Data Communications and Networking

UNIT-2

Data Link Layer

- Objective:
  - Achieving reliable communication between two adjacent machines
- Design Issues:
  - Framing: data are sent in blocks called frames, the beginning and end of each frame must be recognized by the receiver.
  - Error control: bit errors introduced by the transmission system should be detected and/or corrected.
  - Flow control: the sending station must not send frames at a rate faster than the receiving station can absorb them.
  - Addressing: on a multipoint line, such as a LAN, the identity of the two stations involved in a transmission must be specified.
  - Transmit control information and data on the same line

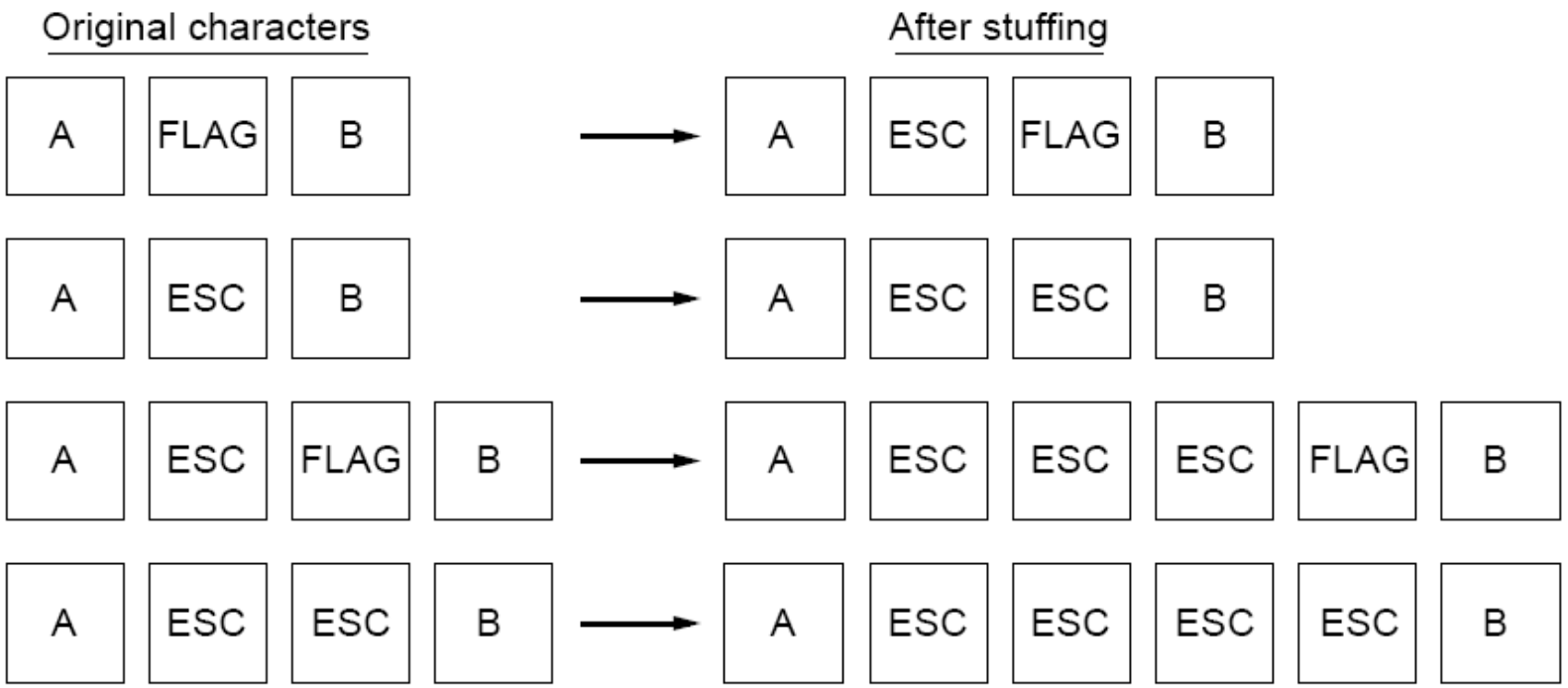
Framing

- Large block of data may be broken up into small frames at the source because:
  - limited buffer size at the receiver
  - A larger block of data has higher probability of error
    - With smaller frames, errors are detected sooner, and only a smaller amount of data needs to be retransmitted
  - On a shared medium, such as Ethernet and Wireless LAN, small frame size can prevent one station from occupying medium for long periods

Framing

- Need to indicate the start and end of a block of data
- Use preamble (e.g., flag byte) and postamble
- If the receiver ever loses synchronization, it can just search for the flag byte.
- Frame: preamble + control info + data + postamble
- Problem: it is possible that the flag byte's bit pattern occur in the data
- Two popular solutions:
  - Byte stuffing
    - The sender inserts a special byte (e.g., ESC) just before each "accidental" flag byte in the data (like in C language, " is replaced with \").
    - The receiver's link layer removes this special byte before the data are given to the network layer.
  - Bit stuffing: each frame starts with a flag byte "01111110".
    - Whenever the sender encounters five consecutive 1s in the data, it automatically stuffs a 0 bit into the outgoing bit stream.
    - When the receiver sees five consecutive incoming 1 bits, followed by a 0 bit, it automatically deletes the 0 bit.

Byte Stuffing

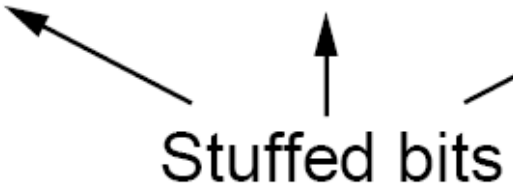


Four examples of byte sequences before and after byte stuffing

Bit Stuffing

(a) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

(b) 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



(c) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

Bit stuffing:

(a) The original data.

(b) The data as they appear on the line.

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

Error Detection: Types of Error

- An error occurs when a bit is altered between transmission and reception
- Single bit errors
  - One bit is altered
  - Adjacent bits are not affected
  - Can occur in the presence of white noise (thermal noise)
- Burst errors
  - A cluster of bits with Length *B*
  - the first and the last and a number of intermediate bits in error (not necessarily all the bits in the cluster suffer an error)
  - More common and more difficult to deal with
  - Can be caused by impulse noise

Data Link Protocols

- Specifications to implement data link layer
- Asynchronous Protocols:
  - Primarily used in modems
  - Feature start and stop bits and variable length gaps between characters
- Synchronous Protocols:
  - Developed for higher speed networks
  - No start and stop bits, much lower overhead, faster transmission