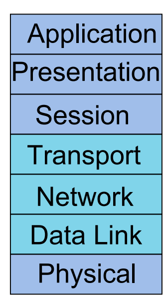
CSE 461: Error Detection and Correction

Next Topic

- Error detection and correction
- Focus: How do we detect and correct messages that are garbled during transmission?
- The responsibility for doing this cuts across the different layers



Errors and Redundancy

- Noise can flip some of the bits we receive
  - We must be able to detect when this occurs!
  - Who needs to detect it? (links/routers, OSs, or apps?)
- Basic approach: add redundant data
  - Error detection codes allow errors to be recognized
  - Error correction codes allow errors to be repaired too

Motivating Example

- A simple error detection scheme:
  - Just send two copies. Differences imply errors.
- Question: Can we do any better?
  - With less overhead
  - Catch more kinds of errors
- Answer: Yes – stronger protection with fewer bits
  - But we can't catch all inadvertent errors, nor malicious ones
- We will look at basic block codes
  - K bits in, N bits out is a (N,K) code
  - Simple, memoryless mapping

Detection vs. Correction

- Two strategies to correct errors:
  - Detect and retransmit, or Automatic Repeat reQuest. (ARQ)
  - Error correcting codes, or Forward Error Correction (FEC)
- Retransmissions typically at higher levels (Network+). Why?

- Question: Which should we choose?

Retransmissions vs. FEC

- The better option depends on the kind of errors and the cost of recovery
- Example: Message with 1000 bits, Prob(bit error) 0.001
  - Case 1: random errors
  - Case 2: bursts of 1000 errors
  - Case 3: real-time application (teleconference)

The Hamming Distance

- Errors must not turn one valid codeword into another valid codeword, or we cannot detect/correct them.
- Hamming distance of a code is the smallest number of bit differences that turn any one codeword into another
  - e.g, code 000 for 0, 111 for 1, Hamming distance is 3
- For code with distance d+1:
  - d errors can be detected, e.g, 001, 010, 110, 101, 011
- For code with distance 2d+1:
  - d errors can be corrected, e.g., 001 → 000

Parity

- Start with n bits and add another so that the total number of 1s is even (even parity)
  - e.g. 0110010 → 01100101
  - Easy to compute as XOR of all input bits
- Will detect an odd number of bit errors
  - But not an even number
- Does not correct any errors

2D Parity

- Add parity row/column to array of bits
- How many simultaneous bit errors can it detect?
- Which errors can it correct?

0101001	1
1101001	0
1011110	1
0001110	1
0110100	1
1011111	0
1111011	0

Checksums

- Used in Internet protocols (IP, ICMP, TCP, UDP)
- Basic Idea: Add up the data and send it along with sum
- Algorithm:
  - checksum is the 1s complement of the 1s complement sum of the data interpreted 16 bits at a time (for 16-bit TCP/UDP checksum)
- 1s complement: flip all bits to make number negative
  - Consequence: adding requires carryout to be added back

CRCs (Cyclic Redundancy Check)

- Stronger protection than checksums
  - Used widely in practice, e.g., Ethernet CRC-32
  - Implemented in hardware (XORs and shifts)
- Algorithm: Given n bits of data, generate a k bit check sequence that gives a combined n + k bits that are divisible by a chosen divisor C(x)
- Based on mathematics of finite fields
  - "numbers" correspond to polynomials, use modulo arithmetic
  - e.g, interpret 10011010 as $x^7 + x^4 + x^3 + x^1$

How is C(x) Chosen?

- Mathematical properties:
  - All 1-bit errors if non-zero x^k and x^0 terms
  - All 2-bit errors if C(x) has a factor with at least three terms
  - Any odd number of errors if C(x) has (x + 1) as a factor
  - Any burst error < k bits
- There are standardized polynomials of different degree that are known to catch many errors
  - Ethernet CRC-32: 100000100110000010001110110110111

Reed-Solomon / BCH Codes

- Developed to protect data on magnetic disks
- Used for CDs and cable modems too
- Property: 2t redundant bits can correct <= t errors
- Mathematics somewhat more involved …

Key Concepts

- Redundant bits are added to messages to protect against transmission errors.
- Two recovery strategies are retransmissions (ARQ) and error correcting codes (FEC)
- The Hamming distance tells us how much error can safely be tolerated.