Computer Networks
ICS 651

- Exam Review:
  - transport layer
  - TCP basics
  - congestion control
  - UDP
  - P2P technology
  - project 2

transport layer

- provides at least the demultiplexing function: between two IP hosts there can be many connections (socket pairs), identified by pairs of port numbers
- mostly TCP for stream- and connection-oriented reliable transmission, UDP for everything else
- TCP provides additional functions, particularly flow control and congestion control
- also SCTP, RTP

TCP basics

- reliable byte stream
- requires sequence and acknowledgment numbers
- TCP is based on **cumulative** acks, but has a Selective Ack option
- also requires state on the endpoints to keep track of sequence numbers and buffers
- state allocation and deallocation is explicit in TCP, letting the application figure out when to open and close connections
- explicit management of each receiver buffer: the TCP **window**

TCP details

- each byte (and the SYN and FIN bits) has its own sequence number
- the sequence number in the packet is the sequence number of the first data byte (or SYN/FIN) in the packet
- the corresponding ACK adds the sequence number and the number of bytes (+SYN/FIN) in the packet
- for sender, left edge of window is ack number received, right edge is ack+window-1
- TCP adaptive timer
- Karn algorithm (do not use retransmitted segments in RTT estimation), Nagle algorithm (only send full segments, or send when everything is acked, or send after a timeout)
- delayed acks
- TCP checksum, pseudo-header
- TCP header, including congestion control bits, urgent pointer
- zero window, silly window syndrome
- for full-speed transmission, window must be larger than bandwidth*delay product

TCP congestion control

- congestion collapse in the 70's and 80s
- AIMD: additive increase, multiplicative decrease
- round-trip-time (RTT) converts TCP congestion window into rate control
- ways of detecting congestion before it occurs: increase in RTT
- TCP Reno: aggressive window decrease, slightly less aggressive when fast retransmit is triggered by duplicate acks
  - TCP Reno always waits until a packet is lost (probably due to congestion) before slowing down
- TCP Vegas: slow down linearly if the RTT is above minimum, increase linearly otherwise
- TCP Cubic: return quickly to almost the window size that experienced packet loss, then grow slowly, then (if no packet loss) start growing more quickly again

queueing and fairness

- FIFO: packets added to end of queue, dropped if queue is full
- Random Early Discard attempts to slow down TCP flows before queue is full
- priority queues can favor some classes of traffic
- global fairness is generally impossible, but can be approximated
- local fairness is easier, but it favors flows that cross fewer congested routers
- fair queueing tries to send the same number of bits per unit time for every flow that has data to send

other transport protocols

- Stream Control Transmission Protocol
- Real Time Protocol and Real Time Control Protocol
- Real Time Streaming Protocol
- Session Initiation Protocol

p2p technology

- bittorrent
- distributed hash tables
- bitcoin
- allnet

project 2

- DNS
- stateless server
  - server does not have to keep track of clients, therefore is stateless
- UDP
- practice using the sockets interface