communication like keyboards	High-speed data transfer like network connections

- **Synchronous Data Link Control (**SDLC) Frame Structure

Flag:

- Each frame starts and ends with a special bit pattern 01111110, known as the flag.

Address Field:

- Identifies the destination.
- Usually 8 bits, but can be 16 bits (must be consistent across the network).

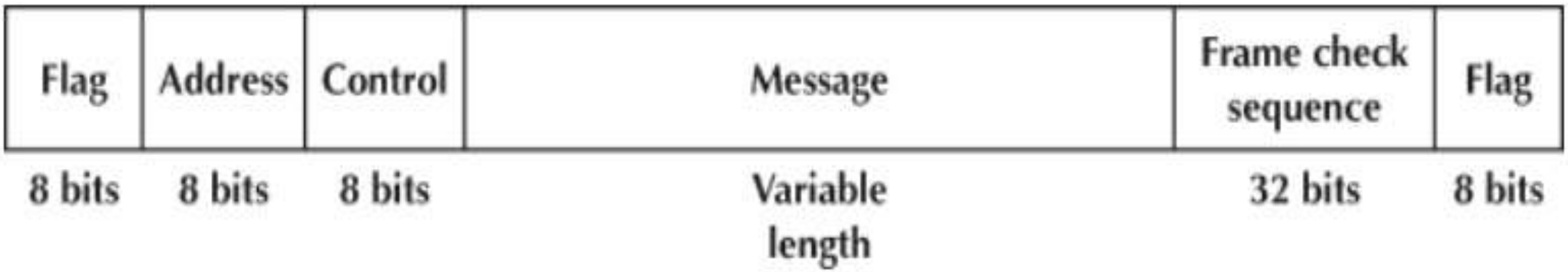
Control Field:

- Indicates the type of frame (information or supervisory).
- Information Frames: Used for message transfer and frame numbering.
- Supervisory Frames: Used for acknowledgments (ACKs/NAKs).

Message Field:

- Contains the user's message.
- Variable length

Frame Check Sequence (FCS):Provides error detection using a 32-bit CRC code (some versions use 16-bit CRC).

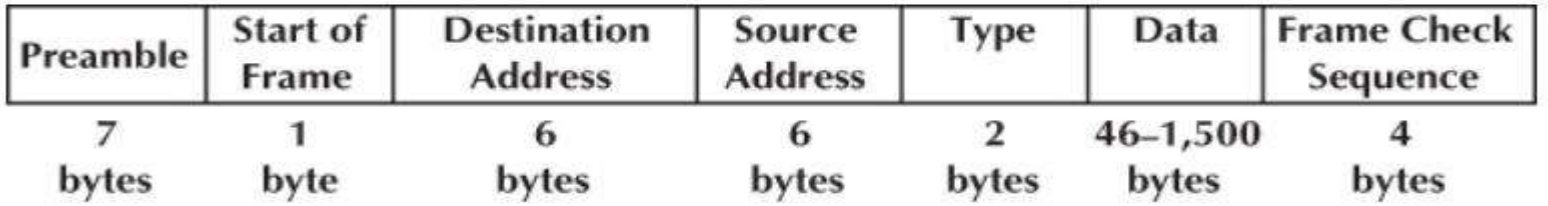

Flag
Address
Control
Message
Frame check sequence
Flag
8 bits
8 bits
8 bits
Variable length
32 bits
8 bits

- **High-Level Data Link Control (HDLC)**
- **Overview**:
  - A bit-oriented protocol for communication over point-to-point and multipoint links.
  - Developed by ISO as a standard for synchronous data link layer communication.
- **Key Features**:
  - **Synchronous Protocol**: Data is transmitted in a synchronized stream using clock signals.
  - **Bit-Oriented**: Similar to SDLC, it uses bit-level framing.
  - **Flexibility**: Supports both connection-oriented and connectionless services.
- **Comparison with SDLC**:
  - HDLC is based on SDLC but allows for longer address and control fields.

- **Ethernet Overview**
- **Introduction**:
  - Conceived by Bob Metcalfe in 1973 and developed by Digital, Intel, and Xerox.
  - Refined into the IEEE 802.3ac standard.
  - Uses a contention media access protocol for LAN communication.
- **Ethernet Frame Structure**:
  - **Preamble**: 7-byte pattern (10101010) to synchronize devices.
  - **Start Frame Delimiter**: Indicates the start of the frame.
  - **Destination & Source Addresses**: Specifies the receiver and sender.
  - **Length**: Indicates the size of the message.
  - **VLAN Tag (Optional)**: Used in virtual LANs.
  - **DSAP & SSAP**: Pass control information, like the type of network layer protocol.
  - **Control Field**: Contains frame sequence numbers and ACKs/NAKs for error control.
  - **CRC-32**: Error detection at the end of the frame.

Preamble	Start of Frame	Destination Address	Source Address	VLAN Tag	Length	DSAP	SSAP	Control	Data	Frame Check Sequence
7 bytes	1 byte	6 bytes	6 bytes	4 bytes	2 bytes	1 byte	1 byte	1–2 bytes	46–1,500 bytes	4 bytes

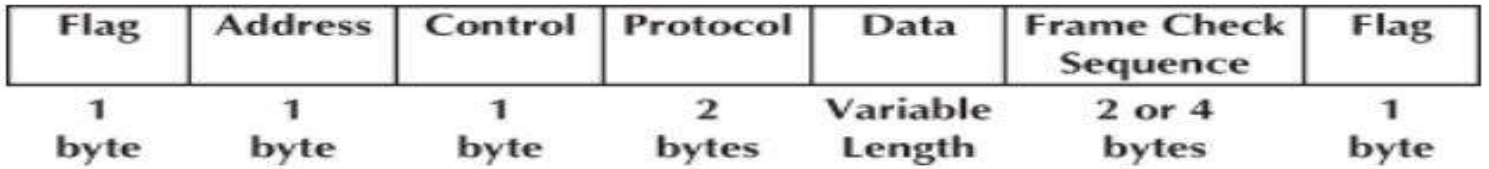
- **Ethernet II Overview**
- **Introduction**:



Preamble	Start of Frame	Destination Address	Source Address	Type	Data	Frame Check Sequence
7 bytes	1 byte	6 bytes	6 bytes	2 bytes	46–1,500 bytes	4 bytes

  - Ethernet II is another common version of Ethernet, similar to SDLC in using a preamble to mark the frame's start.
  - Shares the same source and destination address format as Ethernet 802.3ac.
- **Key Differences**:
  - **Type Field**: Specifies the type of network layer packet (e.g., IP) or an acknowledgment (ACK) frame.
  - **Data & Frame Check Sequence**: Same as in Ethernet 802.3ac.
- **Frame Sizes**:
  - **Jumbo Frames**: Support up to 9,000 bytes of user data, commonly used in gigabit Ethernet.
  - **Super Jumbo Frames**: Experimental frames that can hold up to 64,000 bytes, though not yet standardized.

- **Point-to-Point Protocol**



- **Introduction**:
  - PPP is widely used for point-to-point connections, such as connecting home computers to an ISP via a modem.
  - It is a common protocol for Internet access through telephone lines.
- **PPP Frame Structure**:
  - **Flag**: Marks the beginning of the frame.
  - **Address Field**: 1 byte, not used in point-to-point connections.
  - **Control Field**: Typically not used.
  - **Protocol Field**: Indicates the type of data packet (e.g., IP).
  - **Data Field**: Variable length, up to 1,500 bytes.
  - **Frame Check Sequence**: Usually CRC-16, but can be CRC-32.
  - **Flag**: Marks the end of the frame.

- **Transmission Efficiency**
- **Objective**: Maximize the volume of accurate information transmitted through a network to improve efficiency and reduce costs.
- **Factors Affecting Efficiency**:
- **Circuit Characteristics**: Error rates and maximum transmission speed.
- **Equipment Speed**: Transmitting and receiving equipment performance.
- **Error Detection and Control**: Methodology used.
- **Protocol Overhead**: Extra bits used by protocols for control and error checking.
- **Concepts**:
- **Information Bits**: Bits used to convey the actual message (user's data).
- **Overhead Bits**: Bits used for purposes like error checking and message delineation.

- **Calculation of Efficiency**:
- **Formula**: Transmission Efficiency=Information Bits / total Bits×100%
- **Example**: **Information Bits: 800 bits (the actual data you want to send).Total Bits: 1000 bits**
- Transmission efficiency= 800/1000 x 100%
- efficiency = 0.8x100% =80%
- The transmission efficiency in this example is **80%**. This means that 80% of the bits transmitted are actual information, while the remaining 20% are overhead (such as headers, error-checking bits, etc.).

- **Dial-up Modem Example**:
  - **Modem Speed**: 56 Kbps
  - **Effective Data Rate**: 39.2 Kbps (70% of 56 Kbps)
- **Improving Efficiency**:
  - **Reduce Overhead Bits**: For example, removing stop bits increases efficiency to approximately 77.8%.