Homework 1 (due 2/8/04)

CS519

In class, I described how the packet header always contains two fields, the "length" field and the "where" field. The length field is used to determine where the next packet begins. (1.1.1) Can you think of another way of delimiting the boundaries of variable length "chunks of data" that does not require a length field per se. (1.1.2) If so, what are the pros and cons of this way compared to the packet length field?

Framing

CS519

- A frame is like a packet in that it can be a variable length “chunk of data”
- Use a well-defined bit pattern to indicate “start of frame”
  - Delimiter defines frame boundary
  - Text book calls this delimiter a “sentinel”
- For instance, HDLC and PPP use “01111110” as the delimiter
  - Also sent when there is no data, so that clocks on either end may stay synchronized

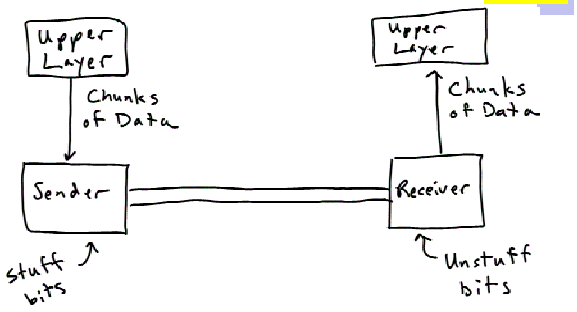
What if the delimiter is in the data???

CS519

- If the actual upper layer packet contains 01111110 it will look like a frame boundary to the receiver
- The sender must “bit stuff” (or “byte stuff”, if using a byte-oriented approach
- Similar to escape characters in strings
  - i.e. `printf(“He said \”I love 519\”.”);`
  - produces: `He said “I love 519”.`

Link framing components:

CS519



HDLC bit stuffing

- Sender:
  - If see “011111”, stuff in a 0
  - If end of frame, stuff in “01111110”
- Receiver:
  - If see “0111110…”, remove the 0
  - If see “01111110”, this is end of frame
  - If see “01111111”, this is an error

Disadvantage of framing

- The sender and receiver must examine and possibly modify every byte (or bit)!
  - Framing normally done in hardware, along with error detection
- With length field, receiver simply counts bytes and starts examining bytes at the next header

Advantage of framing

- Allows sender and receiver to stay in sync
- With headers, what happens if an error causes the receiver to lose track of where header boundaries are?
  - The receiver can never recover!
- With framing, the receiver just looks for the next delimiter