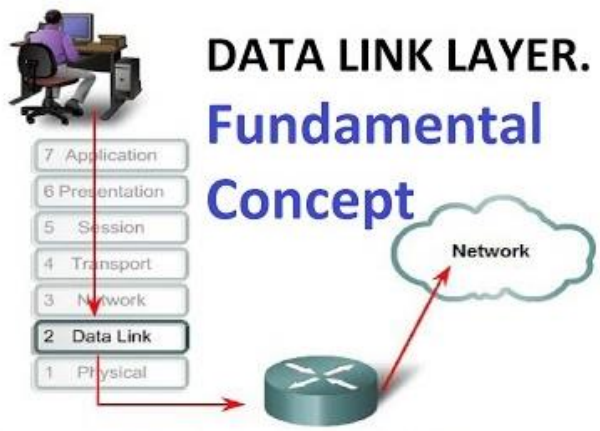
MODULE-3: Data Link Layer

VIT | Vidyalankar Institute of Technology

Accredited A+ by NAAC



Prepared by Prof. Amit K. Nerurkar

Module 3 Data Link Layer

Data Link Layer
3.1 DLL Design Issues (Services, Framing, Error Control, Flow Control), Error Detection and Correction(Hamming Code, CRC, Checksum) , Elementary Data Link protocols , Stop and Wait, Sliding Window(Go Back N, Selective Repeat)

Data Link Layer design issues:

For effective data communication between two directly connected transmitting and receiving stations the data link layer has to carry out a number of specific functions as follows:

1) ***Services provided to the network layer*:** A well-defined service interface to the network layer on source machine to the network layer on destination machine.
2) ***Frame synchronization*:** The source machine sends data in blocks called frames to the destination machine. The starting and ending of each frame should be recognized by the destination machine.
3) ***Flow control*:** The source machine must not send data frames at a rate faster than the destination machine can accept them.
4) ***Error control***: The errors made in bits during transmission from source to destination machines must be detected and corrected.
5) ***Addressing*:** On a multipoint line, such as many machines connected together (LAN), the identity of the individual machines must be specified while transmitting the data frames.
6) ***Control and data on same link*:** The data and control information is combined in a frame and transmitted from the source to destination machine. The destination machine must be able to recognize control information from the data being transmitted.
7) ***Link Management***: The initiation, maintenance and termination of the link between the source and destination are required for effective exchange of data.

For more also refer:
https://www.tutorialspoint.com/data-link-layer-design-issues

Framing

Framing is a function of the data link layer.
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 to make efficient flow control and error control

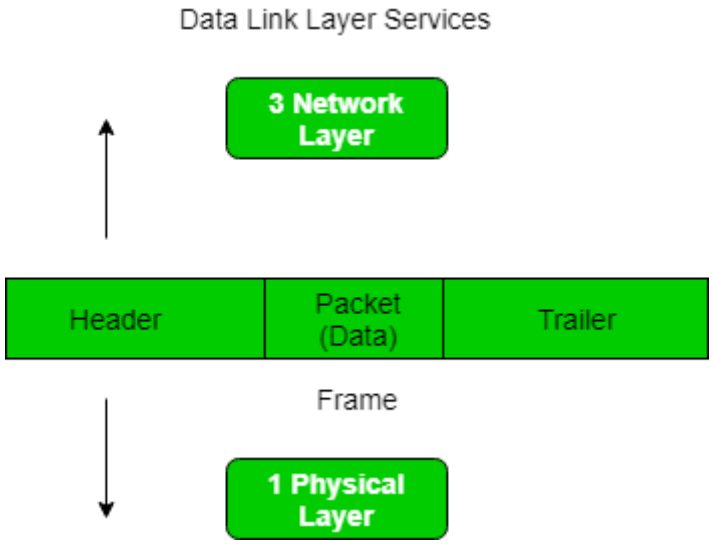


Image: https://www.geeksforgeeks.org/wp-content/uploads/1-30.png

Problems in Framing –

Detecting start of the frame

How do station detect a frame?

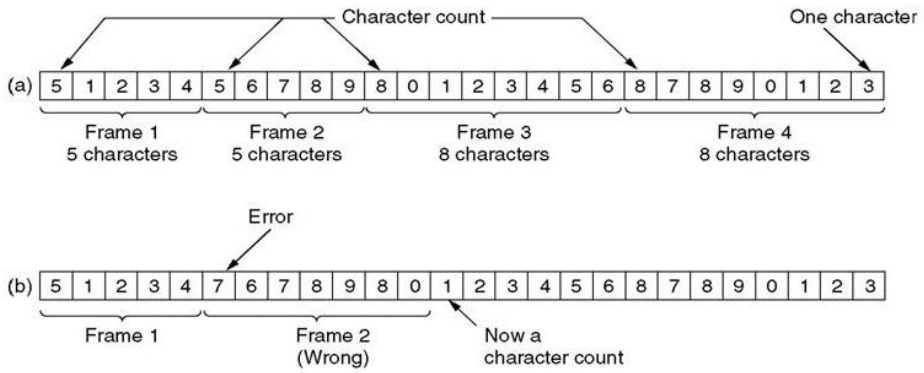
Detecting end of frame

The four different methods used for framing are:

1. Byte count.
2. Flag bytes with byte stuffing.
3. Flag bits with bit stuffing.
4. Physical layer coding violations.

Byte Count

The first framing method uses a field in the header to specify the number of bytes in the frame. When the data link layer at the destination sees the byte count, it knows how many bytes follow and hence where the end of the frame is. This technique is shown in Figure for four small example frames of sizes 5, 5, 8, and 8 bytes, respectively.



A character stream. (a) Without errors. (b) With one error.

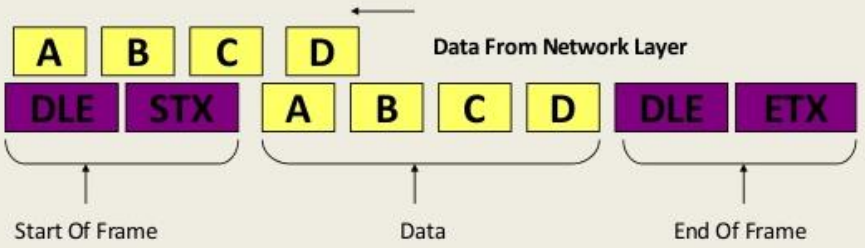
The trouble with this algorithm is that the count can be garbled by a transmission error. For example, if the byte count of 5 in the second frame of Figure (b) becomes a 7 due to a single bit flip, the destination will get out of synchronization. It will then be unable to locate the correct start of the next frame.

Flag bytes with byte stuffing

Byte - **Stuffing** − A **byte** is **stuffed** in the message to differentiate from the delimiter. This is also called character-oriented framing.

Byte-Stuffing(Character-Stuffing)(1)

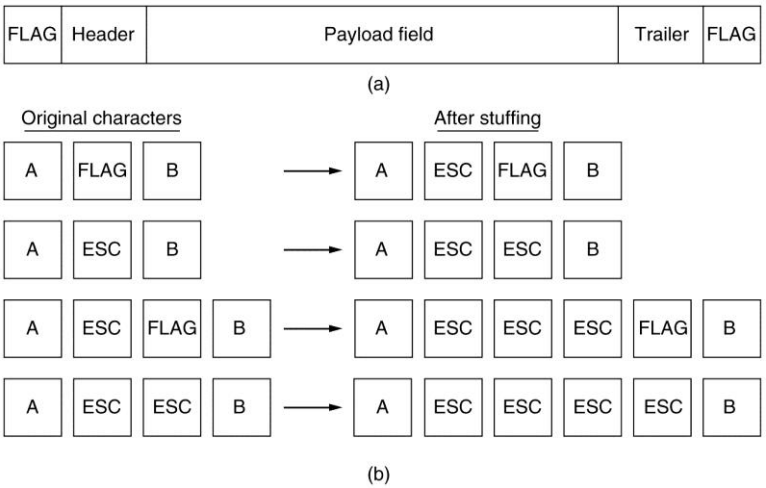
- In this method, Frame starts & end with a special character that mark the beginning & end of frame.
- Each character begins with the **ASCII** character sequence **DLE STX** (data link escape start of text) and end with **ASCII** character sequence **DLE ETX** (data link escape end of text).



Starting & Ending Characters Added By Link Layer

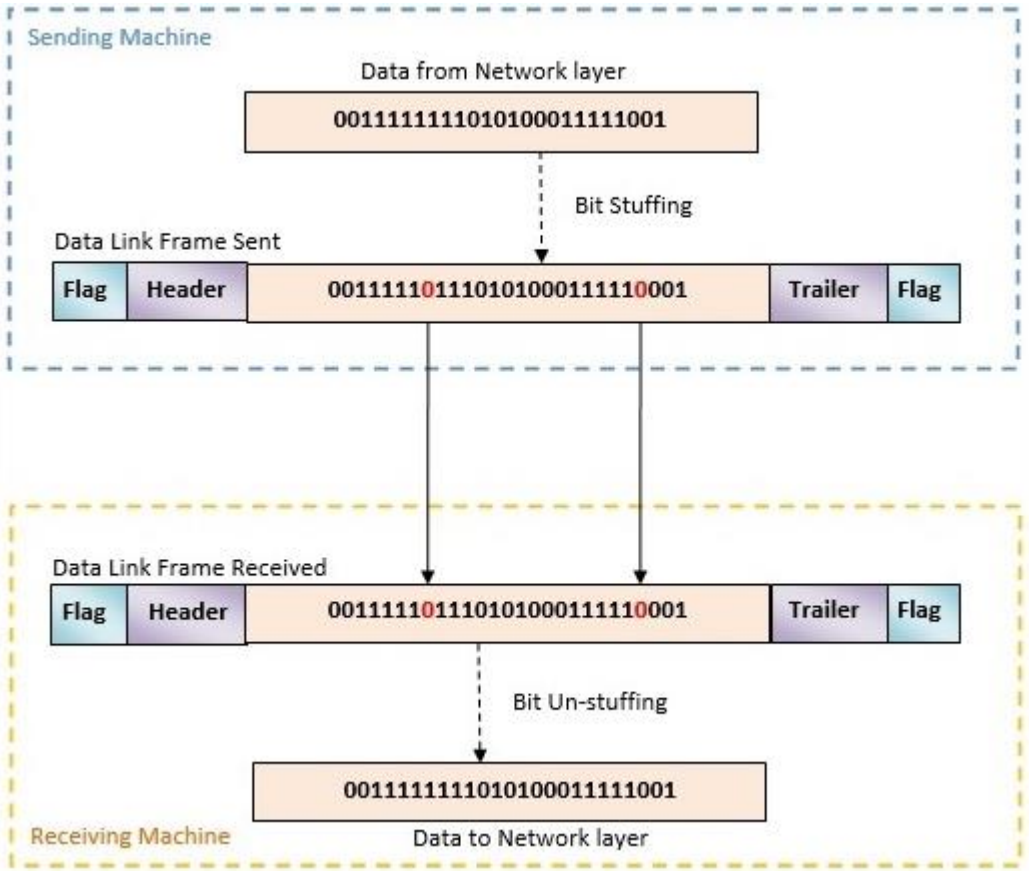
What if flag byte itself is in the data?
Insert special escape byte (ESC) before each FLAG in data. Removed at far end. This is called byte stuffing or character stuffing.

What if ESC itself is in data?
Insert another ESC before it.



Flag bits with bit stuffing

Bit stuffing is the insertion of one or more **bit**s into a transmission unit as a way to provide signaling information to a receiver. The receiver knows how to detect and remove or disregard the **stuffed bits**.

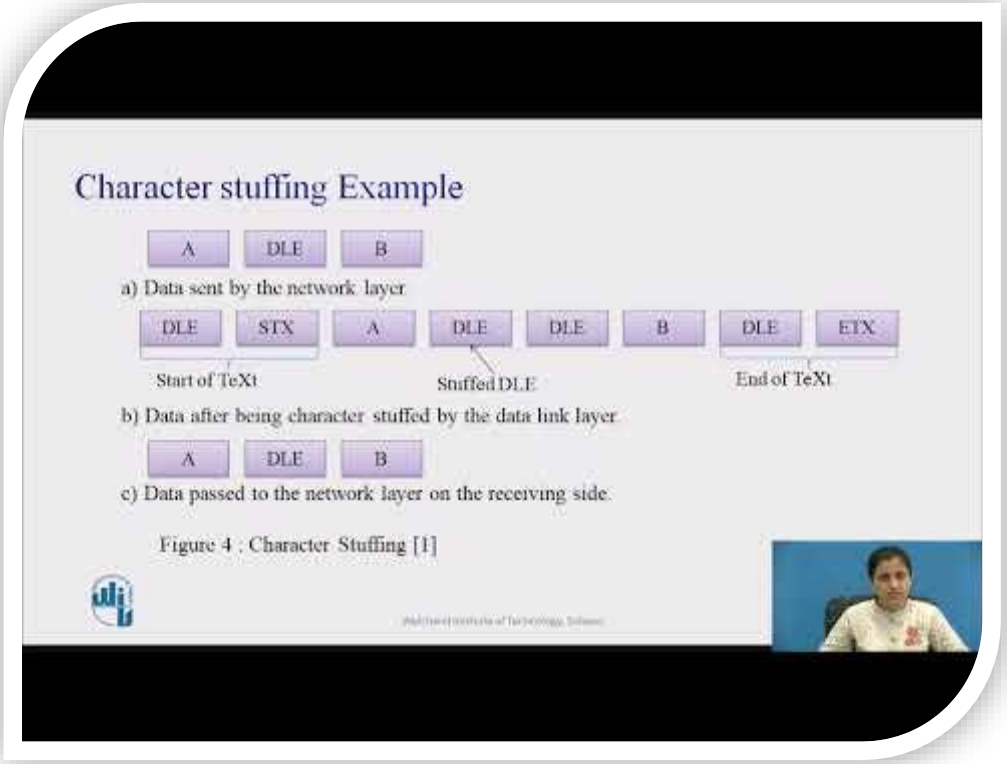


In a data link frame, the delimiting flag sequence generally contains six or more consecutive 1s. In order to differentiate the message from the flag in case of the same sequence, a single bit is stuffed in the message. Whenever a 0 bit is followed by five consecutive 1bits in the message, an extra 0 bit is stuffed at the end of the five 1s.

When the receiver receives the message, it removes the stuffed 0s after each sequence of five 1s. The un-stuffed message is then sent to the upper layers.

Video:

Data Link Layer: Framing



Physical Layer Violations Coding

Encoding of bits as signals often includes redundancy to help the receiver. This redundancy means that some signals will not occur in regular data. This means that 16 out of the 32 signal possibilities are not used. We can use some reserved signals to indicate the start and end of frames.

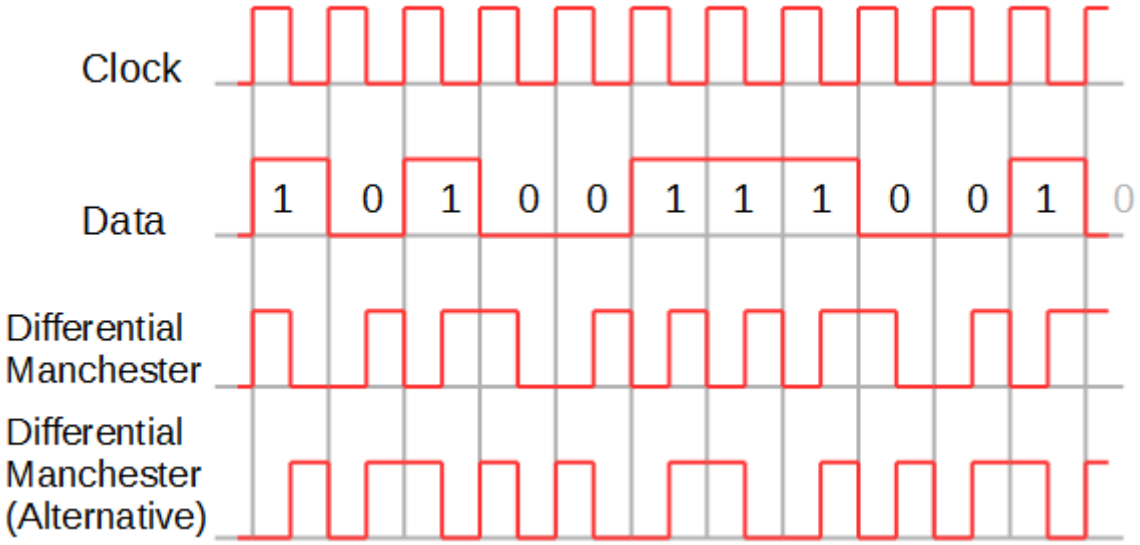


Image:https://upload.wikimedia.org/wikipedia/commons/9/9f/Differential_Manchester_encoding_alternatives.png

Like all other coding methods, Manchester code follows an algorithm to encode data. This algorithm goes like this: The data are represented NOT by logic 1 or 0, but with line transitions. A logic 0 is represented by a transition from **HIGH to LOW**, and a logic HIGH is represented by a transition from **LOW to HIGH**.

In differential Manchester, in the middle of each clock pulse, a transition occurs, regardless of the bit that was sent or is about to be sent. A data bit is send during each negative clock transition. If the data bit is 0, then a polarity transition occurs (if was HIGH it goes LOW, and if it was LOW it goes HIGH), otherwise the line remains unchanged.

References

1. TCP/IP Protocol Suite by Fourozan
2. Computer Networks by Tanenbaum
3. https://www.tutorialspoint.com/data-link-layer-design-issues
4. https://www.linktionary.com/f/framing.html
5. https://networkustad.com/2019/06/02/framing-in-data-link-layer/
6. https://www.computing.dcu.ie/~humphrys/Notes/Networks/data.framing.html
7. https://searchnetworking.techtarget.com/definition/bit-stuffing
8. https://www.tutorialspoint.com/what-is-bit-stuffing-in-computer-networks
9. http://www.pcbheaven.com/wikipages/manchester_coding/

Subscribe to my YouTube Channel Amit Nerurkar

Subjects Taught by Amit K. Nerurkar

1. C programming
2. Data Structure
3. Computer Network
4. Network Security
5. Artificial Intelligence
6. Soft Computing
7. Distributed Systems
8. Internet of Things
9. Linux Administration
10. Database Management System