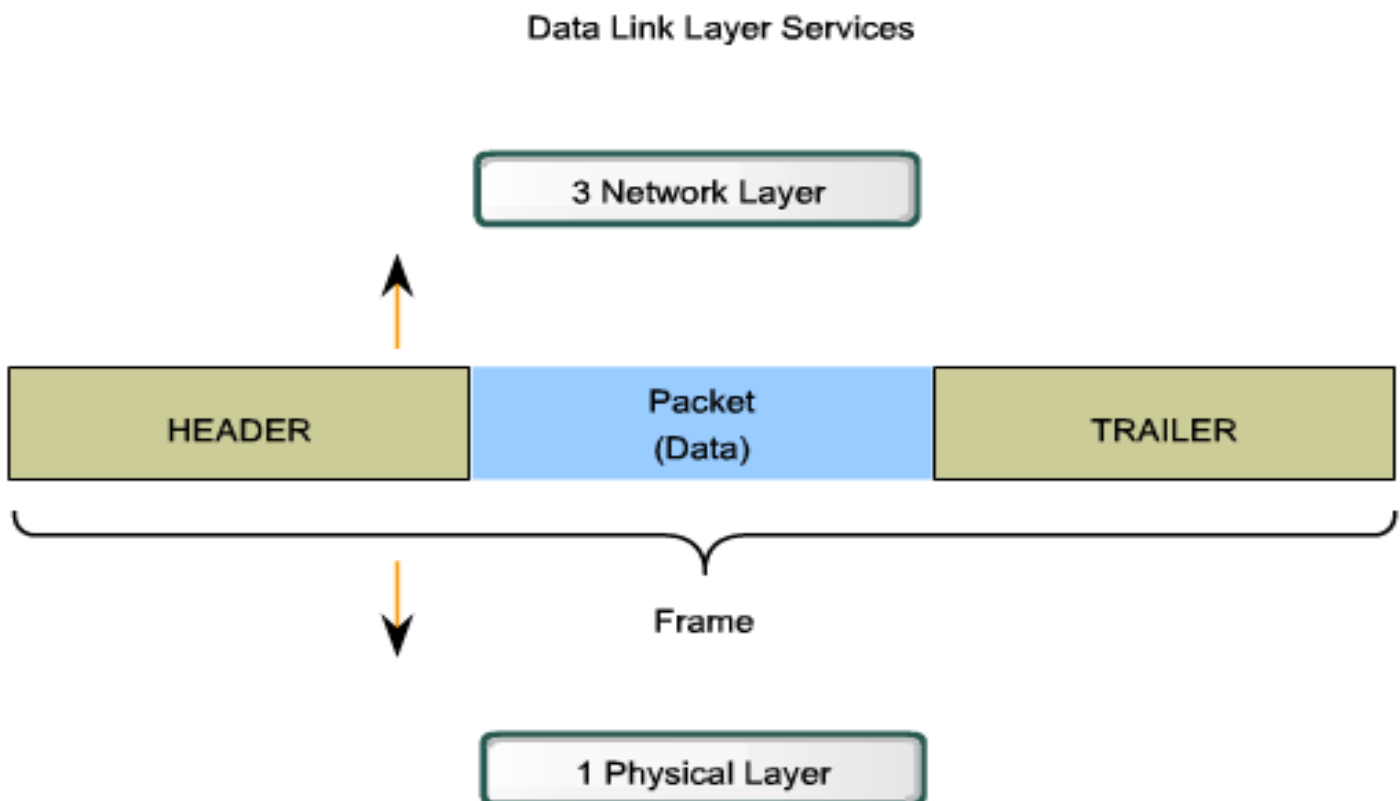# FRAMING

The data link layer, needs to pack bits into frames, so that each frame is distinguishable from another. The Data Link layer prepares a packet for transport across the local media by encapsulating it with a header and a trailer to create a frame.

The Data Link layer frame includes:
- Data - The packet from the Network layer
- Header - Contains control information, such as addressing, and is located at the beginning of the PDU
- Trailer - Contains control information added to the end of the PDU

Data Link Layer Services

3 Network Layer

| HEADER | Packet (Data) | TRAILER |

Frame

1 Physical Layer

Our postal system practices a type of framing. The simple act of inserting a letter into an envelope separates one piece of information from another; the envelope serves as the delimiter. In addition, each envelope defines the sender and receiver addresses since the postal system is a many-to-many carrier facility. Framing in the data link layer separates a message from one source to a destination, or from other messages to other destinations, by adding a sender address and a destination address. The destination address defines where the packet is to go; the sender address helps the recipient acknowledge the receipt.

Fixed-Size Framing

Frames can be of fixed or variable size. In fixed-size framing, there is no need for defining the boundaries of the frames; the size itself can be used as a delimiter. An example of this type of framing is the ATM wide-area network, which uses frames of fixed size called cells.

Variable-Size Framing

variable-size framing is prevalent in local- area networks. In variable-size framing, we need a way to define the end of the frame and the beginning of the next. Historically, two approaches were used for this purpose: **a character-oriented approach and a bit-oriented approach.**

Character-Oriented Protocols
In a character-oriented protocol, data to be carried are 8-bit characters from a coding system such as ASCII (see Appendix A). The header, which normally carries the source and destination addresses and other control information, and the trailer, which carries error detection or error correction redundant bits, are also multiples of 8 bits. To separate one frame from the next, an 8-bit (I-byte) flag is added at the beginning and the end of a frame. The flag, composed of protocol-dependent special characters, signals the start or end of a frame .

Any pattern used for the flag could also be part of the information. To fix this problem, **a byte-stuffing** strategy was added to character-oriented framing. In **byte stuffing (or character stuffing)**, a special byte is added to the data section of the frame when there is a character with the same pattern as the flag. The data section is stuffed with an extra byte. This byte is usually called the escape character (ESC), which has a predefined bit pattern. Whenever the receiver encounters the ESC character, it removes it from the data section and treats the next character as data, not a delimiting flag.

Character-oriented protocols present a problem in data communications. The universal coding systems in use today, such as Unicode, have 16-bit and 32-bit characters that conflict with 8-bit characters. We can say that in general, the tendency is moving toward the bit-oriented protocols that we discuss next.

Bit-Oriented Protocols
In a bit-oriented protocol, the data section of a frame is a sequence of bits to be interpreted by the upper layer as text, graphic, audio, video, and so on. However, in addition to headers (and possible trailers), we still need a delimiter to separate one frame from the other. Most protocols use a special 8-bit pattern flag 01111110 as the delimiter to define the beginning and the end of the frame.
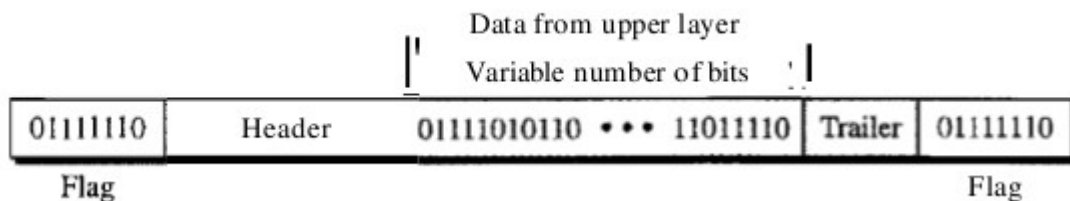

Fig:A frame in a bit-oriented protocol

This flag can create the same type of problem we saw in the byte-oriented protocols. That is, if the flag pattern appears in the data, we need to somehow inform the receiver that this is not the end of the frame. We do this by stuffing 1 single bit (instead of 1 byte) to prevent the pattern from looking like a flag. The strategy is called **bit stuffing.** In bit stuffing, if a 0 and five consecutive 1 bits are encountered, an extra 0 is added. This extra stuffed bit is eventually removed from the data by the receiver. Note that the extra bit is added after one 0 followed by five 1s regardless of the value of the next bit. This guarantees that the flag field sequence does not inadvertently appear in the frame.
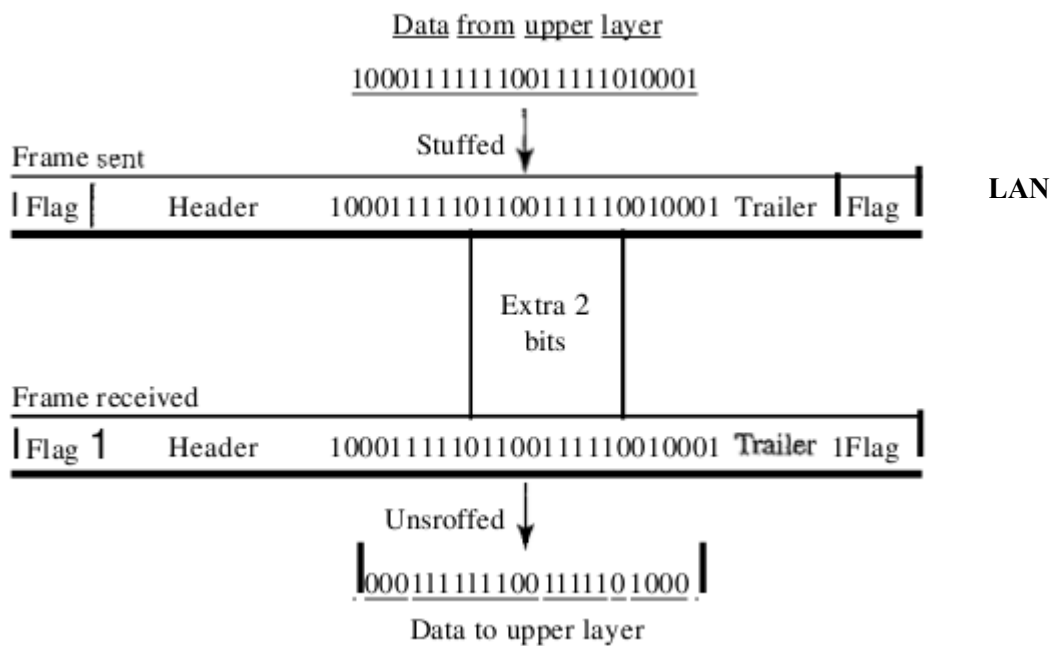
Data from upper layer

100011111110011111010001

Stuffed ↓

Frame sent

| Flag | Header | 10001111101100111110010001 Trailer | Flag | **LAN**

Extra 2 bits

Frame received

| Flag 1 | Header | 10001111101100111110010001 Trailer 1Flag |

Unsroffed ↓

|0001111111001111101000|

Data to upper layer

Fig:Bit Stuffing and unstuffing

This means that if the flaglike pattern 01111110 appears in the data, it will change to 011111010 (stuffed) and is not mistaken as a flag by the receiver. The real flag 01111110 is not stuffed by the sender and is recognized by the receiver.

Architecture:
The architecture of a LAN can be considered as a set of layered protocols.
In OSI terms, the higher layer protocols are totally independent of the LAN architecture. Hence, only lower order layers are considered for the design of LAN architecture.
The datalink layer of LAN is split into two sub layers.
       - Medium Access Control (MAC),
       - Logical Link Control Layer (LLC).
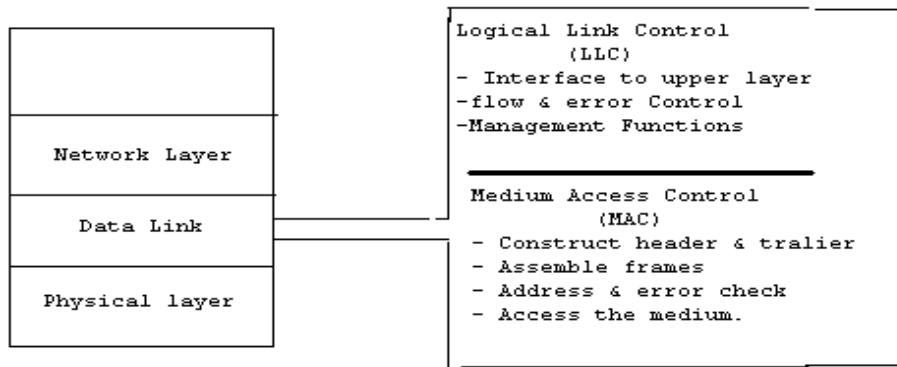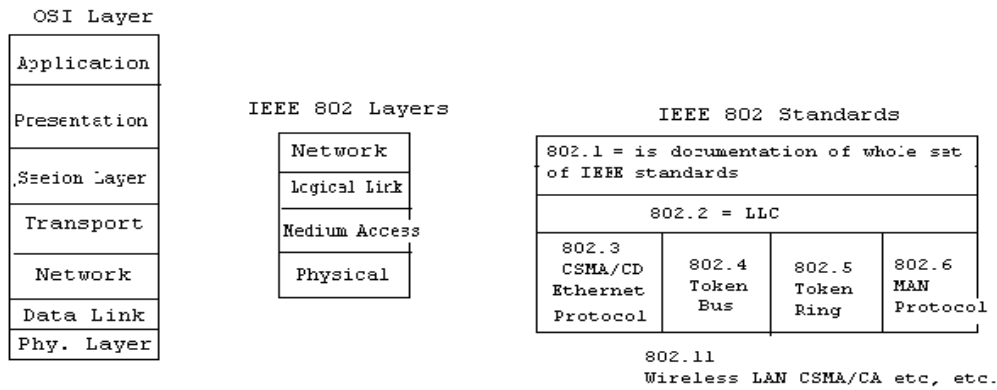The IEEE 802 committee had formulated the standards for LAN.

IEEE Standards:

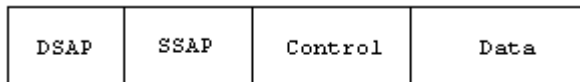

Fig:-OSI Layers for LAN

LLC Frame Format:

Fig :- LLC Frame Format

Destination Service Access Point (DSAP) -- IEEE 802.2 header begins with a 1 byte field, which identifies the receiving upper-layer process.

Source Service Access Point (SSAP) -- Following the DSAP address is the 1-byte address, which identifies the sending upper-layer process.

Control -- The Control field employs three different formats, depending on the type of LLC frame used:

- **Information (I) frame** -- Carries upper-layer information and some control information.
- **Supervisory (S) frame** -- Provides control information. An S frame can request and suspend

transmission, reports on status, and acknowledge receipt of I frames. S frames do not have an Information field.

- **Unnumbered (U) frame** -- Used for control purposes and is not sequenced. A U frame can be used to initialize secondaries. Depending on the function of the U frame, its Control field is 1 or 2 bytes. Some U frames have an Information field.

Data -- Variable-length field bounded by the MAC format implemented. Usually contains IEEE 802.2 Subnetwork Access Protocol (SNAP) header information, as well as application-specific data.
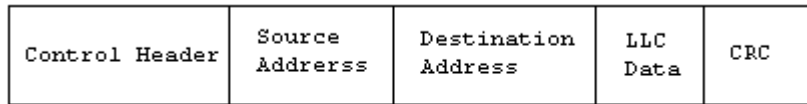
MAC Frame Format:

Control Header	Source Addrerss	Destination Address	LLC Data	CRC

Fig : General MAC frame Format

IEEE 802.3 Ethernet Frame Format:

Ethernet Protocol

A Common Data Link Layer Protocol for LANs

Frame

Field name	Preamble	Destination	Source	Type	Data	Frame Check Sequence
Size	8 bytes	6 bytes	6 bytes	2 bytes	46 - 1500 bytes	4 bytes

Preamble - used for synchronization; also contains a delimiter to mark the end of the timing information.

Destination Address - 48 bit MAC address for the destination node.

Source Address - 48 bit MAC address for the source node.

Type - value to indicate which upper layer protocol will receive the data after the Ethernet process is complete.

Data or payload - this is the PDU, typically an IPv4 packet, that is to be transported over the media.

Frame Check Sequence (FCS) - A value used to check for damaged frames.