ICS 451: Today's plan

- Transport Layer
- Reliable Transfer
- Headers and Control Packets
- Alternating Bit Protocol
- Sliding Window Reliable Transmission

Transport Layer

- The transport layer provides end-to-end services to the application layer:
  - reliable transmission
  - of any size data
  - application (de)multiplexing
- The transport layer uses the end-to-end services of the network layer:
  - unreliable: packet loss, duplication, reordering, delay, corruption
  - maximum transmission unit (MTU)

Reliable Transmission

- reliable transmission must detect and correct:
  - packet corruption (bits flipped or lost)
  - packet loss
  - arbitrary packet delay
  - packet reordering and duplication

- How can you detect packet corruption?

Packet corruption

- packet corruption can be detected by adding additional bits that say something about the packet contents
- e.g. a parity bit:
  - if packet has even number of 1's, add 0
  - if packet has odd number of 1's, add 1
- parity detects any single bit error
  - but may fail to detect multiple bit errors

Checksum

- packet corruption can be detected by adding additional bits that say something about the packet contents
- e.g. an additional byte that is the sum of all bytes modulo 256
  - if any one byte changes, we can detect it.
  - if any two bytes change, we can detect it with probability 255/256
- example: packet 0x11 0x22 0x33
  - with checksum: 0x99 0x11 0x22 0x33
  - sum is 0xff if the packet is correct

CRC

- packet corruption can be detected by adding additional bits that say something about the packet contents
- e.g. the remainder when the packet is divided by a constant
  - if carefully designed, can detect and correct multiple errors
- designed to be implemented in hardware
  - part of Ethernet, WiFi
  - slower in software, so Internet uses checksum

Packet Loss

- packet loss can be due to corruption
  - if an incorrect packet is received, it is discarded
- packet loss can be due to buffers being full
  - if a packet is received, but there is no room in memory for it, the packet is discarded
- packet loss can be due to incorrect routing and other causes

Recovering from Packet Loss

- Transport layer recovers from packet loss by:
  - retransmitting packets
- The sender must know when to retransmit
  - receiver sends sender a special packet, an **ack**, to confirm that a data packet was received
  - so there are *data packets* and *control packets*
- The receiver must be able to distinguish retransmitted (and duplicate) packets from new packets

Packet Headers

- A single bit might distinguish a control packet from a data packet
- There is often other information to send with the data, e.g. port numbers
- This data is collected at the front of the packet, in a **header**
- The header often has a fixed format, as seen in DNS

Reliable Transmission: Alternating Bit Protocol

- Sender sends packet with sequence number 0
- Receiver sends ack(0), or nothing
  - if sender gets ack(0), sends packet(1)
  - if sender times out, resends packet(0)
- after getting ack(1), sender sends the third packet, again with sequence number 0

- Simplest protocol using sequence numbers
  - only works if network delivers packets in order

Alternating Bit Protocol: out-of-order delivery

- Sender sends packet 1 with sequence 0
- Sender times out and retransmits
- Receiver acks sequence 0
- Sender sends packet 2 with sequence 1
- Receiver acks sequence 1
- Sender sends packet 3 with sequence 0
- Network delivers original packet 1 with seq 0
- Receiver cannot distinguish 1 from 3!

Multiple-bit Sequence Numbers

- Receiver cannot distinguish 1 from 3 because they have the same sequence number
  - solution: have more than 2 sequence numbers
  - e.g. 2^n sequence numbers
  - and specify a maximum segment lifetime (MSL)
    - e.g. 2 minutes on the Internet
- then, as long as we use less than $2^n/2$ (2^{n-1}) segments/MSL, there is no danger of confusion

Alternating Bit Protocol: throughput tied to latency

- with the ABP, at most one packet can be sent at a time
  - then, the sender must wait for the ack
- if Round-Trip Time (RTT) is *s* seconds, and MTU is *B* bytes, maximum speed is *B*/*s* bytes/second
  - even on very high-speed networks
- If we can send multiple segments at once, we might get closer to the network throughput

Sliding Window

- With n-bit sequence numbers, 2^{n-1} segments could be in transit at any given time
- The limit becomes the amount of buffer space at the receiver: sender should not send more than receiver can buffer
- This buffer size is called the **send window**
- When the receiver acks one more segment, sender can send one more segment
  - as long as the send window size is the same
  - the window is *sliding* to greater seq numbers

Types of Acknowledgement

- If packets are delivered in order, the receiver only acks (sends the sequence number of) the last received packet to acknowledge all preceding packets
  - this is a **cumulative acknowledgement**
- A receiver might get packets out of order, and save them to see if the earlier packets arrive
  - or are retransmitted

Then the receiver must ack individual packets

  - this is a **selective acknowledgement**

Acks and the Internet

- Every ack must carry a sequence number
  - of the packet it is acking
  - n bits long
- On the Internet, 32-bit sequence numbers
  - sequence numbers count bytes, not packets
- TCP acks are cumulative
  - TCP options also support selective acks
  - TCP options also support extensions to the 32-bit sequence numbers

Ack piggybacking

- TCP connections are bidirectional
  - data (as well as control) may flow in both directions
- so every TCP header carries a sequence number and an ack number
  - except the first TCP segment in a connection
- When data is carried, the ack is said to be *piggybacked* on the data
- A packet with no data is called an ack

Bandwidth-Delay Product

- Suppose a send window is *B* bytes
- and RTT is *s* seconds
- Then the sender can send at most one window per RTT, or *B/s* bytes per second
- If the window is 10,000B, and the RTT is 100ms, what is the maximum throughput?

  give the answer in bits/second
- Buying a faster network won't help!!!