Data Link Layer (Layer 2)

1 Position and Scope of the Data Link Layer

The Data Link Layer (DLL) is responsible for **node-to-node communication over a single physical link**. Its scope is strictly limited to adjacent nodes; it does **not** perform routing or end-to-end delivery.

Key distinction:

- Physical Layer: transmission of raw bits
- Data Link Layer: structured, controlled, and (optionally) reliable frame delivery

A common misconception is that DLL guarantees end-to-end reliability. This is **false**. Reliability, if provided, is limited to one hop.

2 Core Responsibilities (Exact and Canonical)

The Data Link Layer performs the following well-defined responsibilities:

1. Framing
2. Physical (MAC) Addressing
3. Error Detection and Correction
4. Flow Control
5. Medium Access Control (MAC)
6. Reliable Delivery (protocol-dependent)

Each responsibility is now analyzed technically.

3 Framing

3.1 Why Framing is Non-Trivial

The Physical Layer delivers a continuous bit stream:

$$0100110101011100101010111010...$$

There is no inherent information about:

- where a packet starts
- where it ends

Framing introduces explicit boundaries so the receiver can reconstruct packets.

3.2 Frame Structure (Ethernet Example)

An Ethernet frame has the following structure:

Preamble	Dest MAC	Src MAC	Payload	CRC
8 B	6 B	6 B	46–1500 B	4 B

Important detail: Payload smaller than 46 bytes is padded to maintain minimum frame size (64 bytes).

3.3 Framing Techniques

3.3.1 Character Count

The first field specifies the frame length.

Example:

$$[\text{Length} = 100][100 \text{ bytes of data}]$$

Pitfall: If the length field itself is corrupted, the receiver loses synchronization and misinterprets subsequent frames.

This is why character count is rarely used in modern networks.

3.3.2 Byte Stuffing

Special flag bytes mark frame boundaries.

Example:

```
FLAG | DATA ESC FLAG DATA | FLAG
```

Whenever FLAG appears in data, an ESC byte is inserted.

Limitation: Inefficient for binary data and depends on character encoding.

3.3.3 Bit Stuffing (HDLC, CAN)

Used in bit-oriented protocols.

Rule: After five consecutive 1s, the sender inserts a 0.

Example:

```
Original data:  1111101111110
After stuffing:1111100 11111010
```

The receiver removes the stuffed 0.

Why exactly 5 ones? Because the flag pattern is:

$$01111110$$

Stuffing prevents accidental appearance of this pattern inside data.

3.3.4 Physical Layer Coding Violations

Certain physical encodings have unused signal patterns. These illegal patterns are exploited to mark frame boundaries.

Example: Used in some Ethernet variants.

Pitfall: This technique is unavailable if all signal patterns are already used.

4 Physical (MAC) Addressing

4.1 MAC Address Format

A MAC address is a 48-bit identifier.

Example:

`08:00:27:AB:4C:91`

- First 24 bits: Organizationally Unique Identifier (OUI)
- Last 24 bits: Device-specific

4.2 Use in Frame Delivery

When a host sends a frame:

- Destination MAC = next-hop device
- Source MAC = sender

Important pitfall: MAC addresses change at every hop, unlike IP addresses.

5 Error Detection and Correction

5.1 Nature of Errors

Errors are primarily due to:

- Noise
- Attenuation
- Interference

Most common error type is a **burst error**, not a single-bit error.

5.2 Cyclic Redundancy Check (CRC)

CRC treats the frame as a polynomial over GF(2).

5.2.1 Sender Side

Let:

- Data polynomial = $D(x)$
- Generator polynomial = $G(x)$ (degree r)

The transmitted frame is:

$$T(x) = D(x) \cdot x^r + R(x)$$

where $R(x)$ is the remainder.

5.2.2 Receiver Side

The receiver computes:

$$T(x) \mod G(x)$$

- Zero remainder ⇒ accept frame
- Non-zero remainder ⇒ error detected

5.3 Detection Capability

CRC detects:

- All single-bit errors
- All double-bit errors (if $G(x)$ has at least 3 terms)
- All burst errors of length $\leq r$

Pitfall: CRC detects errors but does **not** correct them.

6 Flow Control

6.1 Why Flow Control Exists

Receiver buffer size is finite.

Example:

- Sender rate = 1 Gbps
- Receiver processing rate = 100 Mbps

Without flow control, buffer overflow causes frame loss.

6.2 Stop-and-Wait

Sender transmits one frame, then waits for ACK.

Link utilization:

$$U = \frac{T_{trans}}{T_{trans} + RTT}$$

Very inefficient for long-delay links.

6.3 Sliding Window

Allows multiple outstanding frames.

6.3.1 Go-Back-N (GBN)

- Window size = N
- Receiver only accepts in-order frames
- On error, retransmit from the erroneous frame onward

Pitfall: Wastes bandwidth due to unnecessary retransmissions.

6.3.2 Selective Repeat (SR)

- Receiver buffers out-of-order frames
- Only erroneous frames are retransmitted

Constraint: Window size $\leq \frac{1}{2}$ of sequence number space.

7 Medium Access Control (MAC)

7.1 Why MAC is Needed

If multiple nodes transmit simultaneously on a shared medium, collisions occur.

7.2 CSMA/CD (Ethernet)

- Sense channel before transmitting
- Detect collision during transmission
- Abort and backoff using binary exponential backoff

Important detail: Minimum frame size ensures collision detection before transmission completes.

7.3 CSMA/CA (Wi-Fi)

Collision detection is impossible in wireless due to:

- Hidden terminal problem
- Signal attenuation

Hence, collision avoidance is used.

8 Reliable Delivery

Reliable delivery at DLL is **optional**.

Examples:

- Ethernet: unreliable
- Wi-Fi: reliable (ACK-based)

Reliability uses:

- ACKs
- Timeouts
- Retransmissions

9 Common Exam Pitfalls

- Confusing MAC addressing with IP addressing
- Claiming DLL provides end-to-end reliability
- Ignoring minimum frame size in Ethernet
- Assuming CRC can correct errors
- Forgetting sequence number constraints in SR

10 Conclusion

The Data Link Layer is a technically rich layer responsible for framing, addressing, error handling, flow control, and medium access. Its mechanisms form the foundation for efficient and reliable communication in both wired and wireless networks.