Announcements

- Homework #3 is
- Reading
  - Today: 3.5- 3.6

Error Detection

- Less bits are required
  - if errors are infrequent, then then this works better
  - assumes that re-transmission is possible
- Cyclic Redundancy Codes (CRC)
  - Use a generator function G(x) of degree r
  - let M' be the message with r 0's on the end of it
  - divide M' into G(x) and compute remainder
    - use this as the r bit CRC code
  - a code with r bits will detect all burst errors less than r bits
  - several G's are standardized
    - CRC-12 = $x^{12} + x^{11} + x^3 + x^2 + x + 1$
    - CRC-16 = $x^{16} + x^{15} + x^2 + 1$
    - CRC-CCITT = $x^{16} + x^{12} + x^5 + 1$
  - 16 bit CRC will catch
    - all single and double bit errors
    - all errors with an odd number of bits
    - all burst errors of length less than 16

CRC Example

Frame : 1 1 0 1 0 1 1 0 1 1
Generator: 1 0 0 1 1
Message after appending 4 zero bits: 1 1 0 1 0 1 1 0 0 0 0

```
                   1 1 0 0 0 0 1 0 1 0
         1 0 0 1 1 ┌────────────────────────────
                   │ 1 1 0 1 0 1 1 0 1 1 0 0 0 0
                     1 0 0 1 1
                     ─────────
                       1 0 0 1 1
                       1 0 0 1 1
                       ─────────
                         0 0 0 0 1
                         0 0 0 0 0
                         ─────────
                           0 0 0 1 0
                           0 0 0 0 0
                           ─────────
                             0 0 1 0 1
                             0 0 0 0 0
                             ─────────
                               0 1 0 1 1
                               0 0 0 0 0
                               ─────────
                                 1 0 1 1 0
                                 1 0 0 1 1
                                 ─────────
                                   0 1 0 1 0
                                   0 0 0 0 0
                                   ─────────
                                     1 0 1 0 0
                                     1 0 0 1 1
                                     ─────────
                                       0 1 1 1 0
                                       0 0 0 0 0
                                       ─────────
                                         1 1 1 0   Remainder
```

Transmitted frame: 1 1 0 1 0 1 1 0 1 1 1 1 1 0

PPP Protocol

- Link Protocol for Serial Lines
  - Supports multiple network protocols: IP, IPX, CLNP, ...
  - designed for dialup or leased lines
- Link Establishment
  - configure-request: list of proposed options and values
  - configure-{ack/nack}: will (won't) use the requested option
  - NCP protocol
    - per network level protocol
    - used to establish network attributes (e.g. addresses)

1	1	1	1 or 2	variable	2 or 4	1
flag 01111110	Address 11111111	control 00001110	protocol	payload	checksum	flag 01111110

From: *Computer Networks*, 3rd Ed. by Andrew S. Tanenbaum, (c)1996 Prentice Hall.

ATM Datalink Protocol

- Header
  - use CRC over the 32 bits of the header
- How to find cell boundary?
  - use shifty register to check for valid checksum
    - 1/256 chance of a random match
  - use HUNT mode to increase chances
    - after a good cell, skip to the next cell boundary
    - must receive δ cells with checksum matches
- Detecting loss of synchronization
  - one bad cell is probably an error
  - many bad cells is likely a slip (loss of sync)
  - if α bad cells are seen in a row, switch to hunt mode

Medium Access Layer

- Broadcast Networks
  - share a common resource for communication
    - bus, wire, air, etc.
  - need to coordination access to this resource
- Limits of Static Channel Allocation
  - suitable for constant rate traffic of similar speeds
  - however, bursty traffic results in poor channel utilization
  - consider one queue vs. separate queues for each person
    - n queues with bursty arrival have mean delay n times 1 queue
- Dynamic Allocation
  - only use channel when have something to send
  - need to control access to the channel

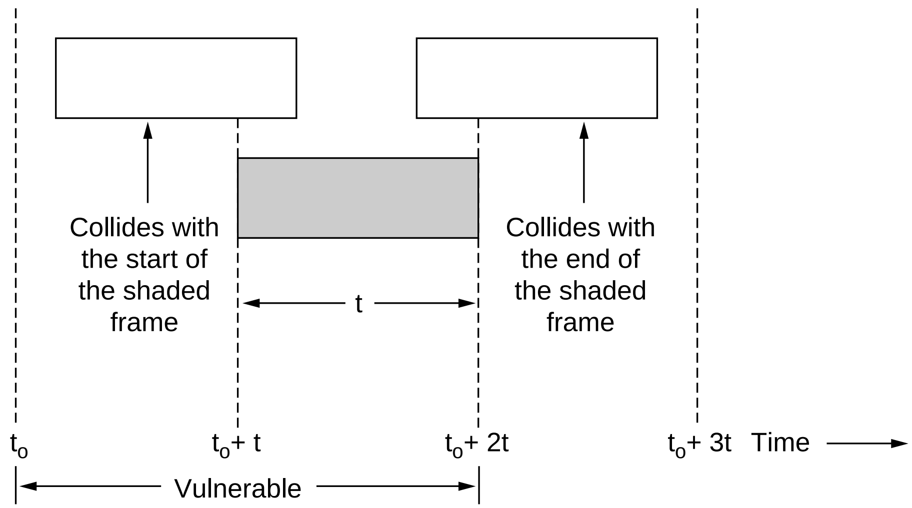
Shared Channel Model

- Station model
  - N independent stations
  - each wants to send λ frames per second
  - a station may not send another frame until the first is sent
- Single Channel Assumption
  - all stations communicate over a single shared channel
- Collisions: two stations attempt to send at once
  - neither transmission succeeds
- Time
  - continuous time: frame transmissions can start anytime
  - discrete time: clock ensures all sends initiate at the start of a slot
- Carrier Sense
  - stations can tell if channel is in use before sending
  - stations must wait to know if channel was in use

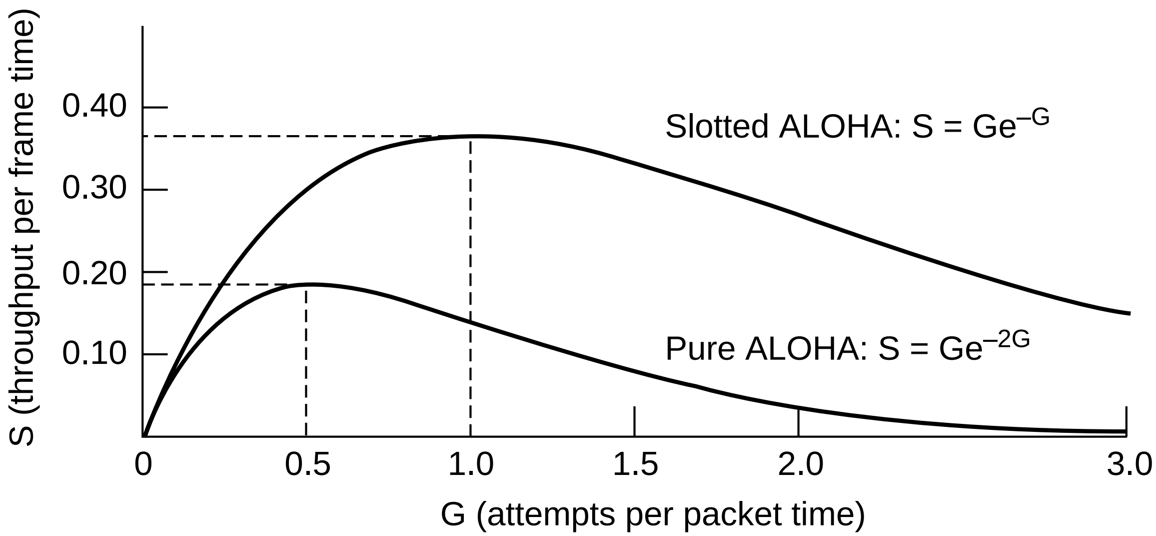
Aloha

- Stations
  - ground based radio stations on islands
- Pure Aloha
  - send data a will, collisions will happen
  - on collision, wait a random amount of time & try again
  - use standard, fixed size packets
  - what is channel efficiency?
    - assume S **new** frames per frame time
    - assume G total frames trying to be sent per frame time
    - $S = G P_0$
    - probability of k frames generated during a frame time
      - $\Pr[k] = G^k e^{-G} / k!$
    - $P_o = e^{-2G}$, so $S = Ge^{-2G}$

Performance of Aloha



From: *Computer Networks*, 3rd Ed. by Andrew S. Tanenbaum, (c)1996 Prentice Hall.



From: *Computer Networks*, 3rd Ed. by Andrew S. Tanenbaum, (c)1996 Prentice Hall.

Aloha (cont.)

- Slotted Aloha
  - Use a central clock
  - Each station only sends at the start of frame
  - Reduces collision window by 1/2
    - $S = G e^{-G}$

Carrier Sense Multiple Access

- look before you leap!
  - don't send if someone else is sending
- collisions are still possible
  - propagation delay induces uncertainty into sensing
  - possible two hosts both start sending at the same time
- persistence: when to send after detecing channel in use
  - 1-persistent
    - as soon as the channel is free, starting sending
  - nonpersistent CSMA
    - if channel is sensed busy, wait a random time and try again
  - p-persistent CSMA
    - if slot is idle send with probability p, else wait for next idle slot