ICS 451: Today's plan

- Sliding Window Reliable Transmission
- Acknowledgements
- Windows and Bandwidth-Delay Product
- Retransmission
- Timers
- Connections

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!!!
- Need window ≥ bandwidth x delay product

Retransmission Strategies I

- Timeout must be > RTT
- Receiver discards packets received out of order
- On timeout, sender retransmits all the unacked segments

Retransmission Strategies II

- Timeout must be > RTT
- Receiver keeps packets received out of order
  - that are in the window
- On timeout, sender retransmits the oldest unacked segment

Retransmission Strategies III

- Timeout must be > RTT
- Receiver keeps packets received out of order
    - that are in the window
- When getting out-of-order packets, receiver sends selective acknowledgement
    - if packets are in order, sends cumulative ack
- Sender retransmits:
    - on selective ack, requested segment
    - on timeout, oldest unacked segment

Window Management

- Receiver must tell sender its window
- Control packets needed to carry acks
- Control packets also carry window information!
- Receiver may change window at any time
  - but preferably not discard any already-sent data

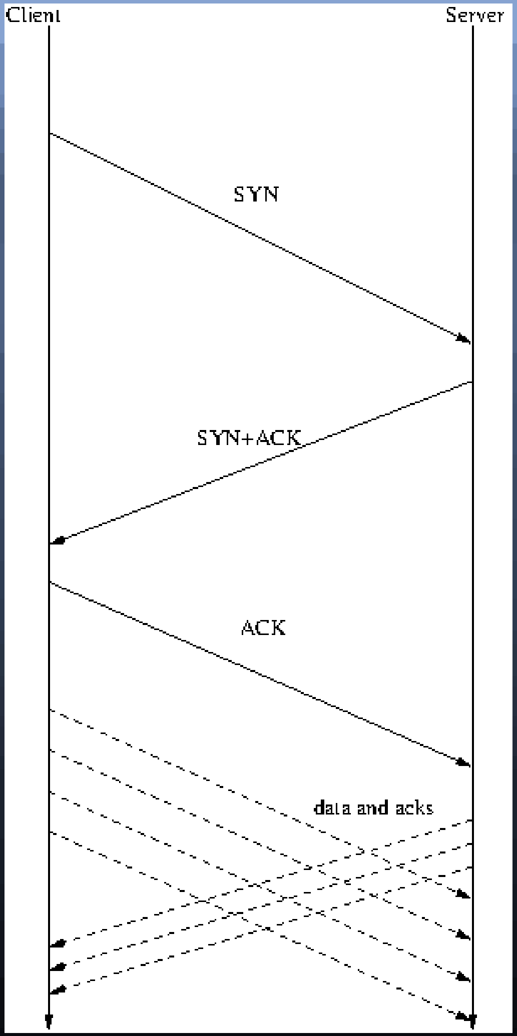
TCP adaptive timer

- TCP sends segments, waits for acks/timeouts
- On every ack, TCP knows how much time has passed since the segment was sent
  - except for retransmitted segments
- so TCP keeps track of the Round-Trip Time
- Timeout set to avg RTT+ 4(variance in RTT)
- At start, timeout is set to 3seconds

TCP Connection Establishment

- The initial sequence number and window size on a TCP connection are not known to the peer
- So TCP sends a special control packet called a SYNchronization (SYN) packet
- A SYN is sent in each direction:
  - from the client to the server
  - then from the server to the client
- Each SYN is ACK'd
- The first ACK is piggybacked with the 2nd SYN

TCP Connection Establishment: 3-way handshake



Three-Way Handshake

- Initial Sequence numbers (ISNs) should be unpredictable
  - and non-repeating
- Each SYN packet must be acked
  - ack carries ISN+1
- SYN packets must be retransmitted on timeout
- At the end, both sides agree to have a connection, and on its window sizes

Three-Way Handshake: Connection Refused

- If the server does not have a socket `listen`ing on the desired port, it resets the connection
- It sends an RST packet to the client
  - same as a SYN, but setting a different bit in the header
- RST can be sent any time during a connection
  - e.g. after rebooting

Three-Way Handshake: Retransmission Scenarios

- Study *and understand* the retransmission scenarios in Section “Connection establishment and release” of the textbook, pp 84-86

Closing a connection

- Each side sends a FIN packet
  - each FIN packet must be acknowledged
- If Alice sends FIN to Bob, she is promising never to send any more data on the connection
  - but Bob can keep sending
- That's why `shutdown` allows “half-closed” connections