Outline

- Error detection
- Reliable transmission

Error Detection

Cyclic Redundancy Check

- Add k bits of redundant data to an n-bit message
  - want k << n
  - e.g., k = 32 and n = 12,000 (1500 bytes)
- Represent n-bit message as n-1 degree polynomial
  - e.g., MSG=10011010 as $M(x) = x^7 + x^4 + x^3 + x^1$
- Let k be the degree of some divisor polynomial
  - e.g., $C(x) = x^3 + x^2 + 1$

CRC (cont)

- Transmit polynomial P(x) that is evenly divisible by C(x)
  - shift left k bits, i.e., $M(x).x^k$
  - subtract remainder of $M(x).x^k/C(x)$ from $M(x).x^k$
- Receiver polynomial P(x) + E(x)
  - E(x) = 0 implies no errors
- Divide (P(x) + E(x)) by C(x); remainder zero if:
  - E(x) was zero (no error), or
  - E(x) is exactly divisible by C(x)

Selecting C(x)

- All single-bit errors, as long as the x^k and x^0 terms have non-zero coefficients.
- All double-bit errors, as long as C(x) contains a factor with at least three terms
- Any odd number of errors, as long as C(x) contains the factor (x + 1)
- Any 'burst' error (i.e., sequence of consecutive error bits) for which the length of the burst is less than k bits.
- Most burst errors of larger than k bits can also be detected
- See Table 2.6 on page 102 for common C(x)

Internet Checksum Algorithm

- View message as a sequence of 16-bit integers; sum using 16-bit ones-complement arithmetic; take ones-complement of the result.

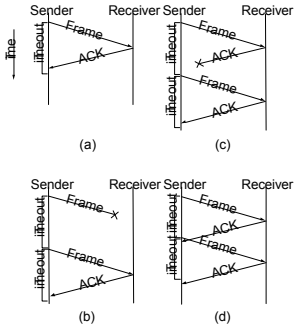
```
u_short cksum(u_short *buf, int count) {
   register u_long sum = 0;
   while (count--){
      sum += *buf++;
      if (sum & 0xFFFF0000){
         /* carry occurred, so wrap around */
         sum &= 0xFFFF;
         sum++;
      }
   }
   return ~(sum & 0xFFFF);
}
```

Reliable Transmission

Acknowledgements and Timeouts

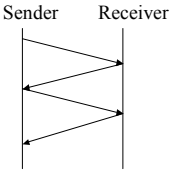
- ACK – Control frame sent back to peers saying that it successfully received an earlier frame
- If no ack without reasonable time, timeout and retransmit packet
- Automatic Repeat Request - ARQ

Acknowledgements & Timeouts



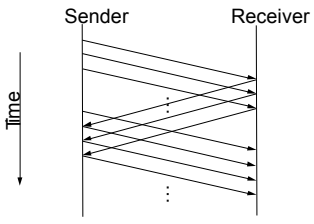
Stop-and-Wait

- Problem: keeping the pipe full
  - (bandwidth delay product)
- Example
  - 1.5Mbps link x 45ms RTT = 67.5Kb (8KB)
  - 1 KB frames imples 1/8th link utilization



Sliding Window

- Allow multiple outstanding (un-ACKed) frames
- Upper bound on un-ACKed frames, called window



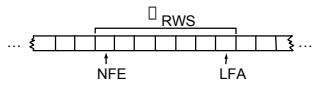
SW: Sender

- Assign sequence number to each frame (SeqNum)
- Maintain three state variables:
  - send window size (SWS)
  - last acknowledgment received (LAR)
  - last frame sent (LFS)
- Maintain invariant: LFS - LAR <= SWS

- Advance LAR when ACK arrives
- Buffer up to SWS frames

SW: Receiver

- Maintain three state variables
  - receive window size (RWS)
  - largest frame acceptable (LFA)
  - last frame received (NFE)
- Maintain invariant: LFA - LFR <= RWS



- Frame SeqNum arrives:
  - if LFR < SeqNum < = LFA → accept
  - if SeqNum < = LFR or SeqNum > LFA → discarded
- Send cumulative ACKs

Sequence Number Space

- SeqNum field is finite; sequence numbers wrap around
- Sequence number space must be larger then number of outstanding frames
- SWS <= MaxSeqNum-1 is not sufficient
  - suppose 3-bit SeqNum field (0..7)
  - SWS=RWS=7
  - sender transmit frames 0..6
  - arrive successfully, but ACKs lost
  - sender retransmits 0..6
  - receiver expecting 7, 0..5, but receives second incarnation of 0..5
- SWS < (MaxSeqNum+1)/2 is correct rule
- Intuitively, SeqNum "slides" between two halves of sequence number space

Concurrent Logical Channels

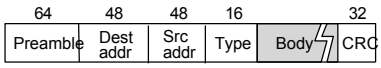
- Multiplex 8 logical channels over a single link
- Run stop-and-wait on each logical channel
- Maintain three state bits per channel
  - channel busy
  - current sequence number out
  - next sequence number in
- Header: 3-bit channel num, 1-bit sequence num
  - 4-bits total
  - same as sliding window protocol
- Separates reliability from order

Shared Access Networks

Bus (Ethernet)
Token ring (FDDI)
Wireless (802.11)

Ethernet Overview

- History
  - developed by Xerox PARC in mid-1970s
  - roots in Aloha packet-radio network
  - standardized by Xerox, DEC, and Intel in 1978
  - similar to IEEE 802.3 standard
- CSMA/CD
  - carrier sense
  - multiple access
  - collision detection
- Frame Format



Ethernet (cont)

- Addresses
  - unique, 48-bit unicast address assigned to each adapter
  - example: 8:0:e4:b1:2
  - broadcast: all 1s
  - multicast: first bit is 1
- Bandwidth: 10Mbps, 100Mbps, 1Gbps
- Length: 2500m (500m segments with 4 repeaters)
- Problem: Distributed algorithm that provides fair access

Transmit Algorithm

- If line is idle…
  - send immediately
  - upper bound message size of 1500 bytes
  - must wait 9.6us between back-to-back frames
- If line is busy…
  - wait until idle and transmit immediately
  - called 1-persistent (special case of p-persistent)

Algorithm (cont)

- If collision…
  - jam for 32 bits, then stop transmitting frame
  - minimum frame is 64 bytes (header + 46 bytes of data)
  - delay and try again
    - 1st time: 0 or 51.2us
    - 2nd time: 0, 51.2, or 102.4us
    - 3rd time51.2, 102.4, or 153.6us
    - nth time: k x 51.2us, for randomly selected k=0..2n - 1
    - give up after several tries (usually 16)
    - exponential backoff

Collisions