Computer Communications Networks

TCP Variants and UDP

Review: TCP

- Connection management
  - packet handshake
- Flow control
  - sliding window
- Error control
  - error detection and recovery
- Congestion control
  - slow start and congestion avoidance

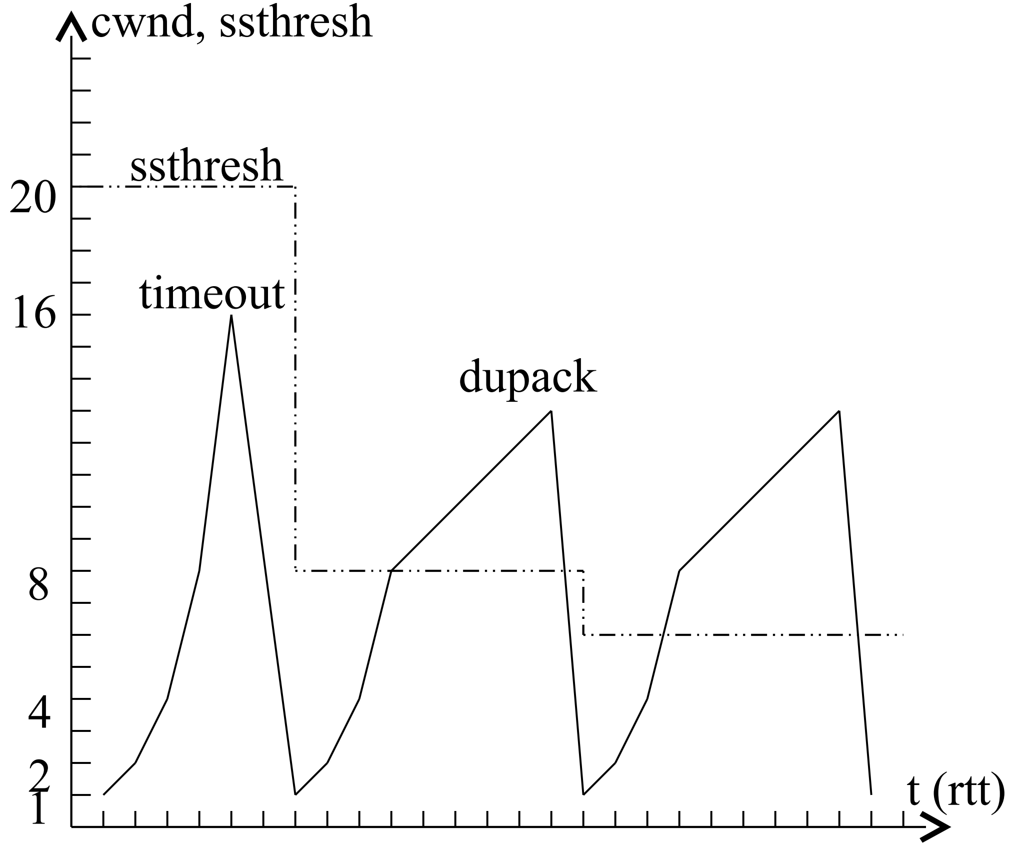
TCP Tahoe

- "Old" TCP
- TCP Tahoe
  - slow start
    - when cwnd < ssthresh, exponential increase
  - congestion avoidance
    - when cwnd >= ssthresh, linear increase
  - timeout
    - ssthresh=cwnd/2, cwnd=1 MSS
  - fast retransmit

Fast retransmit

- Duplicate acknowledgment
  - example
    - rcv: [0, 499], [500, 999], [1500, 1999], [2000, 2499], [2500, 2999]
    - ack: 500, 1000, 1000, 1000, 1000 (3rd dupack)
- Congestion control (fast retransmit)
  - on 3rd dupack: ssthresh=cwnd/2; cwnd=1 MSS
- Error control
  - retransmit: [1000,1499]

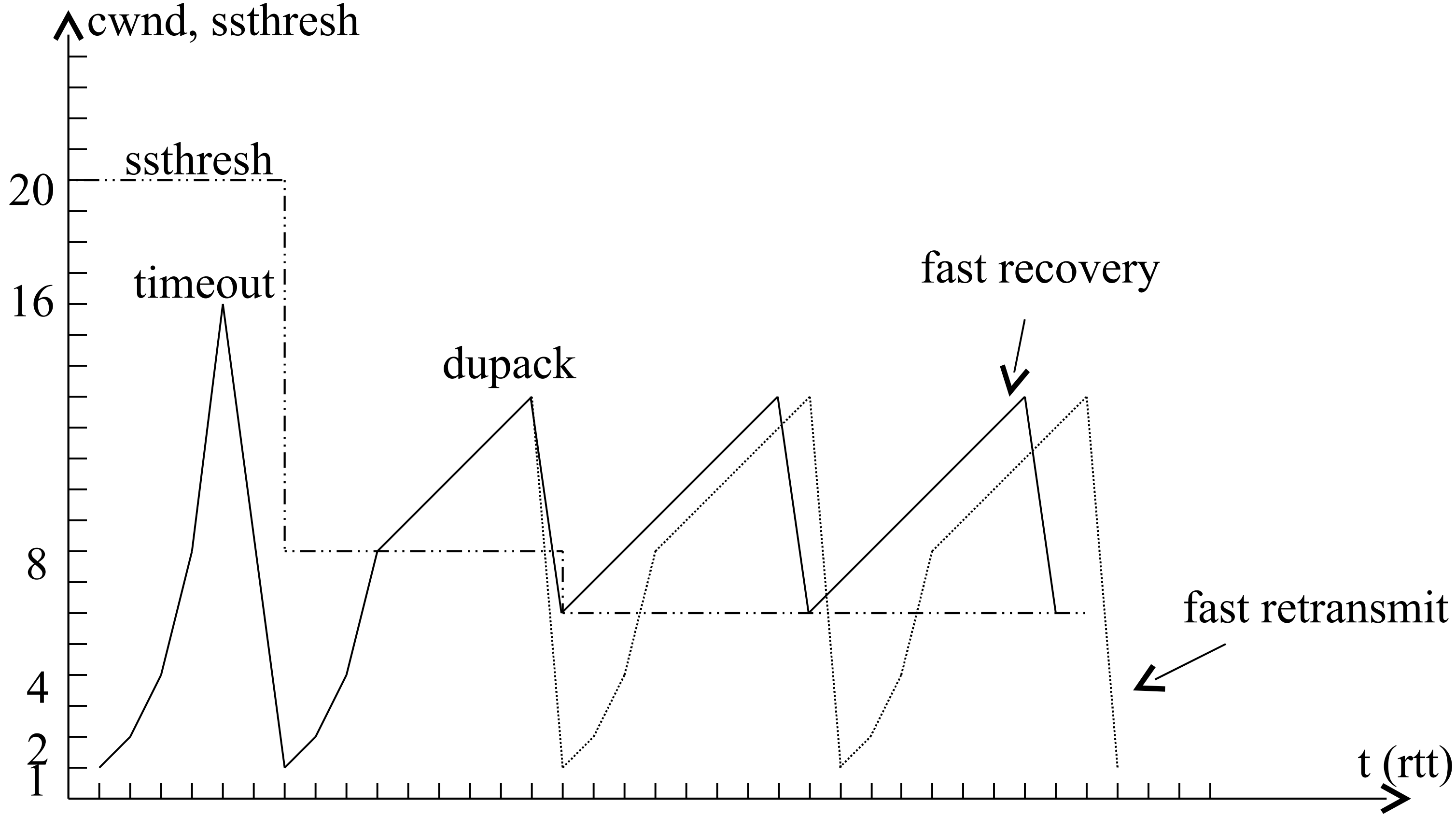
Fast retransmit: cwnd



TCP Reno

- TCP Reno
  - slow start
  - congestion avoidance
  - timeout
  - on 3rd dupack, fast recovery
    - ssthresh=cwnd/2
    - cwnd=ssthresh

Fast recovery: cwnd



More TCP variants

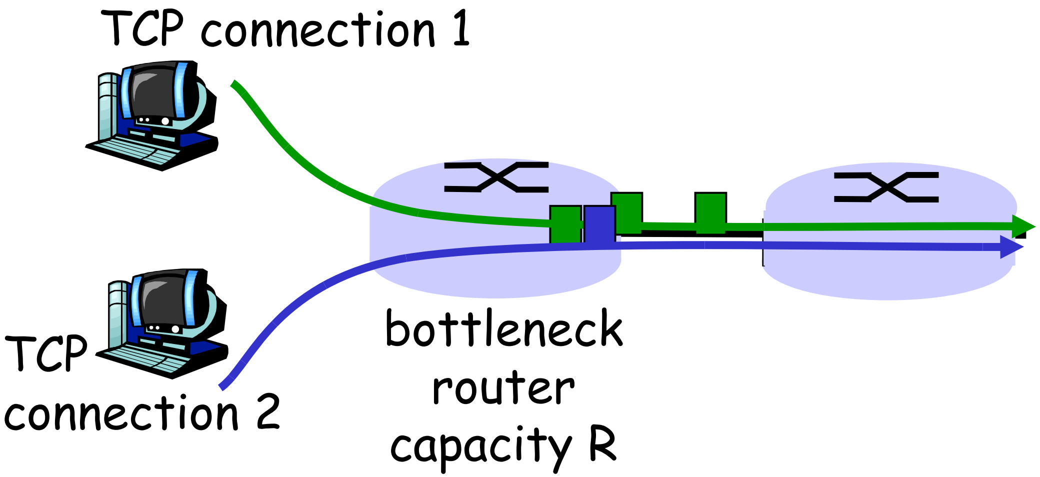
- TCP NewReno
  - partial acknowledgment (for multiple losses)
  - now popular over the Internet
- TCP SACK
  - selective acknowledgment
- TCP Vegas
  - delay-based congestion control
  - increased delay indicates network congestion

TCP throughput

- What's the average throughout of TCP (Reno) as a function of window size and RTT?
  - Ignore slow start
- Let W be the window size when loss occurs.
- When window is W, throughput is W/RTT
- Just after loss, window drops to W/2, throughput to W/2RTT.
- Average throughout: 0.75 W/RTT
- Loss event rate: 1/[3/8(W/MSS)^2]

TCP fairness

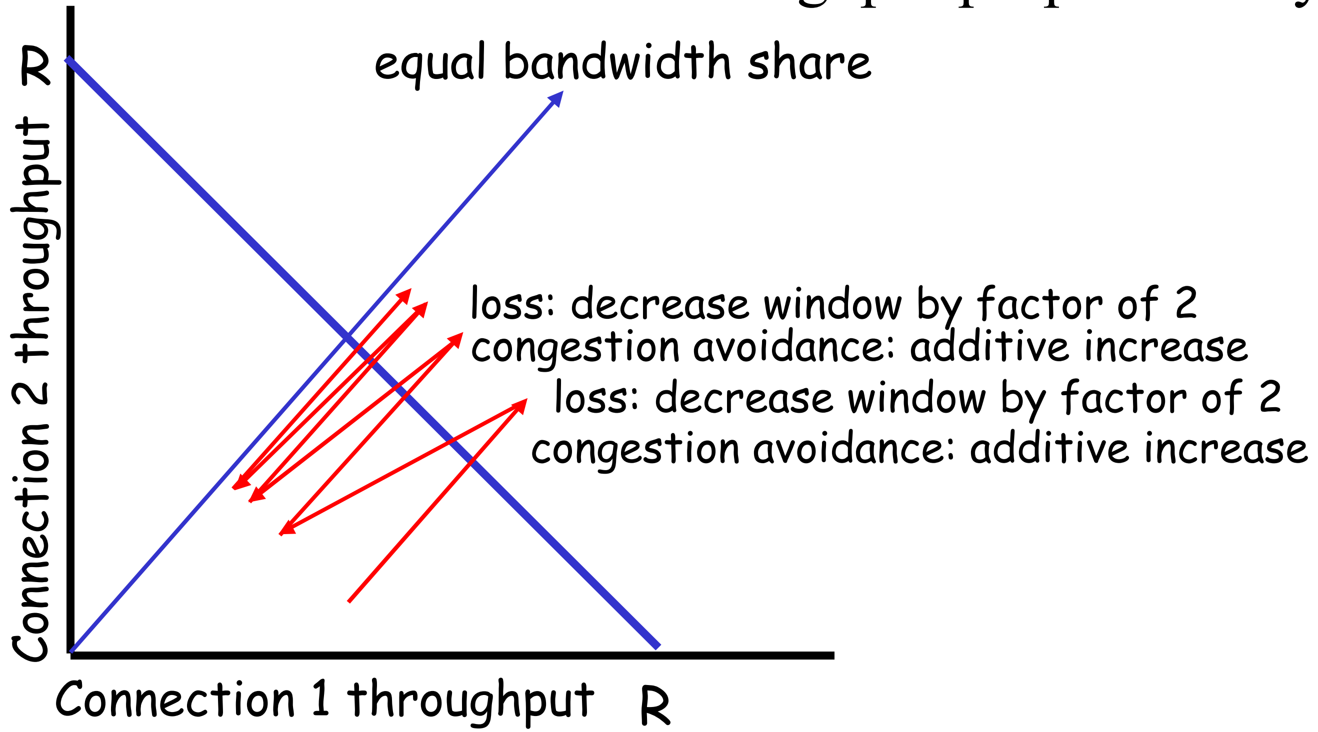
Fairness goal: if K TCP sessions share same bottleneck link of bandwidth R, each should have average rate of R/K



Why is TCP fair?

Two competing sessions:

- Additive increase gives slope of 1, as throughout increases
- Multiplicative decrease decreases throughput proportionally



Challenges on TCP

- TCP over high-speed (long-delay) networks
  - limited sequence space
  - limited window size
    - TCP large window
  - “slow” congestion recovery
    - cwnd: linear increase per RTT
  - high-speed TCP, FAST, etc
    - http://www.icir.org/floyd/longpaths.html

Challenges on TCP: “long, fat pipes”

- Example: 1500 byte segments, 100ms RTT, want 10 Gbps throughput
- Requires window size W = 83,333 in-flight segments
- Throughput in terms of loss rate:

$$\frac{1.22 \cdot MSS}{RTT\sqrt{L}}$$

- → $L = 2 \cdot 10^{-10}$ *Wow*
- New versions of TCP for high-speed needed!

Challenges on TCP: wireless

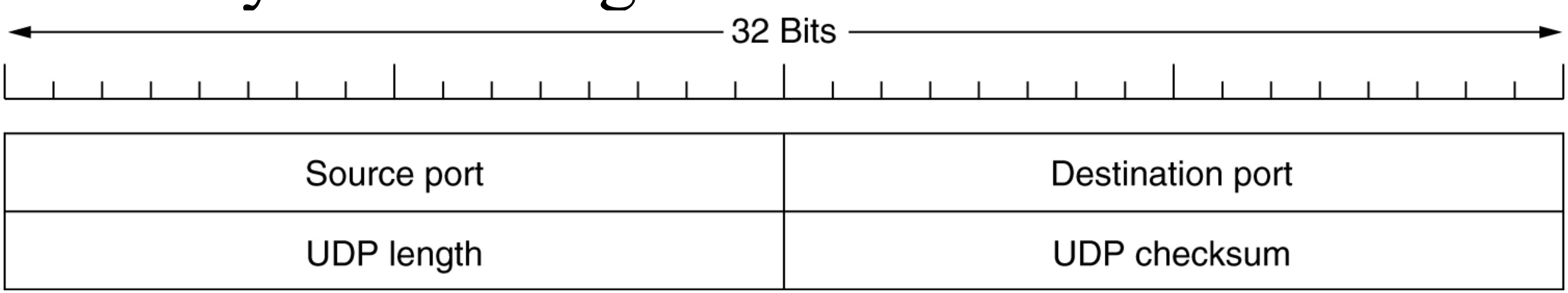
- TCP over wireless
  - packet loss
    - transmission error vs network congestion
  - local retransmission
    - link-layer retransmission
    - reduced packet loss ratio
    - increased variability: effective bandwidth and delay
  - http://www.icir.org/floyd/tcp_small.html

UDP

- User Datagram Protocol (UDP)
  - connectionless
    - no connection management
  - unreliable
    - no flow, error, congestion control
      - TCP-friendly congestion control
- Why UDP?
  - sometimes TCP is an overkill
  - e.g., multimedia

UDP header

- Multiplex
  - source/destination port number
- Error checking (optional)
  - checksum (TCP/IP-style)
- Why “UDP length”?



Summary

- TCP
  - fast retransmit
  - fast recovery
- UDP

Next

- The Application Layer Protocol
  - read CN Section 7.3