Computer Network(CSC 503)

Shilpa Ingoley

Lecture 12

3. 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)

3.2 **Medium Access Control sublayer**

- Channel Allocation problem, Multiple access Protocol(Aloha, Carrier Sense Multiple Access (CSMA/CD)

DLL Design Issues

- DLL uses the services of the physical layer to send and receive bits over communication channels
- **Functions of DLL**
  - 1. Providing a well-defined service interface to the network layer.
  - 2. Dealing with transmission errors.
  - 3. Regulating the flow of data so that slow receivers are not swamped by fast senders.

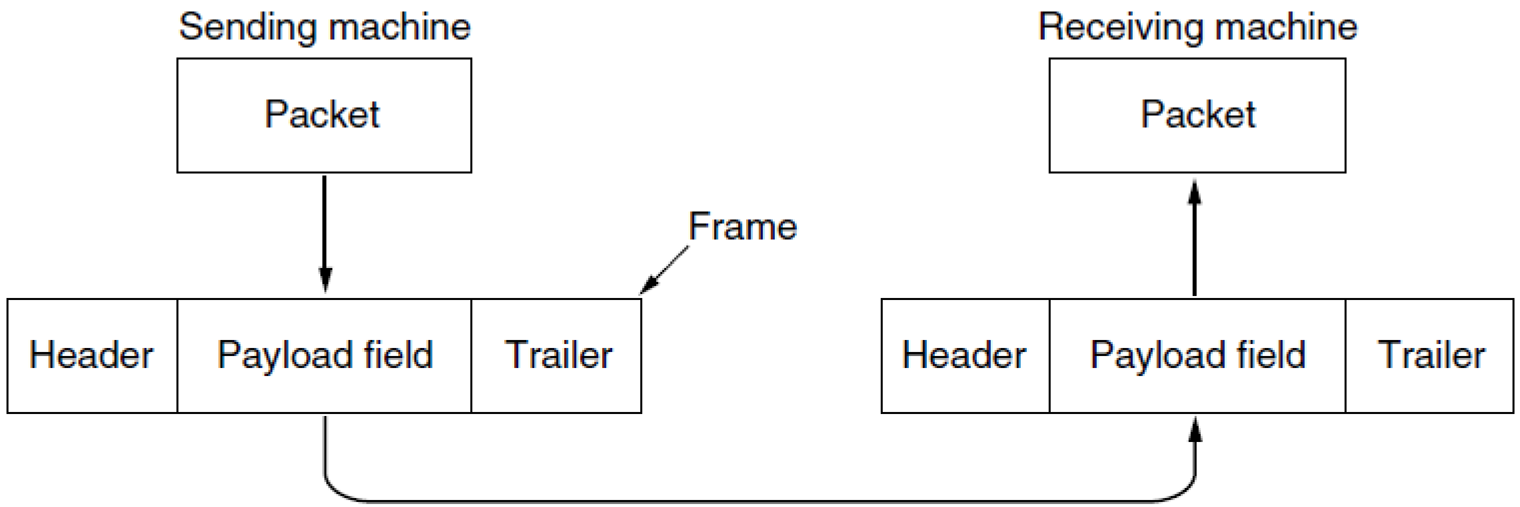


Fig : Relationship between packets and frames.

Services Provided to the Network Layer

The data link layer offers three types of services.

1. Unacknowledged connectionless service.
2. Acknowledged connectionless service.
3. Acknowledged connection-oriented service.

- Note: Actual services that are offered vary from protocol to protocol.

Contd...

Virtual Communication: No physical medium is present for Data Link Layer to transmit data. It can be only be **visualized and imagined** that two Data Link Layers are communicating with each other with the help of or using data link protocol.

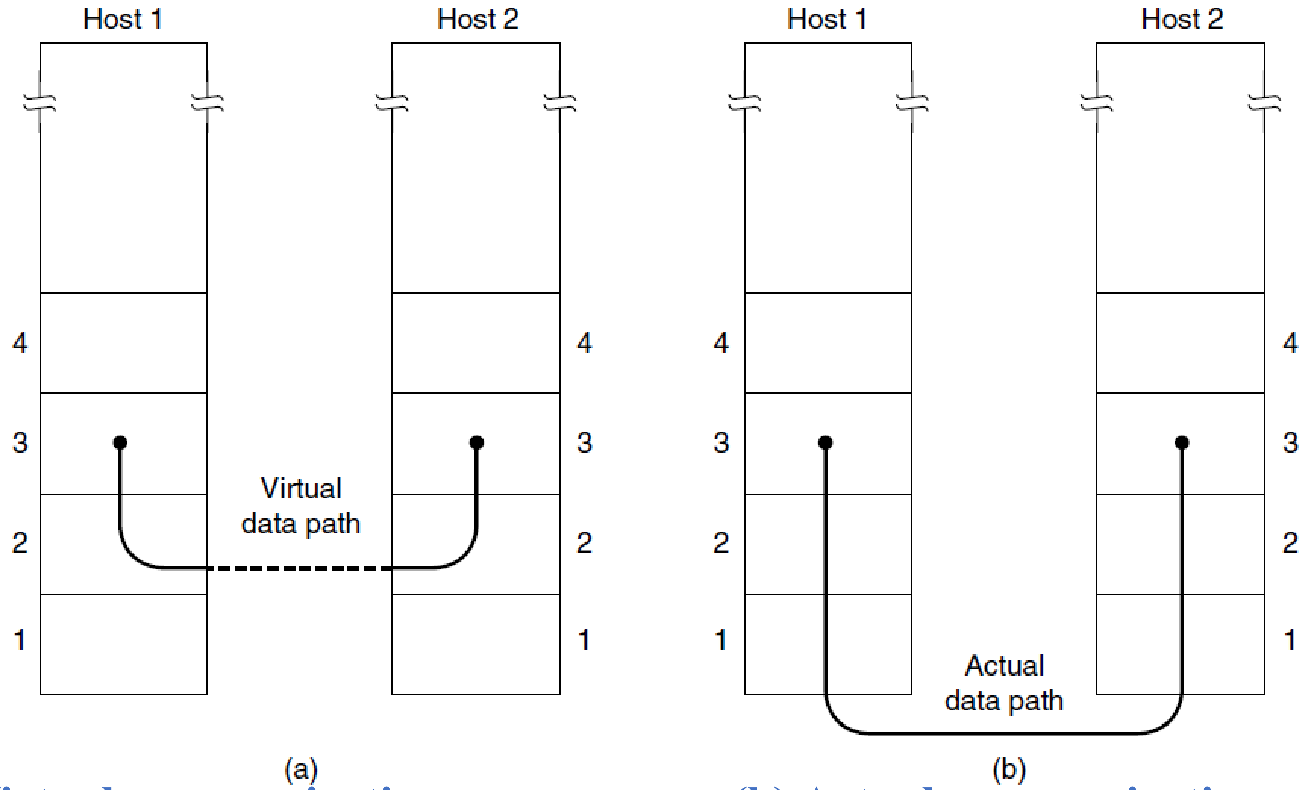


Fig: (a) Virtual communication. (b) Actual communication.

Actual communication: The **actual path** is **Network Layer -> Data link layer -> Physical Layer** on sending machine, then to physical media and after that to **Physical Layer -> Data link layer -> Network Layer** on receiving machine.

Contd...

1. **Unacknowledged connectionless service**
   - Here, the data link layer of the sending machine sends independent frames to the data link layer of the receiving machine.
   - The receiving machine does not acknowledge receiving the frame.
   - No logical connection is set up between the host machines.
   - Error and data loss is not handled in this service.
   - This is applicable in Ethernet services and voice communications.

2. **Acknowledged connectionless service**
   - Here, no logical connection is set up between the host machines, but each frame sent by the source machine is acknowledged by the destination machine on receiving.
   - If the source does not receive the acknowledgment within a stipulated time, then it resends the frame. This is used in Wifi (IEEE 802.11) services.

Contd...

3.Acknowledged connection-oriented service

- This is the best service that the data link layer can offer to the network layer.
- A logical connection is set up between the two machines and the data is transmitted along this logical path.
- The frames are numbered, that keeps track of loss of frames and also ensures that frames are received in correct order.

The service has three distinct phases −

- **Set up of connection** –A logical path is set up between the source and the destination machines. Buffers and counters are initialized to keep track of frames.
- **Sending frames** –The frames are transmitted.
- **Release connection** –The connection is released, buffers and other resources are released.

- It is appropriate for satellite communications and long-distance telephone circuits.

Framing

- Determines how the bits of the physical layer are grouped into frames (framing)
- The Data Link Layer is the second layer in the OSI model, above the Physical Layer, which ensures that the error free data is transferred between the adjacent nodes in the network. It breaks the datagram passed down by above layers and converts them into frames ready for transfer. This is called Framing.

Contd…

Two types of framing

- Frames can be of fixed or variable size.

1)Fixed-Size Framing

- In fixed-size framing, there is no need for defining the boundaries of the frames; the size itself can be used as a delimiter

2)Variable size framing

- The contents of each frame are encapsulated between a pair of reserved characters or bytes for frame synchronization.

There are four methods:

1. Character count.
2. Flag bytes with byte stuffing.
3. Starting and ending flags, with bit stuffing.
4. Physical layer coding violations.

Framing Methods

1. Character count

- The first framing method that is character count method uses a field in the header to specify the number of characters in the frame. When the data link layer at the destination sees the character count, it knows how many characters follow and also can come to know where the frame will end.
  - There are **four** frames in this figure with size **5, 5, 8** and **8**. The first frame has data A, H, 7 and 8.

5	A	H	7	8	**5**	2	7	9	7	**8**	s	e	2	R	4	6	6	**8**	3	5	G	9	J	A	E

- There is one problem with this method. The count can be garbled by a transmission error

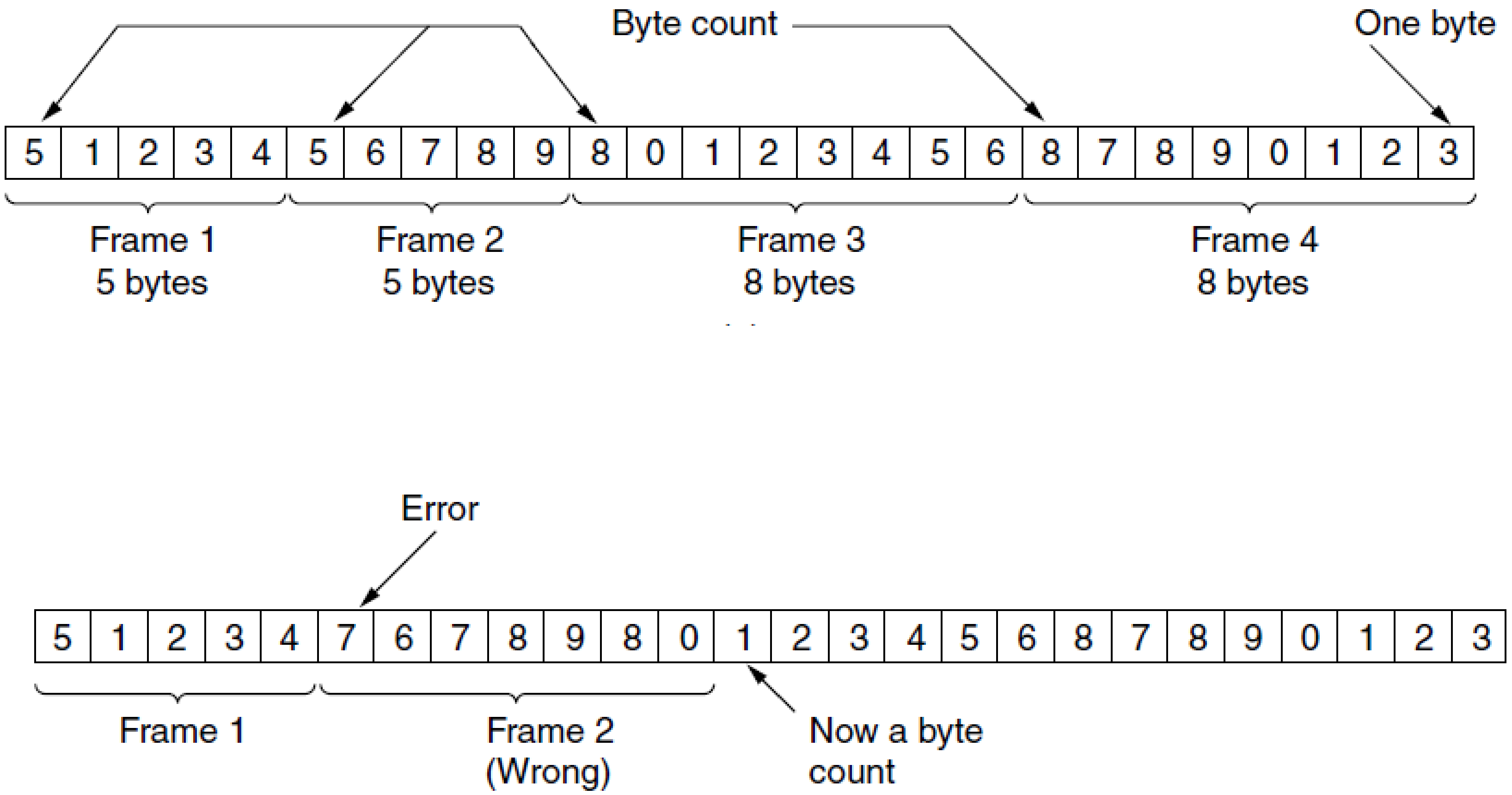
Contd…

- If the character count of 5 in the second frame becomes 7, the destination will get out of synchronization and will be **unable to locate the start of the next frame**.
- ▸For this reason the character count method is rarely used nowadays.
- **Example1:**

5	A	H	7	8	**7**	2	7	9	7	**8**	s	e	2	R	4	6	6	**8**	3	5	G	9	J	A	E

Contd...

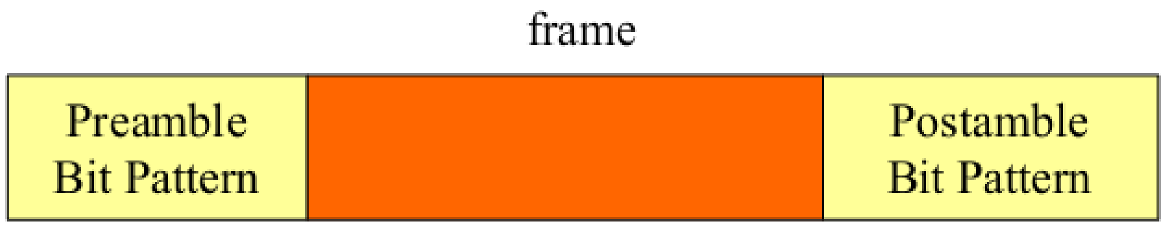
Example2:



Contd…

- Alternative to character count

The contents of each frame are *encapsulated* between a pair of reserved characters or bytes for frame synchronization.



Contd…

2. Flag bytes with byte stuffing/Character stuffing.

- The second framing method is flag bytes insertion using **byte stuffing**. This is also referred to as **character stuffing**.
- ASCII characters are used as framing **delimiters** for example **DLE STX** (start of the frame) and **DLE ETX**(End of the frame).
- If however, binary data is being transmitted then there exists a possibility of the characters DLE STX and DLE ETX occurring in the data. Since this can interfere with the framing, a technique called character stuffing is used. The sender's data link layer inserts an ASCII DLE character just before the DLE character in the data. The receiver's data link layer removes this DLE before this data is given to the network layer. However character stuffing is closely associated with 8-bit characters and this is a major hurdle in transmitting arbitrary sized characters. The problem occurs when these character patterns occur within the data.

Contd…

- This framing mechanism uses the special ASCII characters 0x10 (called DLE or Data Link Escape), 0x02 (STX, Start of Text), 0x03 (ETX, End of Text). Each packet is framed as between "DLE STX" and "DLE ETX" as follows:
- 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
- | DLE | STX | Payload (DLEs byte-stuffed) ... | DLE | ETX |
- ▶So, when sending a packet on a stream, just transmit "DLE STX", then the payload of bytes (the content of the packet), and finally "DLE ETX".
- ▶**When sending the payload data, each DLE character in the data must be doubled (i.e., DLE becomes DLE DLE).**

Contd…

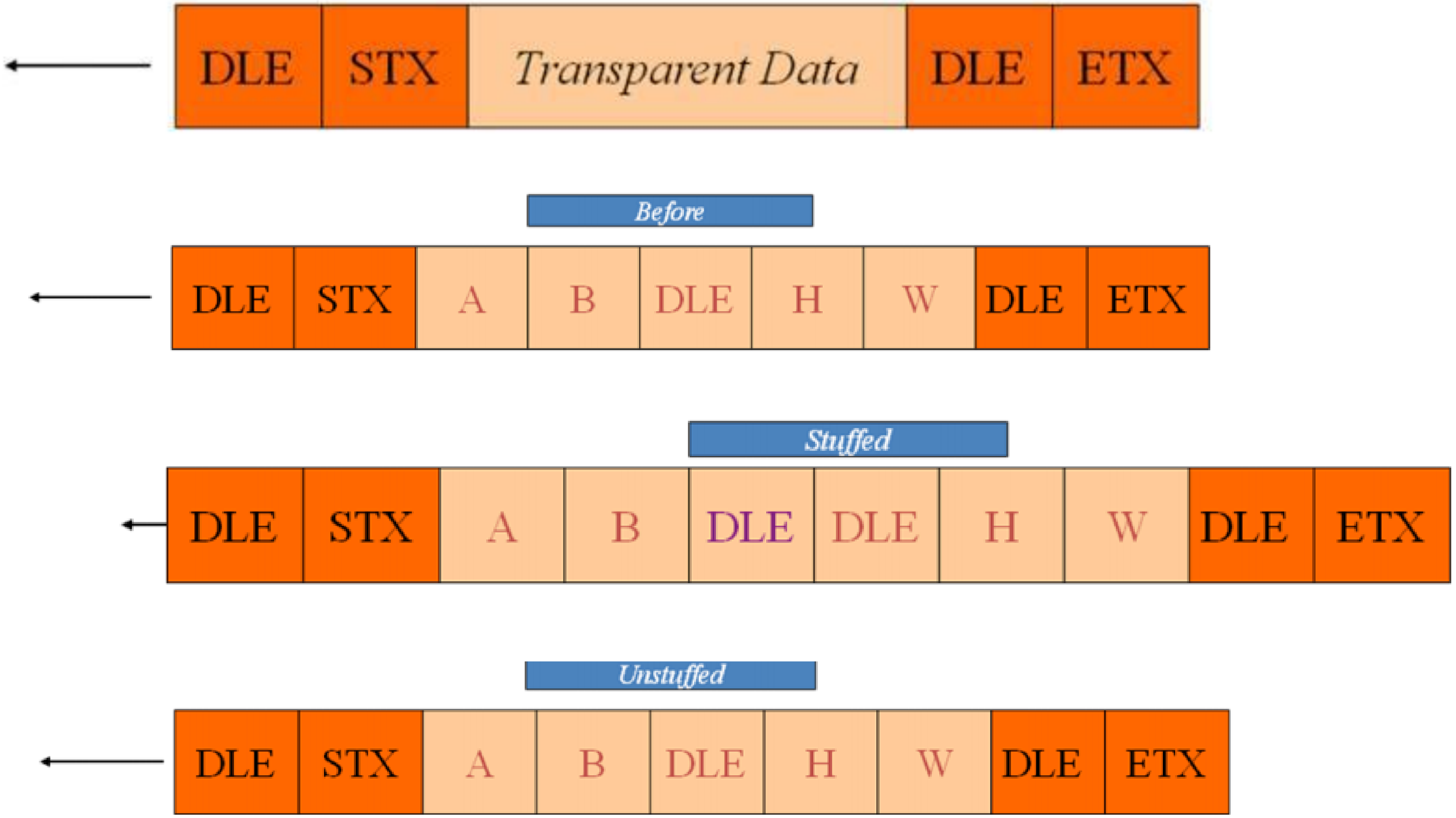
- Solution to this problem is to stuff an **extra DLE** into the data stream just before each occurrence of an "accidental" DLE in the data stream. The data link layer on the receiving end unstuffs the DLE before giving the data to the network layer as shown in this figure. The flag byte DLE and STX are inserted at the start and DLE ETX bytes are inserted in the end.

Contd…

- If you want to transmit the payload “DLE“.
- ▶The final packet is **DLE STX** DLE DLE **DLE ETX**.
- ▶A receiver must therefore look for packet start (i.e. DLE STX), read the payload and remove the second DLE if **two consecutive ones** are received (de-stuffing), and continue buffering until DLE ETX is found.

Contd...

- Example:



Contd...

- Example 2: **In this example ESC char is stuff**
-

Fig: (a) A frame delimited by flag bytes.

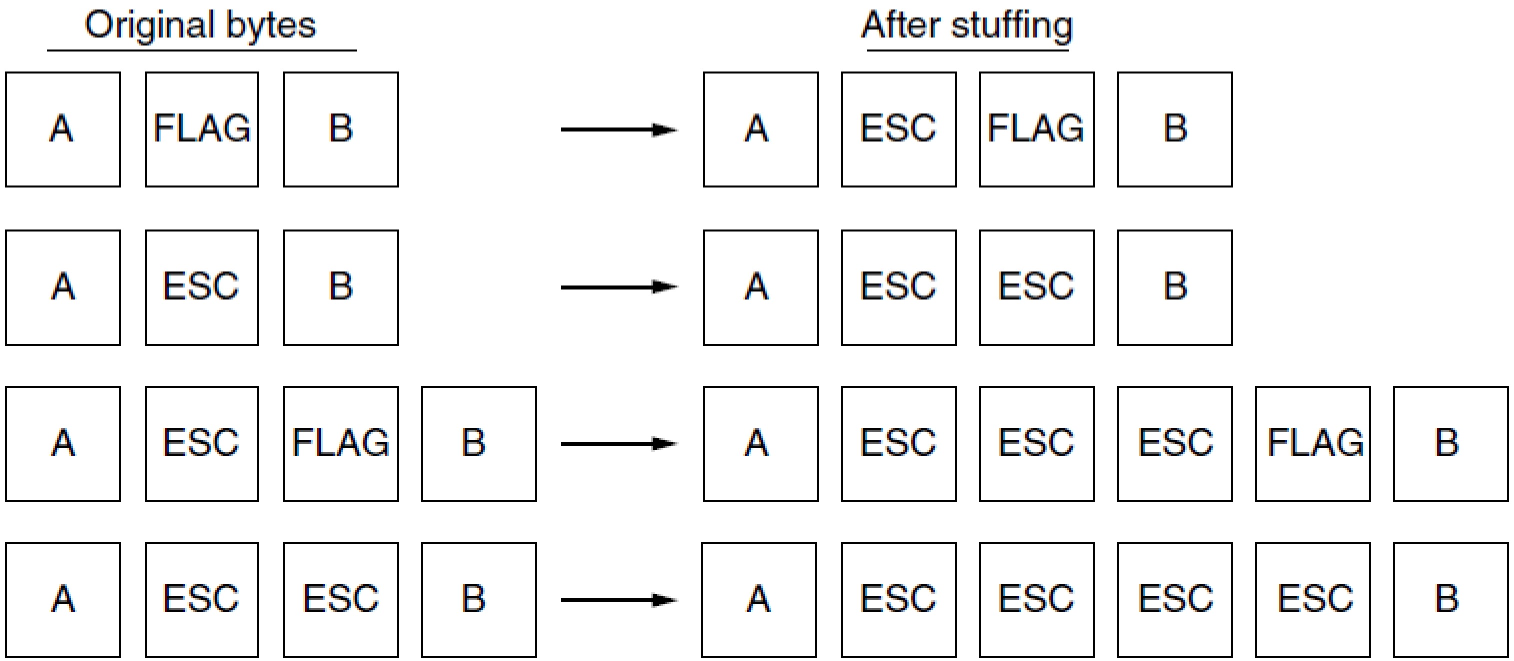


Fig: (b) Four examples of byte sequences before and after byte stuffing.

Contd…

3. Bit Stuffing

- In this method each frame begins and ends with a special bit pattern called a flag byte which is 01111110.
- In this technique also there is a possibility that this particular pattern of 01111110 may appear in data itself.
- In order to take care of this problem, whenever sender data link layer encounters five consecutive ones in the data stream, it automatically stuff a 0 bit into the outgoing stream.
- When the receiver sees **five consecutive** incoming ones followed by a 0 bit , it automatically destuffs the 0 bit before sending the data to the network layer.

Contd...

- **Example 1:**

(a)

Data to be sent

$$0110\underbrace{11111}\underbrace{11111}00$$

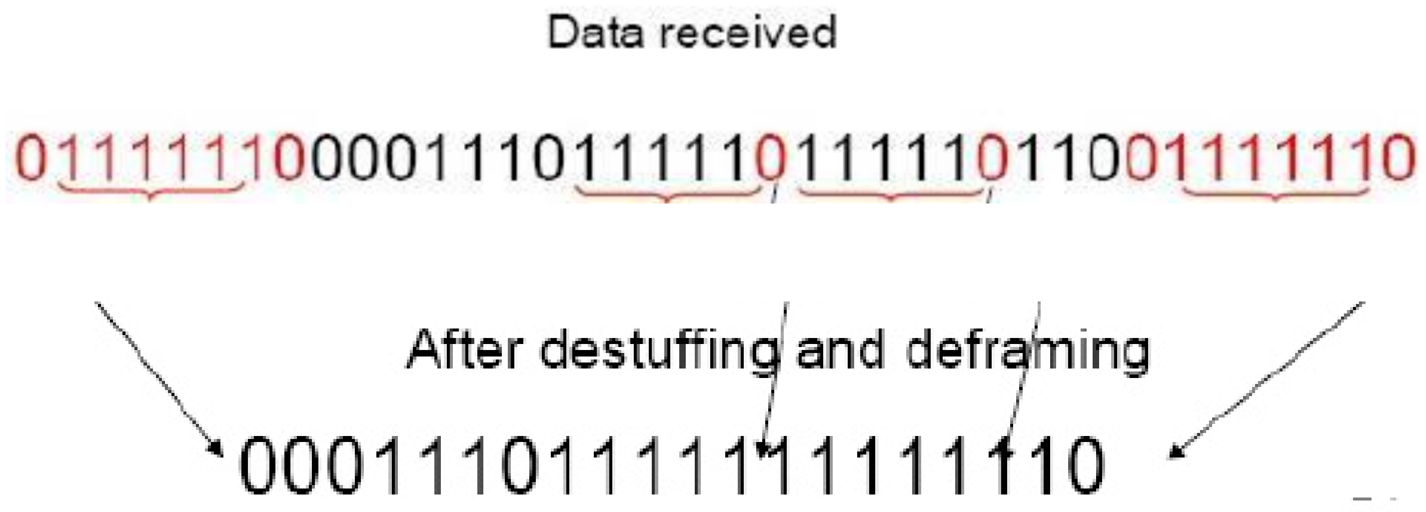
After stuffing and framing

$$\mathit{01111110}\,011011111\underline{0}11111\underline{0}00\,\mathit{01111110}$$

- With bit stuffing, the boundary between two frames can be unambiguously recognized by the flag pattern.

Contd…

- **Example 2:**



Contd…

Example 3:

Input stream of data

0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

(a) The original data

0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0

Stuffed bits

(b) The data as they appear on the line

0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

(c) The data as they are stored in the receiver’s memory after destuffing.

Contd...

- **Example 4:**

Input Stream

0110111111100111110111111111100000

Stuffed Stream

01101111101100111110011111011111000000

Stuffed bits

Unstuffed Stream

0110111111100111110111111111100000

4. Physical layer coding violations : **Encoding Violations**

- The **encoding of bits** as signals often **includes redundancy** to help the receiver. This redundancy means that some signals will not occur in regular data.
- **Example :** In the 4B/5B line code **4 data bits** are **mapped to 5 signal bits** to ensure sufficient bit transitions.
- 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. In effect, we are using **"coding violations"** to delimit frames.
- In these technique they are reserved signals, it **is easy to find the start and end of frames** and there is **no need to stuff the data**. Many data link protocols use a combination of these methods for safety.
- A common pattern used for **Ethernet and 802.11** is to have a **frame begin with a well-defined** pattern called a **preamble**.
- This pattern might be **quite long (72 bits is typical for 802.11)** to allow the receiver to prepare for an incoming packet. The **preamble** is then **followed** by a **length** (i.e., count) **field** in the header that is used to **locate the end of the frame**.

Contd…

- An encoding technique converts a stream of data bits in a predefined code that can be recognized by both the transmitter and the receiver. Using predefined patterns helps to differentiate data bits from control bits and provide better media error detection.
- 4b/5b encoding is **a type of 'Block coding'**. This processes groups of bits rather than outputting a signal for each individual bit (as in Manchester encoding). A group of 4 bits is encoded so that an extra 5th bit is added. Since the input data is taken 4-bits at a time, there are 2^4, or 16 different bit patterns.
- A group **of 4 bits is encoded so that an extra 5th bit is added**. Since the input data is taken 4-bits at a time, there are 2^4, or 16 different bit patterns. The encoded bits use 5-bit, and hence have 2^5 or 32 different bit patterns.
- 4B/5B code = a very clever way to provide synchronization opportunities to the receiver by: **Encoding 4 bits using 5 bits**. **The encoding of** 5 bits makes sure that there is a transition within the 5 bits
- Advantages of 4b/5b encoding: **More bandwidth efficient (only 25% overhead)**. Allows extra codes to be used for control information.