CS 640: Computer Networks

Aditya Akella

Lecture 6 -
Error/Flow Control
&
Intro to Switching
and Medium Access Control

Error Coding

- Transmission process may introduce errors into a message.
  - Single bit errors versus burst errors
- Detection: e.g. CRC
  - Requires a check that some messages are invalid
  - Hence requires extra bits
  - "redundant check bits"
- Correction
  - Forward error correction: many related code words map to the same data word
  - Detect errors and retry transmission

Parity

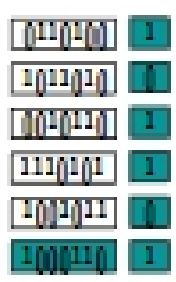
- Even parity
  - Append parity bit to 7 bits of data to make an even number of 1's
  - Odd parity accordingly defined.

- 1 in 8 bits of overhead?
  - When is this a problem?
- Can detect a single error

- But nothing beyond that

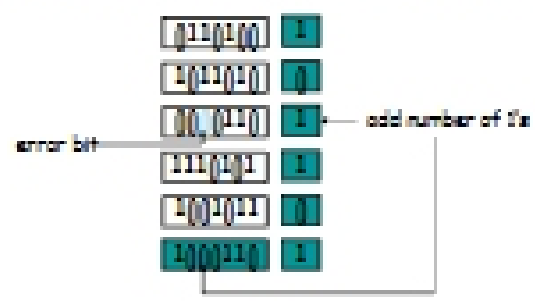
2-D Parity

- Make each byte even parity
- Finally, a parity byte for all bytes of the packet
- Example: five 7-bit character packet, even parity



Effectiveness of 2-D Parity

- 1-bit errors can be detected, corrected
- Example with even parity per byte:



Effectiveness of 2-D Parity

- 2-bit errors can also be detected
- Example:

- What about 3-bit errors? >3-bit errors?

Cyclic Redundancy Codes (CRC)

- Commonly used codes that have good error detection properties
  - Can catch many error combinations with a small number of redundant bits
- Based on division of polynomials
  - Errors can be viewed as adding terms to the polynomial
  - Should be unlikely that the division will still work
- Can be implemented very efficiently in hardware
- Examples:
  - CRC-32: Ethernet
  - CRC-8, CRC-10, CRC-32: ATM

Link Flow Control and Error Control

- Dealing with receiver overflow: flow control.
- Dealing with packet loss and corruption: error control.
- Actually these issues are relevant at many layers.
  - Link layer: sender and receiver attached to the same "wire"
  - End-to-end: transmission control protocol (TCP) – sender and receiver are the end points of a connection
- How can we implement flow control?
  - "You may send" (windows, stop-and-wait, etc.)
  - "Please shut up" (source quench, 802.3x pause frames, etc.)

Flow Control: A Naïve Protocol

- Sender simply sends to the receiver whenever it has packets.
- Potential problem: sender can outrun the receiver.
  - Receiver too slow, small buffer overflow, ..
- Not always a problem: receiver might be fast enough.