Week 3 –
Data Link Layer Contd

COMP90007
Internet Technologies

Reading Reminder

- Please read all of Chapter 3 for this layer…

Services Provided to Network Layer

- Principal concern is transferring data from network layer on source host to network layer on destination host
- Services provided:
  - Unacknowledged connectionless service
  - Acknowledged connectionless service
  - Acknowledged connection-oriented service

Unacknowledged Connectionless Service

- Source host transmits **independent frames** to recipient host with **no acknowledgement**
- **No logical connection establishment** or release
- **No lost frame recovery mechanism** (or left to higher levels)
- E.g. Ethernet LANs (No logical connection is established beforehand or released afterward)
- Use: Real-time traffic, e.g., voice

Acknowledged Connectionless Service

- Source host transmits independent frames to recipient host with **acknowledgement**
- No logical connection establishment or release
- Each frame individually acknowledged (*retransmission if lost or errors*)
- E.g. Wireless – IEEE 802.11 WiFi

Acknowledged Connection-Oriented Service

- Source host transmits frames to recipient host after connection establishment and with acknowledgement
- **Connection established and released**
- **Frames numbered, counted, acknowledged with logical order enforced**
- E.g., unreliable links such as satellite communications

First order of Business: *Framing*

- Physical layer provides no guarantee that a raw stream of bits is error free
- Framing is the method used by data link layer to **break raw bit stream into discrete units** and then we can generate a *checksum* for the unit
- **Checksums can be computed and embedded** at the source, then computed and compared at the destination *checksum = f(payload)*
- A key purpose therefore of framing is **to provide a unit** for running a function that gives some level of reliability over an unreliable physical layer

Framing Methods

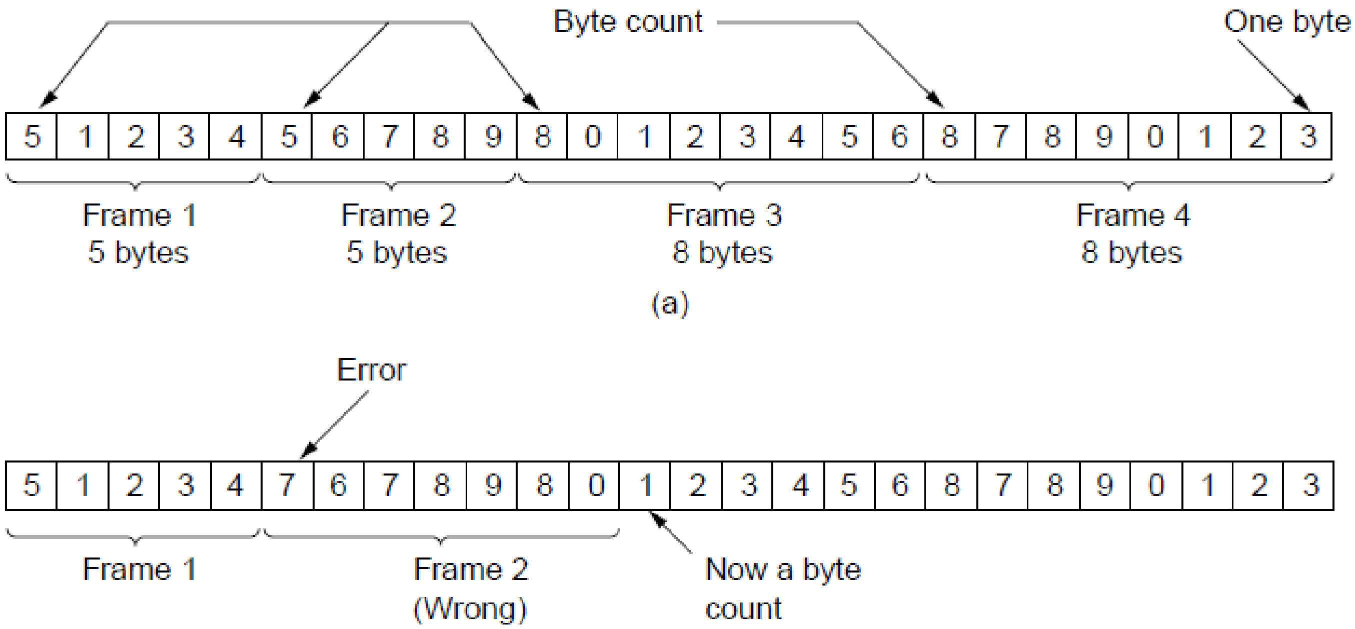
- Framing methods:
  - Character(Byte) count
  - Flag bytes with byte stuffing
  - Start and end flags with bit stuffing
- Most data link protocols use a combination of character count and one other method as the first method is inadequate by itself

Character Counts

- Uses a field in the frame header to specify the number of characters in a frame

No error

Case with error

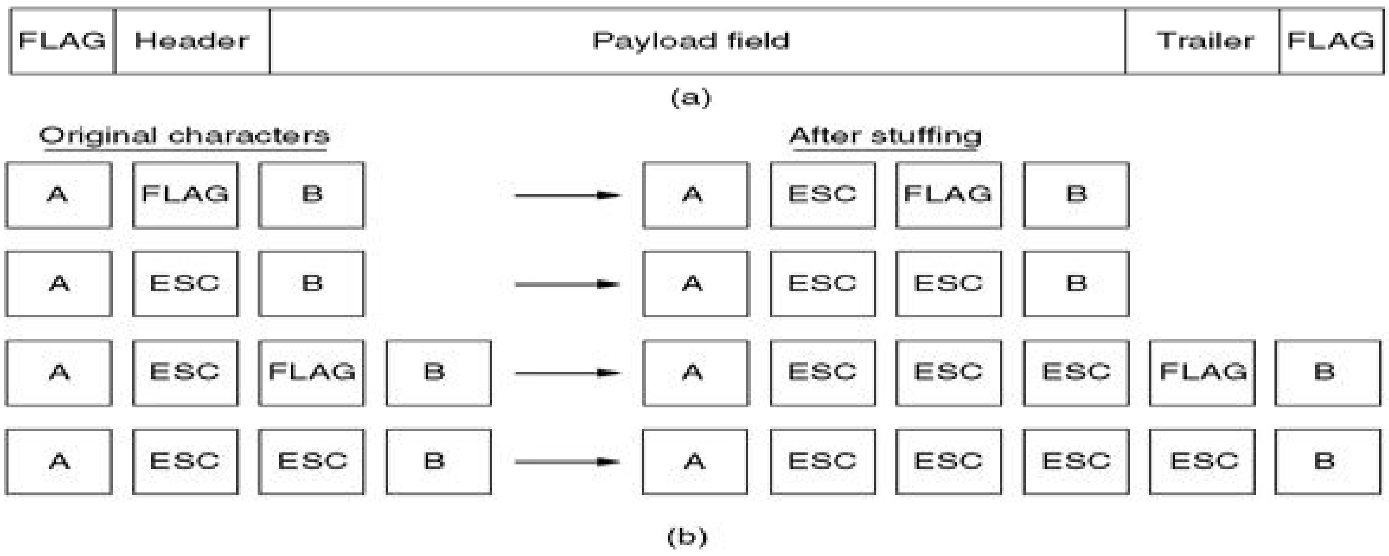


5 1 2 3 4 5 6 7 8 9 8 0 1 2 3 4 5 6 8 7 8 9 0 1 2 3

5 1 2 3 4 7 6 7 8 9 8 0 1 2 3 4 5 6 8 7 8 9 0 1 2 3

Flag Bytes with Byte Stuffing

- Each frame starts and ends with a special byte -“flag byte”



FLAG Header Payload field Trailer FLAG

(a)

Original characters After stuffing

A FLAG B → A ESC FLAG B

A ESC B → A ESC ESC B

A ESC FLAG B → A ESC ESC ESC FLAG B

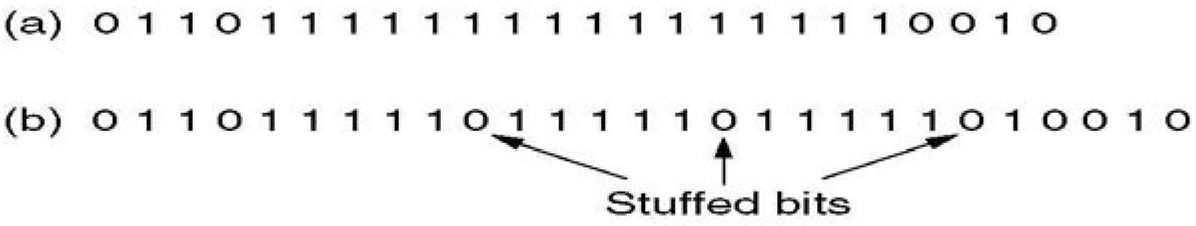
A ESC ESC B → A ESC ESC ESC ESC B

(b)

Start and End flags with Bit stuffing

- Frames contain an arbitrary number of bits
- With an arbitrary number of bits per character
- Each frame begins and ends with a special bit pattern

01111110 for example; but what happens if data has this pattern as well. Solution:



(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

Stuffed bits

(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

The original data

Sent data

Destuffing at receiver

Insert 0 after five ones (11111) basically…

Now With Error Control

- Ensuring that a garbled message by the physical layer is not considered as the original message by the receiver such as with a method that adds *check bits*
- Error Control deals with
  - **Detecting** the error
  - **Correcting** the error if we can
  - **Re-transmitting** lost frames
- Note: Link layer deals with bit errors

Error Detection&Correction Methods

- Errors may occur **randomly or in bursts**
- Bursts of errors are easier to detect but harder to resolve and some methods are good for only some cases
- Resolution needs to occur before handing data to network layer regardless
- Key goals
  - ***Fast*** mechanism and ***low computational overhead***
  - Detection of ***different kinds of error***
  - ***Minimum amount of extra bits*** send with the data

Example with a Simple Method

- Repeat the bits (if a copy is different than the other there is an error) for example:
  - 01101 -> 000 111 111 000 111
- Repeats the same bit three times in this case

- How many errors can this correct? For each bit, if one of the copies only is flipped then it can be corrected
- What is the minimum number of bit flips that can fail the algorithm? = 2
- What is the overhead? Sent data is 3 times the original size, so 2..