# Data Link Layer

# Introduction

- The data link layer is the second layer from the bottom in the OSI (Open System Interconnection) network architecture model. It is responsible for the node-to-node delivery of data. Its major role is to ensure error-free transmission of information. DLL is also responsible for encoding, decoding, and organizing the outgoing and incoming data.
- This is considered the most complex layer of the OSI model as it hides all the underlying complexities of the hardware from the other above layers. In this article, we will discuss Data Link Layer in Detail along with its functions, and sub-layers.

# SubLayer of DLL

**Logical Link Control (LLC)**

- This sublayer of the data link layer deals with multiplexing, the flow of data among applications and other services, and LLC is responsible for providing error messages and acknowledgments as well.

**Media Access Control (MAC)**

- MAC sublayer manages the device's interaction, responsible for addressing frames, and also controls physical media access.
- The data link layer receives the information in the form of packets from the Network layer, it divides packets into frames and sends those frames bit-by-bit to the underlying physical layer.

# Functions of DLL

- Framing
- Physical Addressing
- Flow Control
- Error Control
- Access Control

## Framing

- The packet received from the Network layer is known as a frame in the Data link layer. At the sender's side, DLL receives packets from the Network layer and divides them into small frames, then, sends each frame bit-by-bit to the physical layer. It also attaches some special bits (for error control and addressing) at the header and end of the frame. At the receiver's end, DLL takes bits from the Physical layer organizes them into the frame, and sends them to the Network layer.

## Physical Addressing

- The data link layer encapsulates the source and destination's MAC address/ physical address in the header of each frame to ensure node-to-node delivery. MAC address is the unique hardware address that is assigned to the device while manufacturing.

## Error Control

- Data can get corrupted due to various reasons like noise, attenuation, etc. So, it is the responsibility of the data link layer, to detect the error in the transmitted data and correct it using error detection and correction techniques respectively. DLL adds error detection bits into the frame's header, so that receiver can check received data is correct or not. It adds reliability to phyiscal layer by adding mechansims to detect and retransmit damaged or lost frames.

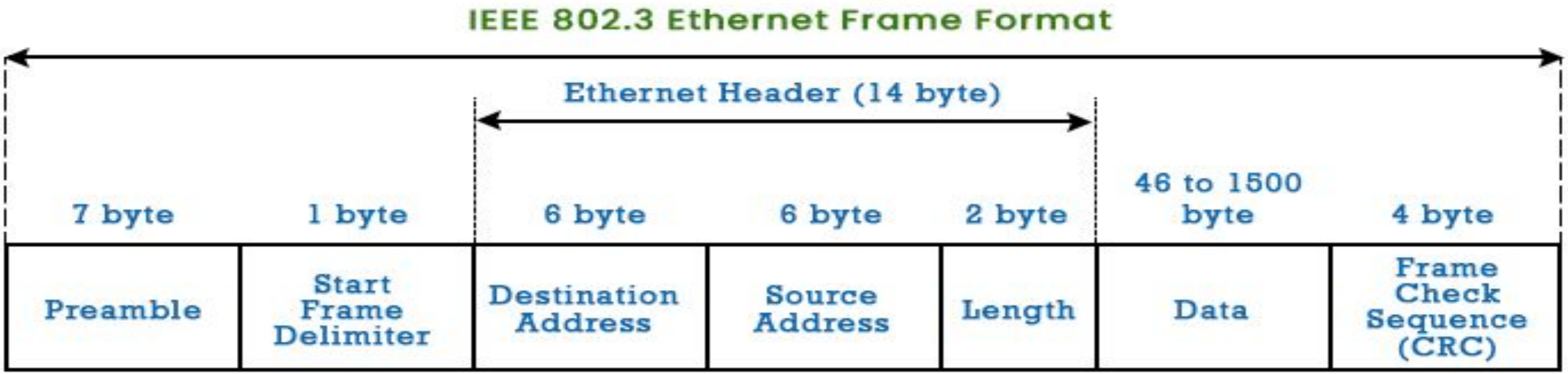
## Flow Control

- If the receiver's receiving speed is lower than the sender's sending speed, then this can lead to an overflow in the receiver's buffer and some frames may get lost. So, it's the responsibility of DLL to synchronize the sender's and receiver's speeds and establish flow control between them.

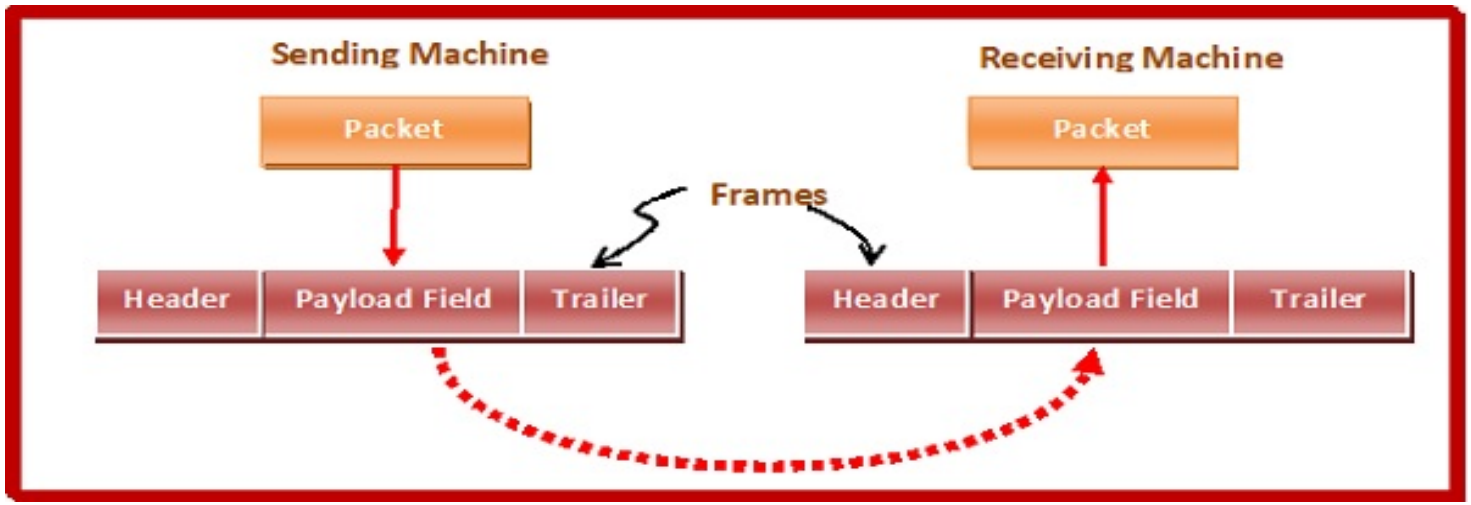
## Access Control

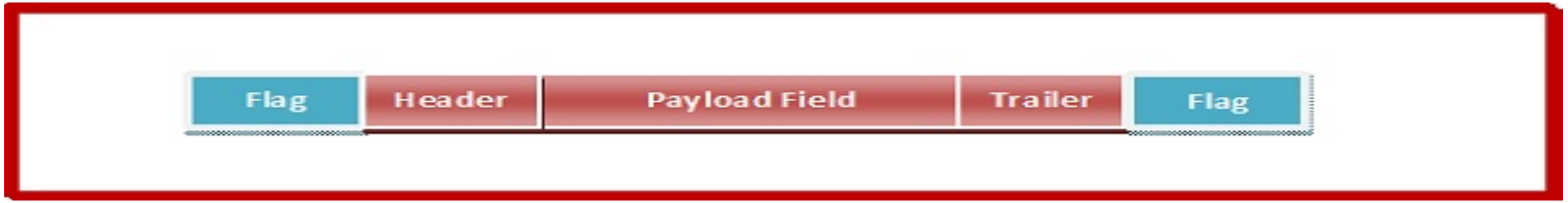
- When multiple devices share the same communication channel there is a high probability of collision, so it's the responsibility of DLL to check which device has control over the channel and CSMA/CD and CSMA/CA can be used to avoid collisions and loss of frames in the channel.

# Framing


IEEE 802.3 Ethernet Frame Format
Ethernet Header (14 byte)
7 byte
1 byte
6 byte
6 byte
2 byte
46 to 1500 byte
4 byte
Preamble
Start Frame Delimiter
Destination Address
Source Address
Length
Data
Frame Check Sequence (CRC)

- 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.



# Part of 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.

# Types of Framing

Framing can be of two types, fixed sized framing and variable sized framing.

- **Fixed-sized Framing**
- Here the size of the frame is fixed and so the frame length acts as delimiter of the frame. Consequently, it does not require additional boundary bits to identify the start and end of the frame.
- Example − ATM cells.

# Variable – Sized Framing

- Here, the size of each frame to be transmitted may be different. So additional mechanisms are kept to mark the end of one frame and the beginning of the next frame.
- It is used in local area networks.
- Two ways to define frame delimiters in variable sized framing are −
- Length Field − Here, a length field is used that determines the size of the frame.
- End Delimiter − Here, a pattern is used as a delimiter to determine the size of frame. It is used in Token Rings. If the pattern occurs in the message, then two approaches are used to avoid the situation −
  - Byte – Stuffing − A byte is stuffed in the message to differentiate from the delimiter. This is also called character-oriented framing.
  - Bit – Stuffing − A pattern of bits of arbitrary length is stuffed in the message to differentiate from the delimiter. This is also called bit – oriented framing.

# 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.

# Flow Control Mechanism

Stop & Wait ARQ

- Stop-and-wait ARQ is a technique used to retransmit the data in case of damaged or lost frames.
- This technique works on the principle that the sender will not transmit the next frame until it receives the acknowledgement of the last transmitted frame.

# Working Mechanism (Stop-and-Wait ARQ)

- The sending device keeps a copy of the last transmitted frame until the acknowledgement is received. Keeping the copy allows the sender to retransmit the data if the frame is not received correctly.
- Both the data frames and the ACK frames are numbered alternately 0 and 1 so that they can be identified individually. Suppose data 1 frame acknowledges the data 0 frame means that the data 0 frame has been arrived correctly and expects to receive data 1 frame.
- If an error occurs in the last transmitted frame, then the receiver sends the NAK frame which is not numbered. On receiving the NAK frame, sender retransmits the data.
- It works with the timer. If the acknowledgement is not received within the allotted time, then the sender assumes that the frame is lost during the transmission, so it will retransmit the frame.

# Go-Back-N ARQ

- Go-Back-N ARQ is form of ARQ protocol in which transmission process continues to send or transmit total number of frames that are specified by window size even without receiving an ACK (Acknowledgement) packet from the receiver. It uses sliding window flow control protocol. If no errors occur, then operation is identical to sliding window.

# Selective Repeat ARQ

- Selective Repeat ARQ is also form of ARQ protocol in which only suspected or damaged or lost data frames are only retransmitted. This technique is similar to Go-Back-N ARQ though much more efficient than the Go-Back-N ARQ technique due to reason that it reduces number of retransmission. In this, the sender only retransmits frames for which NAK is received. But this technique is used less because of more complexity between sender and receiver and each frame must be needed to be acknowledged individually.

# Piggybacking

- Piggybacking is the technique of delaying outgoing acknowledgment and attaching it to the next data packet.
- When a data frame arrives, the receiver waits and does not send the control frame (acknowledgment) back immediately. The receiver waits until its network layer moves to the next data packet. Acknowledgment is associated with this outgoing data frame. Thus the acknowledgment travels along with the next data frame. This technique in which the outgoing acknowledgment is delayed temporarily is called Piggybacking.