Misc.

- Shannon vs. Nyquist
- Class participation grade
Data Link Layer

• Unacknowledged connectionless service
• Acknowledged connectionless service
• Acknowledged connection-oriented service
Framing

- Character Count
- Starting and ending character, with stuffing
- Starting and ending flags with bit stuffing
- Physical Layer coding violations
Error correcting codes

• N bits form a codeword
  • XOR two codewords for Hamming distance
  • Hamming distance equals number of bit errors
    • Required to turn one code into another

• Complete Code
  • Not all codes are possible
    • Find closest Hamming distance

• Detecting n errors, you need n+1 Hamming distance
• Correcting n errors, requires 2n+1 Hamming distance
  • Original code must be closest
Error detecting code example

- Parity bit
  - Single parity bit
    - Hamming distance of 2
    - Can detect single errors
      - Can not correct any errors
4 word code with 10 bits

- Hamming distance of 5
  - Can detect up to 4 bits of error
  - Can only correct up to 2 bits
Better Error detection

- Easier to detect errors and resend
- Matrix
  - Build matrix from block of data
    - Parity bit for every column
    - Check all bits on receipt
  - For matrix N bits wide
    - Can detect static burst of N
    - Static burst N+1 not detected
    - Probability of detection is $2^{-N}$
Polynomial

- Cyclic redundancy code (CRC)
  - $k$ bits represented as coefficients for $x^{k-1}$ polynomial
  - Choose generator polynomial
    - Leading and ending coefficients are 1
    - Divide message poly by generator
    - Remainder is checksum
  - Error will be detected unless divisible by $G(x)$
Base 2 arithmetic

- No odd poly is divisible by $x+1$
  - Substitute 1, $E(x)$ will always be 1
  - Catch all odd bit errors with $x+1$
Common Generator Polynomials

- CRC-12: \( x^{12} + x^{11} + x^{3} + x^{2} + x^{1} + 1 \)
  - 6 bit characters
- CRC-16 = \( x^{16} + x^{15} + x^{2} + 1 \)
- CRC-CCITT = \( x^{16} + x^{12} + x^{5} + 1 \)
  - Both used for 8-bit chars
  - Usually computed in hardware
16 Bit CRC calculation

• Will catch the following errors
  • All single and double errors
  • All errors with an odd number of bits
  • All burst errors of length equal or less than 16
  • 99.997% of 17 bit errors
  • 99.998% of 18 bit and longer bursts
Protocol 1

- sender()
  - while (true) {
    - from_net_layer(&buf);
    - sbuf = buf;
    - to_phys_layer(&sbuf); }

- Receiver()
  - while (true) {
    - wait_for_event(&event)
    - from_physical_layer(&rbuf)
    - to_network_layer(&rbuf) }
Protocol 2

- **sender()**
  - while (true) {
    - from_net_layer(&buf);
    - sbuf = buf;
    - to_phys_layer(&sbuf);
    - wait_for_event(&event);
  }

- **Receiver()**
  - while (true) {
    - wait_for_event(&event);
    - from_physical_layer(&rbuf);
    - to_network_layer(&rbuf);
    - to_physical_layer(&sbuf); }

- from_net_layer
- to_net_layer
- from_phys_layer
- to_phys_layer
- wait_for_event
Protocol 3

- sender3()
  - while (true) {
    - s.info = buffer; /*construct frame to send
    - s.seq = next_frame;
    - to_physical_layer(&s);
    - start_timer(s.seq);
    - wait_for_event(&event);
    - if (event == frame_arrival) {
      - from_physical_layer(&s);
      - if (s.ack == next_frame_to_send) {
        - from_network_layer(&buffer)
        - inc(next_frame_to_send); } } } /*max=1
Protocol 3 (cont.)

• receiver3()
  • while (true) {
    • wait_for_event(&event);
    • if (r.seq==frame_arrival) {
        • from_physical(&r);
        • if (r.seq == frame_expected) {
            • to_network(&r.info);
            • inc(frame_expected); } } 
    • s.ack = 1 - frame_expected;
    • to_physical_layer(&s) }
  • } }
Protocol 4

- Sliding Window, Bi-Directional
  - 1-bit
  - A times out too fast
    - A sends repeated frames, B keeps sending
      seq=ack=0
    - Simultaneous sending
      - (seq, ack, packet #)
Protocol 5

- Pipelining
  - Transmit up to Max_SEQ
  - Allows multiple outstanding frames
    - Retransmits everything after error
      - Does poorly with a lot of errors
Protocol 6

- Pipeline, non-sequential errors
  - Transmit up to Max_SEQ
  - Allows multiple outstanding frames
    - Selective Retransmit
      - NAK--negative acknowledgement
Protocol Specifications

• Finite State Machine Models
  • Protocol machine is a specific state at any one time
    • For software, means all possible variable values
  • Example: Protocol 3
    • Positive Ack with Retransmission
    • State triplet:({0,1}.{0,1},{0,1,A,-})
      • Transmit packet number
      • Expected Receive packet number
    • Channel state
    • One cycle is 4 states+error conditions
Verification

- With 1 bit sequence numbers
  - Receiver should not deliver 2 odd or even packets
    - No transition double 1 transition
      - Without a 3 transition
  - Deadlock
    - There should exist no subset of states such that
      - No transition out of the subset
      - No transition in the subset that causes progress
Sample Protocols

- SLIP
  - IP packets with flag at beginning and end
    - 0xC0, quote with 0xDB
- PPP
Midterm

- Second half of class
- Questions
  - Tanenbaum Chapter 1-3
    - Questions like the homework
      - Review homework
      - Review other questions in Tanenbaum
  - Systems Admin
    - Lecture Notes
Midterm

- Programming question
  - Real World Example
    - Write finite state diagram
    - Write Pseudo code
      - Think high level!
      - Write high level steps
      - Ignore details until the end.
    - Land and Hold Short Example