Unit - 5

🌐 Data Link Layer – Introduction

- **Nodes** → Hosts & Routers (devices).
- **Links** → Channels connecting adjacent nodes (Wired, Wireless, LANs).
- At this layer, data is called a **Frame** (datagram + header & trailer).
- **Job** → Transfer a datagram from one node to its physically adjacent node.

⚡ Link Layer Services

1. **Framing**
    - Encapsulates datagram into a frame (adds header + trailer).
2. **Link Access**
    - Uses **MAC addresses** in frame headers to identify source & destination.
    - MAC ≠ IP (different addressing).
3. **Reliable Delivery**
    - Guarantees each datagram moves across the link without error.
    - Done using **Acknowledgment + Retransmission**.
4. **Flow Control**
    - Ensures proper pacing between sender & receiver.
5. **Error Detection & Correction**
    - Errors occur due to **signal attenuation & noise**.
    - Receiver detects/corrects errors or requests retransmission.

🧻 Framing Techniques

1. Bit Stuffing

- **Used in bit-oriented protocols.**
- Frames are marked with an **8-bit flag: 01111110** at beginning & end.
- Problem → Same flag pattern may appear in data part.
- Solution → After **five consecutive 1s**, sender inserts an extra **0**.
- Receiver removes the stuffed 0.

📌 **Example (from PPT):**
If data has sequence 01111110 (flag), sender changes it to 011111010. Receiver removes 0 before passing data.

---

2. Byte Stuffing

- Uses a **special flag byte** at start & end of frame.
- Problem → If flag byte appears inside data, confusion occurs.
- Solution → Sender inserts an **ESC (Escape) byte** before it.
- Receiver removes ESC before passing data up.

📌 **Example (from PPT):**
If data has accidental flag 01111110, sender sends ESC + flag. Receiver removes ESC and keeps flag as data.

🛠️ **Error Detection & Correction Techniques**

1. Parity Check

- Add **1 extra bit** → keeps total number of 1s **even (even parity)** or **odd (odd parity)**.
- Receiver counts 1s → If parity matches → data accepted.
- Detects **single-bit errors**, but not multiple-bit errors.

📌 **Example:**
Data: 1011 (3 ones = odd). With even parity → add 1 → 10111. Receiver checks → total 4 ones (even) → correct.

---

2. Checksum

- **Sender side:**
  - Data divided into k segments of m bits.
  - Segments added using **1's complement arithmetic**.
  - Complement of sum = checksum → sent with data.
- **Receiver side:**
  - Adds all segments + checksum.
  - If result = 0 → correct. Else → error.

📌 **Example (from PPT):**
Two 16-bit words are added.
If wraparound occurs, carry is added back.
If final result = 1111111111111111, data is correct.
If any 0 appears, error detected.

---

3. Cyclic Redundancy Check (CRC)

- **Most powerful** error detection. Based on **binary division**.
- **Sender side:**
    1. Append (k–1) zeros (where k = degree of polynomial + 1).
    2. Divide data by generator polynomial.
    3. Remainder is added to data → transmitted.
- **Receiver side:**
    - Divides received data by same polynomial.
    - If remainder = 0 → no error. Else → error.

📌 **Example (from PPT):**
Data = 100100, Polynomial = 1101 (x^3+x^2+1).
Sender appends 3 zeros → 100100000.
Remainder = 001.
Transmitted data = 100100001.
Receiver divides → remainder 000 → data correct.

📡 **Multiple Access Links**

- **Point-to-Point Link**
    - One sender ↔ One receiver.
    - Example: Direct cable connection between two computers.
- **Broadcast Link**
    - Multiple senders & receivers share one channel.
    - When one sends, **all receive a copy**.
    - Example: Classic Ethernet LAN → all devices connected via same bus cable.