16.36 Communication Systems Engineering
Spring 2009

16.36: Lecture 16 - A

The Data Link Layer: Framing

Eytan Modiano

Data Link Layer (DLC)

- **Responsible for reliable transmission of packets over a link**
  - **Framing: Determine the start and end of packets**
  - **Error Detection: Determine when a packet contains errors**
  - **Error recovery: Retransmission of packets containing errors**
    - **DLC layer recovery**
    - **May be done at higher layer**

Framing

010100111010100100101010100111000100

Where is the DATA??

- **Three approaches to find frame and idle fill boundaries:**

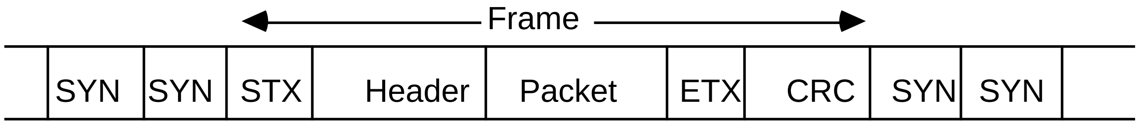
  1) Character oriented framing

  2) Length counts

  - fixed length

  3) Bit oriented protocols (flags)

Character Based Framing



Frame

SYN	SYN	STX	Header	Packet	ETX	CRC	SYN	SYN

SYN is synchronous idle
STX is start text
ETX is end text

- **Standard character codes such as ASCII and EBCDIC contain special communication characters that cannot appear in data**

- **Entire transmission is based on a character code**

Issues With Character Based Framing

- Character code dependent
  - How do you send binary data?
- Frames must be integer number of characters
- Errors in control characters are messy

NOTE: Primary Framing method from 1960 to ~1975

Length field approach (DECNET)

- Use a header field to give the length of the frame (in bits or bytes)
  - Receiver can count until the end of the frame to find the start of the next frame
  - Receiver looks at the respective length field in the next packet header to find that packet's length
- Length field must be $\log_2$ (Max_Size_Packet) + 1 bits long
  - This restricts the packet size to be used
- Issues with length counts
  - Difficult to recover from errors
  - Resynchronization is needed after an error in the length count

Fixed Length Packets (e.g., ATM)

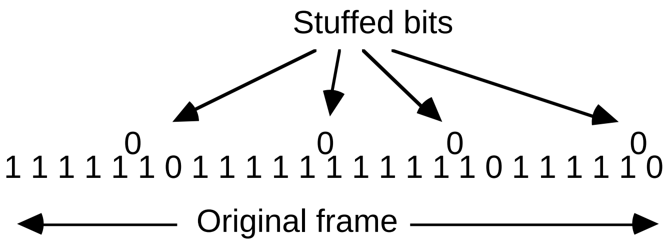
- All packets are of the same size
  - In ATM networks all packets are 53 Bytes
- Requires synchronization upon initialization
- Issues:
  - Message lengths are not multiples of packet size
    - Last packet of a message must contain idle fill (efficiency)
  - Synchronization issues
  - Fragmentation and re-assembly is complicated at high rates

Bit Oriented Framing (Flags)

- A flag is some fixed string of bits to indicate the start and end of a packet
  - A single flag can be used to indicate both the start and the end of a packet
- In principle, any string could be used, but appearance of flag must be prevented somehow in data
  - Standard protocols use the 8-bit string 01111110 as a flag
  - Use 01111111..1110 (<16 bits) as abort under error conditions
  - Constant flags or 1's is considered an idle state
- Thus 0111111 is the actual bit string that must not appear in data
- INVENTED ~ 1970 by IBM for SDLC (synchronous data link protocol)

BIT STUFFING (Transmitter)

- **Used to remove flag from original data**
- **A 0 is stuffed after each consecutive five 1's in the original frame**



- **Why is it necessary to stuff a 0 in 0111110?**
  - **If not, then**

    0111110111 -> 0111110111
    011111111 -> 0111110111

  - **How do you differentiate at the receiver?**

DESTUFFING (Receiver)

- **If 0 is preceded by 011111 in bit stream, remove it**
- **If 0 is preceded by 0111111, it is the final bit of the flag.**

Example: Bits to be removed are underlined below

$$10011111\underline{0}1100111011111\underline{0}110\underset{\text{flag}}{\underline{01111110}}$$

Overhead

- In general with a flag 01^K0 the bit stuffing is require whenever 01^{k-1} appears in the original data stream
- For a packet of length L this will happen about $L/2^k$ times

$$E\{OH\} = L/2^k + (k+2) \text{ bits}$$

- For 8 bit flag OH ~ 8 + L/64
  - For large packets efficiency ~ 1 - 1/64 = 98.5 (or 1.5% overhead)
- Optimal flag length
  - If packets are long want longer flag (less stuffing)
  - If packets are short want short flag (reduce overhead due to flag)

$$K_{opt} \sim \log_2(L)$$

Framing Errors

- All framing techniques are sensitive to errors
  - An error in a length count field causes the frame to be terminated at the wrong point (and makes it tricky to find the beginning of the next frame)
  - An error in DLE, STX, or ETX causes the same problems
  - An error in a flag, or a flag created by an error causes a frame to disappear or an extra frame to appear
- Flag approach is least sensitive to errors because a flag will eventually appear again to indicate the end of a next packet
  - Only thing that happens is that an erroneous packet was created
  - This erroneous packet can be removed through an error detection technique