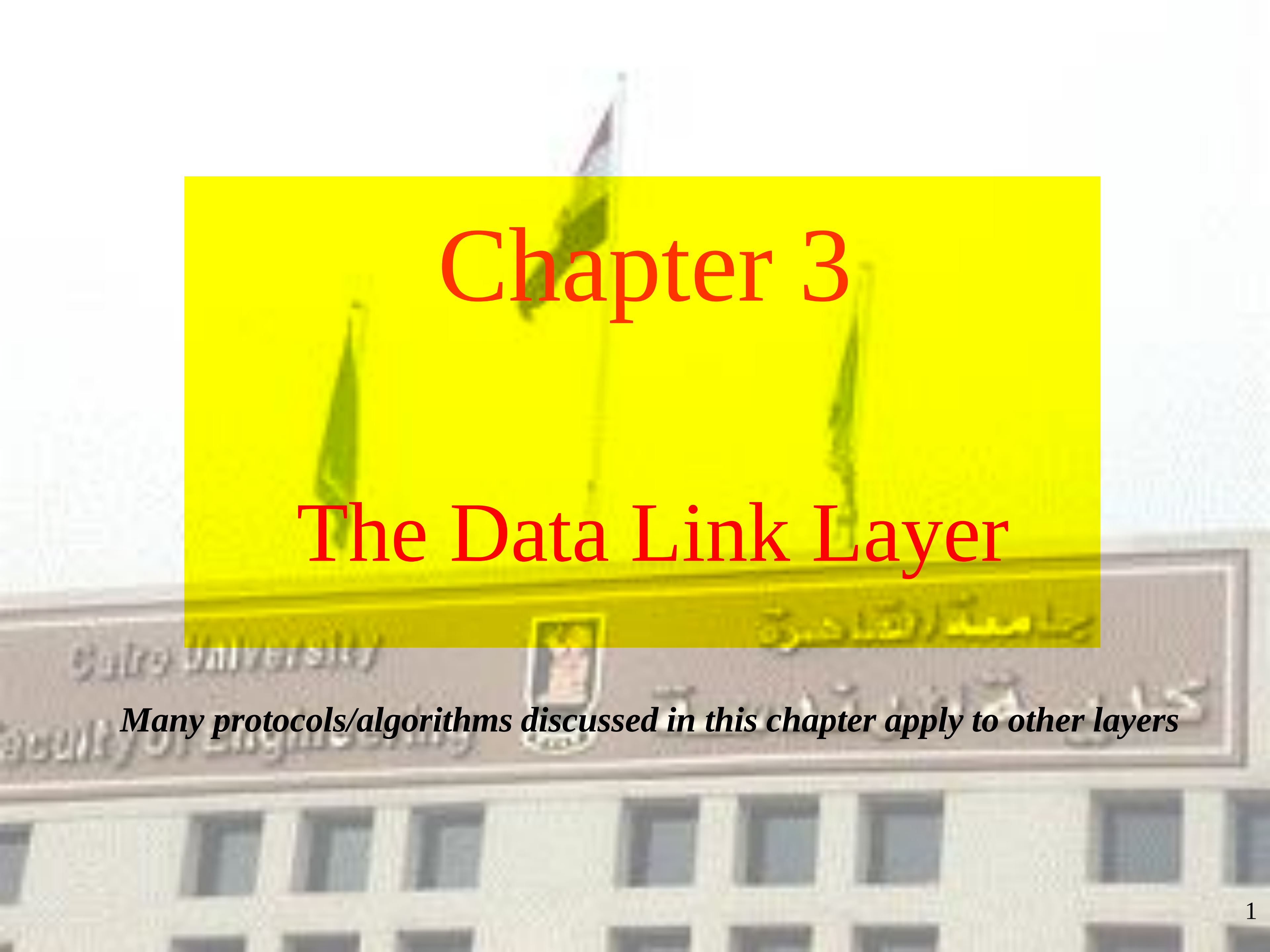


Chapter 3

The Data Link Layer

Many protocols/algorithms discussed in this chapter apply to other layers

Functions of the Data Link Layer

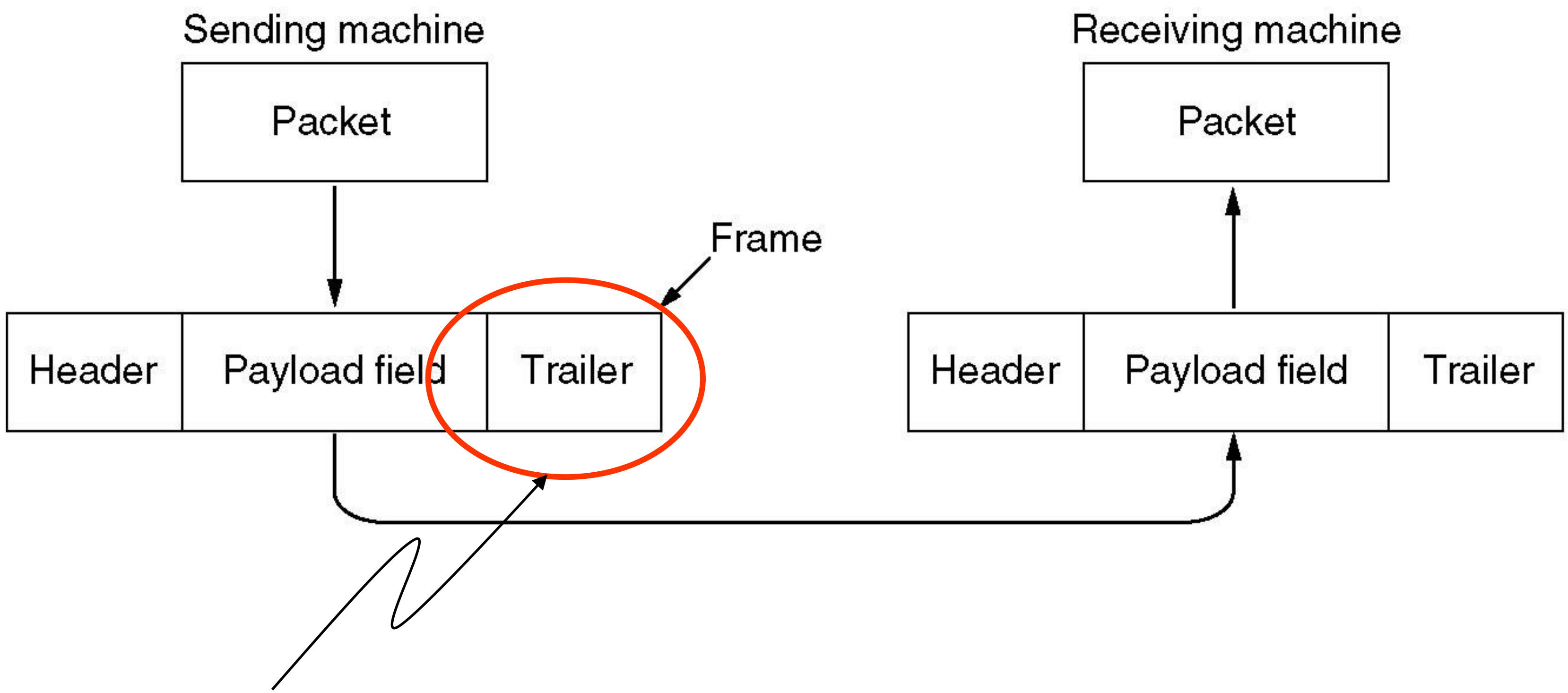
- Provide service interface to the network layer
- Dealing with transmission errors.
  - Either error free or detectable
- In broadcast networks there are additional problems
  - Co-ordination (e.g. CDMA, token, central arbiter)
  - Local Addressing (within a network)
- Regulating data flow
  - Slow receivers not swamped by fast senders
  - Not always implemented at the datalink layer
  - Provided only up to the interface to network layer
    - Does not guard against packet drop in upper layer

Data Link Layer Design Issues

- Services Provided to the Network Layer
  1. Framing
  2. Error Control
  3. Flow Control
  4. Local Addressing
- Many of the above functions are also provided at upper layers. E.g.
  - E.g. Error control and flow control is provided at the transport layer
  - The principle is the same in all layers

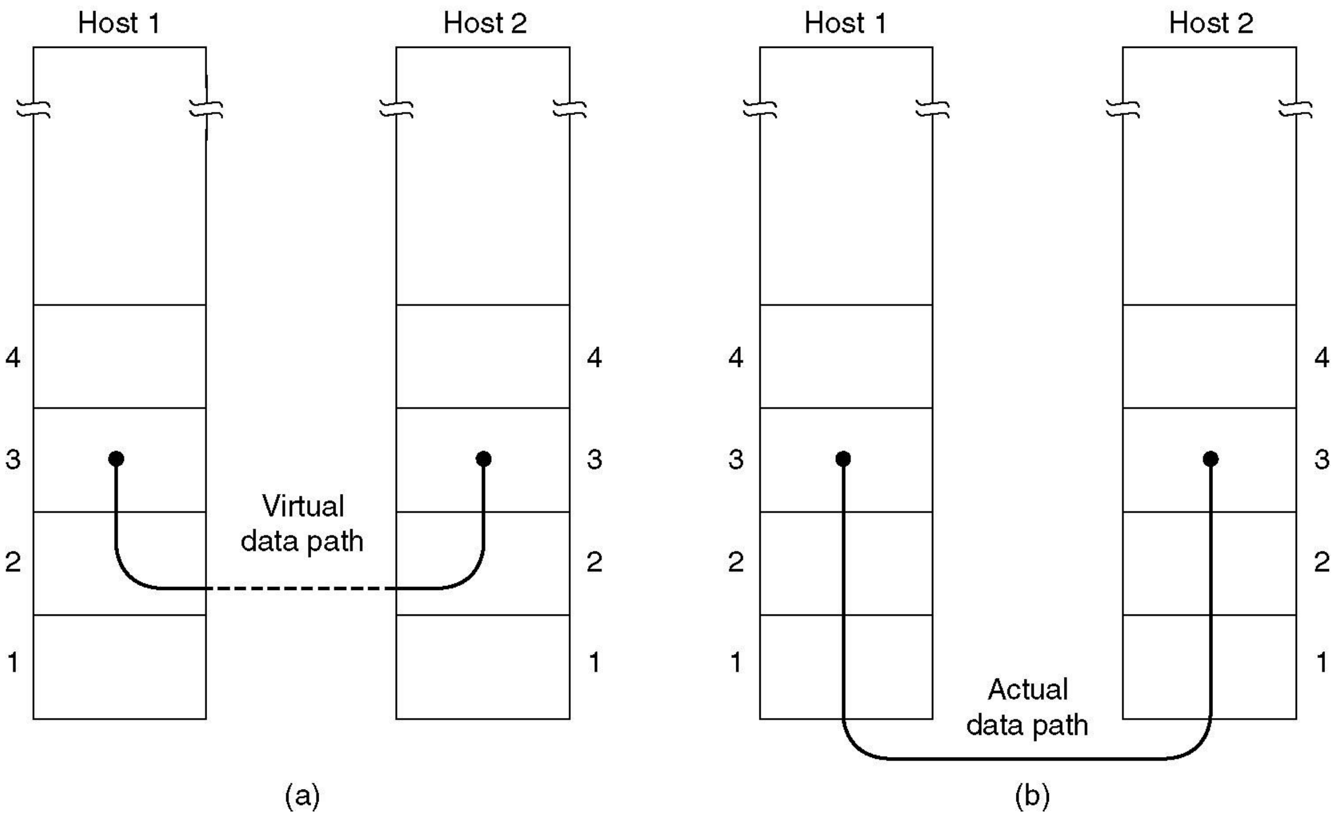
Relation Between Packets and Frames

Relationship between packets and frames.



Not all datalink protocols use trailers

Services Provided to Network Layer



Types of Connections

- Unacknowledged connectionless
  - Usually on reliable links (e.g fiber).
  - Extra weight and complexity of acknowledgement not necessary
- Acknowledged connectionless
  - Good for unreliable links* (e.g. wireless)
  - Sub-optimal for reliable channels (e.g. fiber)
  - No need for connection establishment
    - Mobility and simplicity
    - No need to keep state with every peer
  - Can deliver *out of order* packets
- Connection oriented
  - Establish connection prior to sending
  - Information is acknowledged
  - Easy to provide flow control
  - Each packet gets a sequence number
    - In-order delivery
    - Easy to detect lost packet and expedite retransmit (E.g. by sending NACK)
  - PPP, HDLC

1. Framing

- Framing is breaking up stream of bits into well defined bounded chunks

- Good for reliability
  - Error detection/correction codes
  - Allows for Ack

- We' ll talk about 3 frame demarcation methods

  - Character count
  - Flag bytes with byte stuffing
  - Starting and ending flags with bit stuffing

2. Error Control

- Error control repairs frames that are received in error
  - Requires errors to be detected at the receiver
  - Typically retransmit the unacknowledged frames
  - Timer protects against lost acknowledgements

- Detecting errors and retransmissions are next topics.

3. Flow Control

- Prevents a fast sender from out-pacing a slow receiver
- Two methods are generally used:
  1. Receiver gives feedback on the data it can accept
     - Rare in the Link layer as network interface cards "NICs" run at "wire speed"
     - Receiver can take data as fast as it can be sent
  2. Rate-based flow control, the protocol has a built-in mechanism that limits the rate at which senders may transmit data, without using feedback from the receiver (used in Transport Layer)
- Flow control is a topic in the Data Link and Transport layers

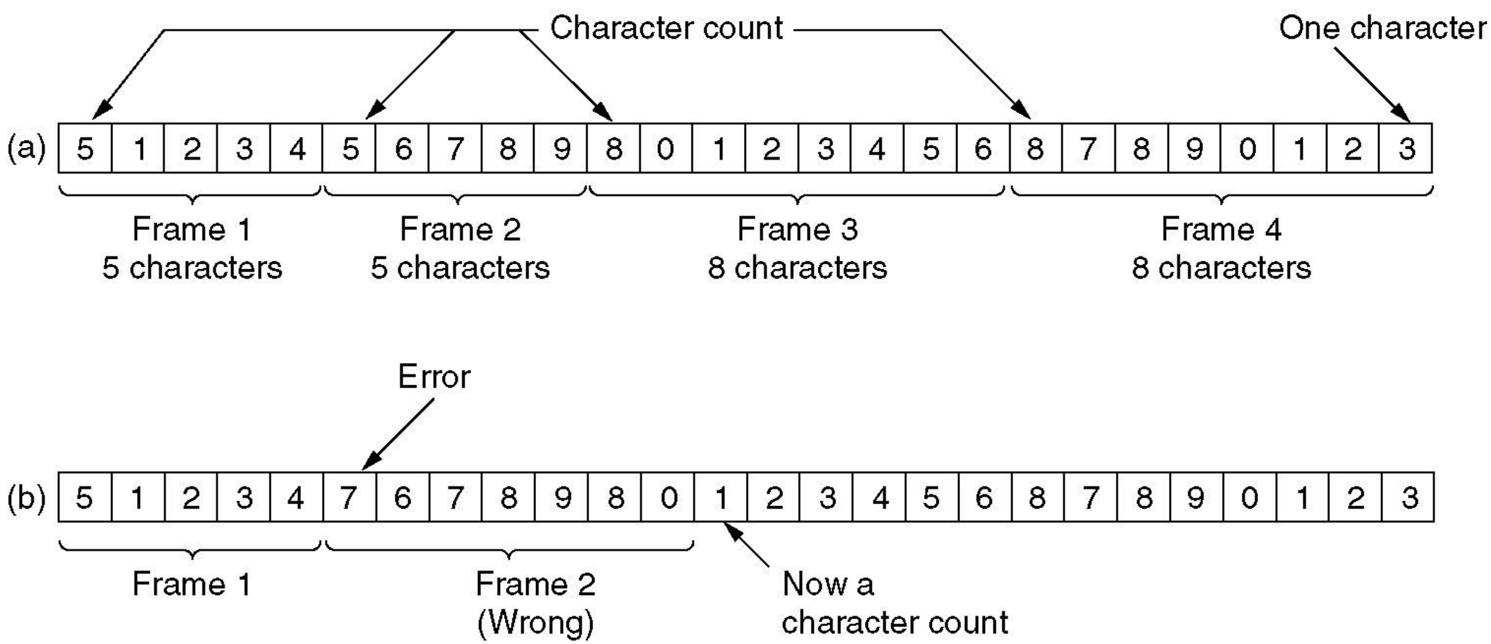
1. Framing

- We'll talk about 3 frame demarcation methods

1. Character count
2. Flag bytes with byte stuffing
3. Starting and ending flags with bit stuffing

Framing Using Character Count

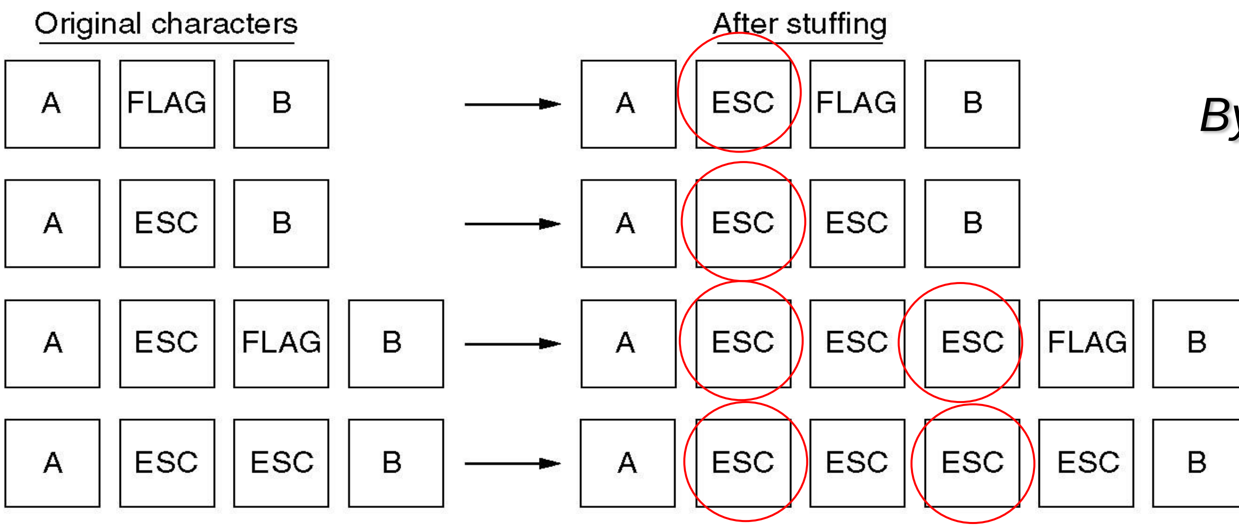
- Idea
  - Put the character count in the header of each frame



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

Framing Using Flag Byte

(a)



Byte Stuffing

(b)

Flag is called Frame Sync Sequence (FSS) as in HDLC

(a) A frame delimited by FLAG bytes.

I.e. Each frame is started and ended by a FLAG byte

The Start and end FLAG bytes may be identical

Why it is better to have a FLAG byte at the beginning and the end of each frame instead of just one FLAG byte at the beginning of each frame?

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

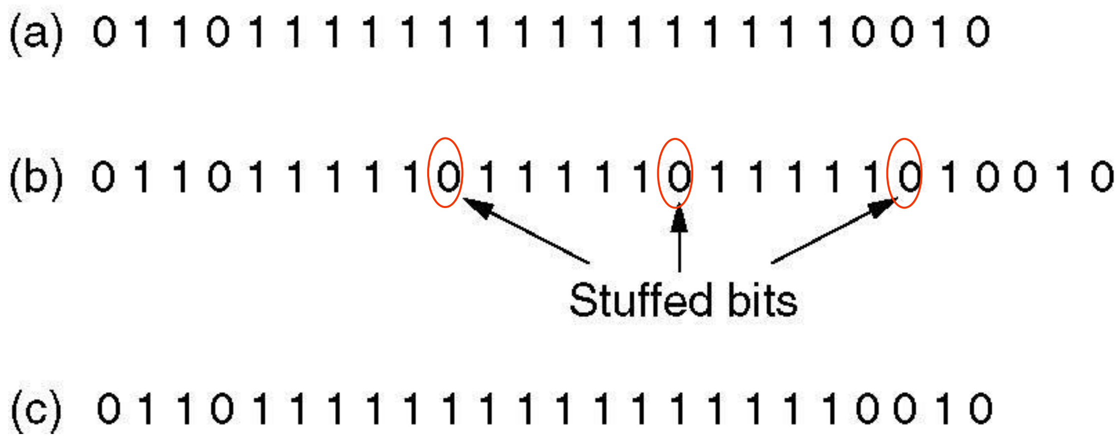
(c) Insert ESC before **each** *accidental* FLAG in data stream

(d) Insert ESC before **each** *accidental* ESC in data stream

(e) A single FLAG is a flag. Double ESC is a single ESC

What is the disadvantage of byte stuffing?

Framing Using Bit Stuffing



- Idea
  - Add (***stuff***) a bit so that the flag sequence is never transmitted inside the payload data
  - Allow the receiver to detect the "***stuffed***" bit and remove it to restore data to its original value
- Flag is 01111110
- When sender sees a sequence of 5 1's it stuffs **0** into the stream
- When receiver sees 5 1's followed by **0**, it *destuffs* the **0**
- Hence we can ***never*** get 01111110 in the data stream
- If stream contains 01111110, it is transmitted as 011111**0**10
- When receiver sees "011111**0**", it removes **0** to get "011111"

Framing by Stuffing

- Stuffing Framing Overhead:
  - Message of 100 bytes that contains only flag bytes
  - What will be the frame size to be sent?
    - **Byte Stuffing:**
    - **Bit Stuffing**:

2. Error Detection and Correction

- **Two main strategies**
  - **Error-Correcting Codes** (Forward Error Correction) FEC : enable the receiver to deduce what the transmitted data must have been
  - **Error-Detecting Codes** : allow the receiver to deduce that an error has occurred (but not which error) and have it request a retransmission

- **FEC** good for low reliability (e.g. wireless) because retransmission is frequent and may contain error itself
- **Error detection** is good for reliable links because it is cheaper to re-transmit the rare corrupted frames

- An error is a bit the value of which is reversed
  - Error sometimes is referred to as "corruption"

- Types of Errors: Isolated Single bit errors OR Bursty errors
  - Bursty errors cause less packets to be corrupted than random error
  - Much harder to correct because *many* bits are corrupted.