Advanced Computer Networks

Congestion Control

Jianping Pan
Summer 2007

Review: Internet design

- Design principles
  - store-and-forward packet switching
  - end-to-end argument
- TCP/IP protocol stack
  - IP: best-effort packet delivery
    - possible errors: loss, duplication, corruption, out-of-order
  - TCP: connection-oriented, reliable data transfer
    - connection management
      - 3-way handshake
    - flow, error and congestion control

Flow control

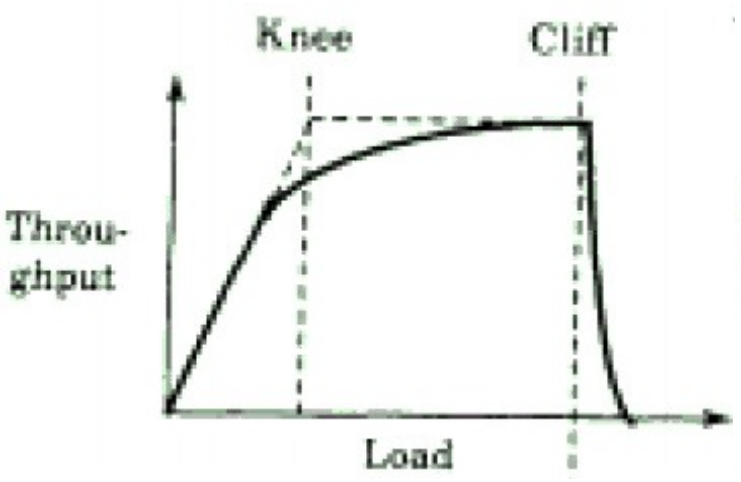
- Problem
  - a fast sender to overflow a slow receiver's buffer
- Approach
  - stop-and-go, or
  - let receiver advertise available buffer space, or
  - let receiver choose the sending rate
- TCP flow control
  - sliding window with variable size
  - advertised by the receiver: ack number, win size

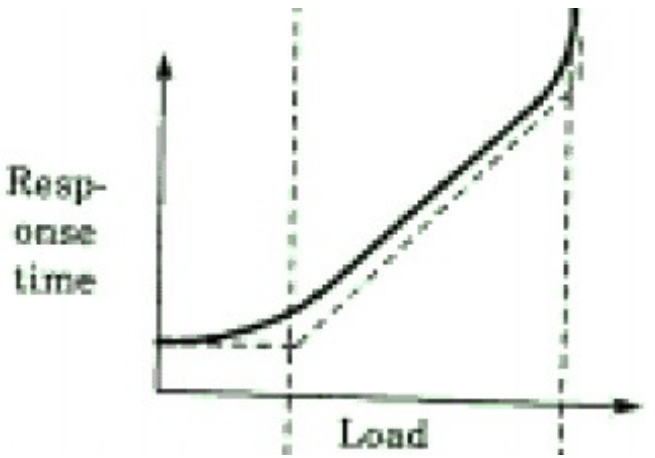
Error control

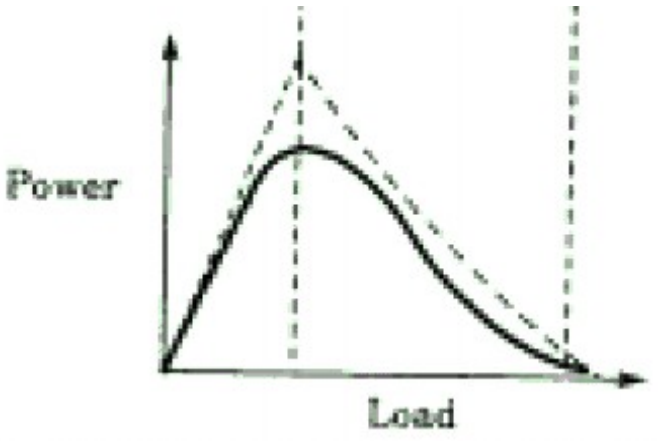
- Problem
  - packets get lost, duplicated, corrupted, reordered
- Approach
  - error checking and correction
  - error notification and recovery
- TCP error control
  - sequence number, checksum
  - receiver acknowledgment, sender timer
  - sender retransmission

Congestion control

- Problem
  - “network buffer overflow”
  - packet loss, retransmission, more packet loss
  - congestion collapse







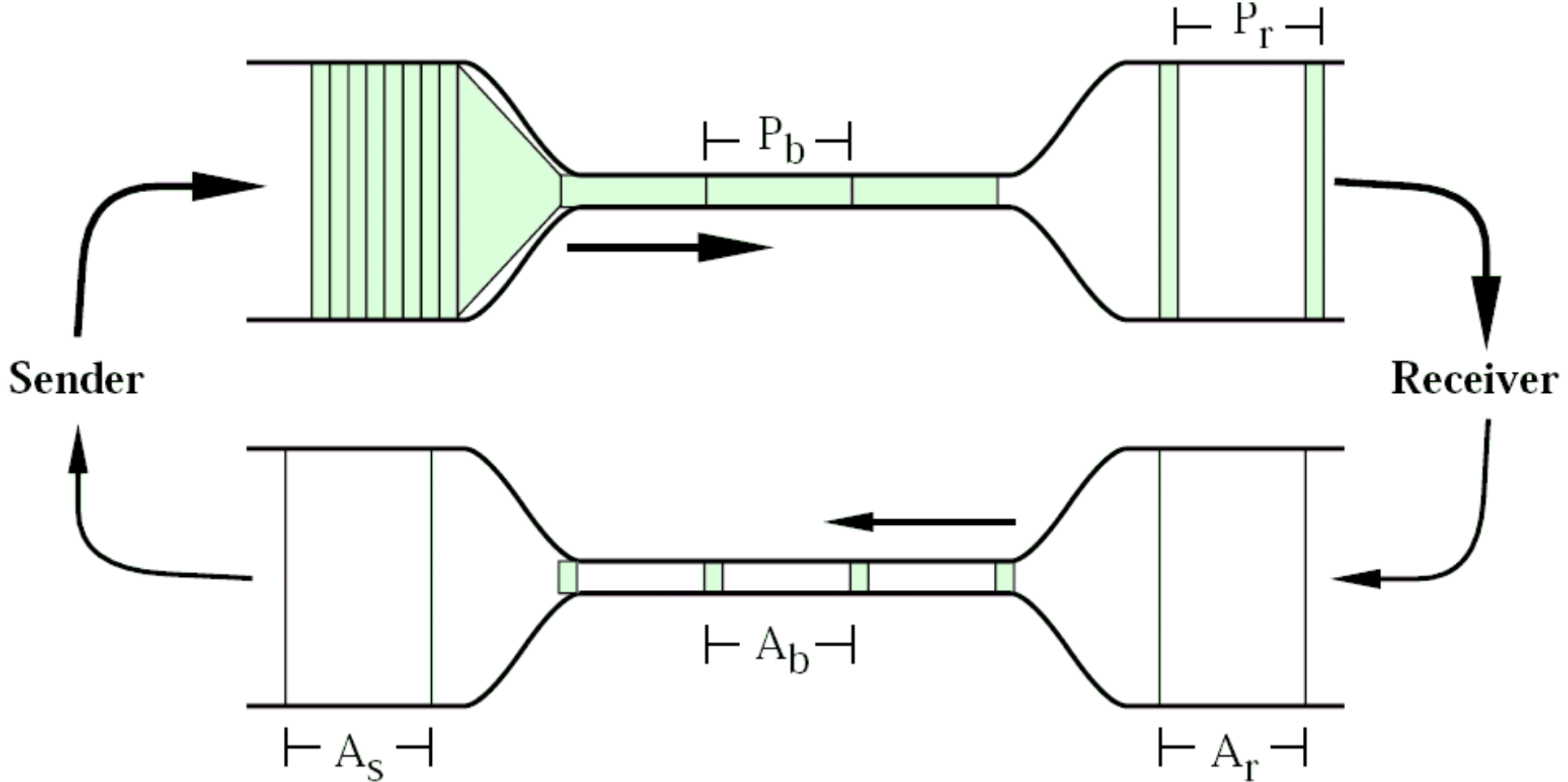
DECbit [RJ88]

- Binary feedback
  - router: set congestion bit
    - congestion detection: queue length
    - feedback filter: average since the previous renewal point
    - feedback selection: throughput fair share
  - user (endpoint): respond to congestion bit
    - signal filter: binary decision based on congestion bits
    - decision frequency
      - one rtt to get signal, another rtt to know reaction
    - increase/decrease algorithm
      - additive/multiplicative increase/decrease: 2x2
      - AIMD: additive increase: 1; multiplicative decrease: 0.875

Q: why AIMD prevails?

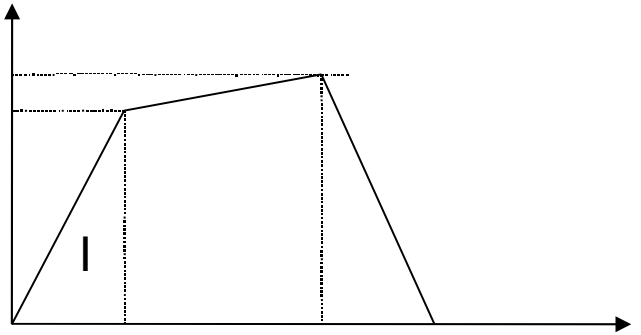
TCP congestion control [JK88]

- Principle
  - packet conservation
  - “ack self-clocking”



“Slow” start

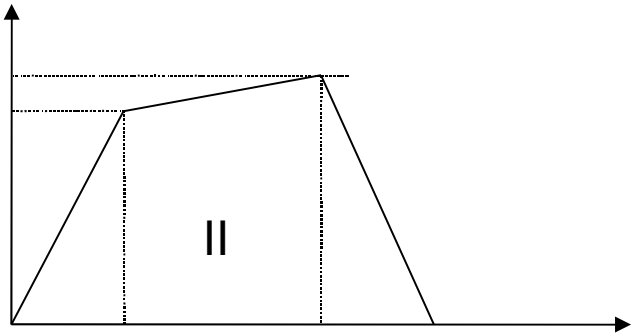
- Sender variables
  - congestion window (cwnd)
    - sender window = min {buffer size, receiver window, cwnd}
    - initially, cwnd = 1 MSS: maximum segment size
  - slow-start threshold (ssthresh)
- Slow start
  - when cwnd < ssthresh
  - on each new ack
    - cwnd += 1 MSS
    - effectively doubling cwnd every RTT



Q: how to determine ssthresh?

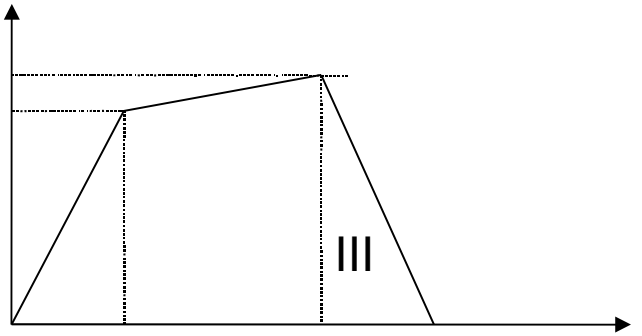
Congestion avoidance

- Congestion avoidance
  - when cwnd > ssthrehsh
  - on each new ack
    - cwnd += MSS^2/cwnd
    - effectively cwnd += 1 MSS every RTT
    - linear increment



Network congestion

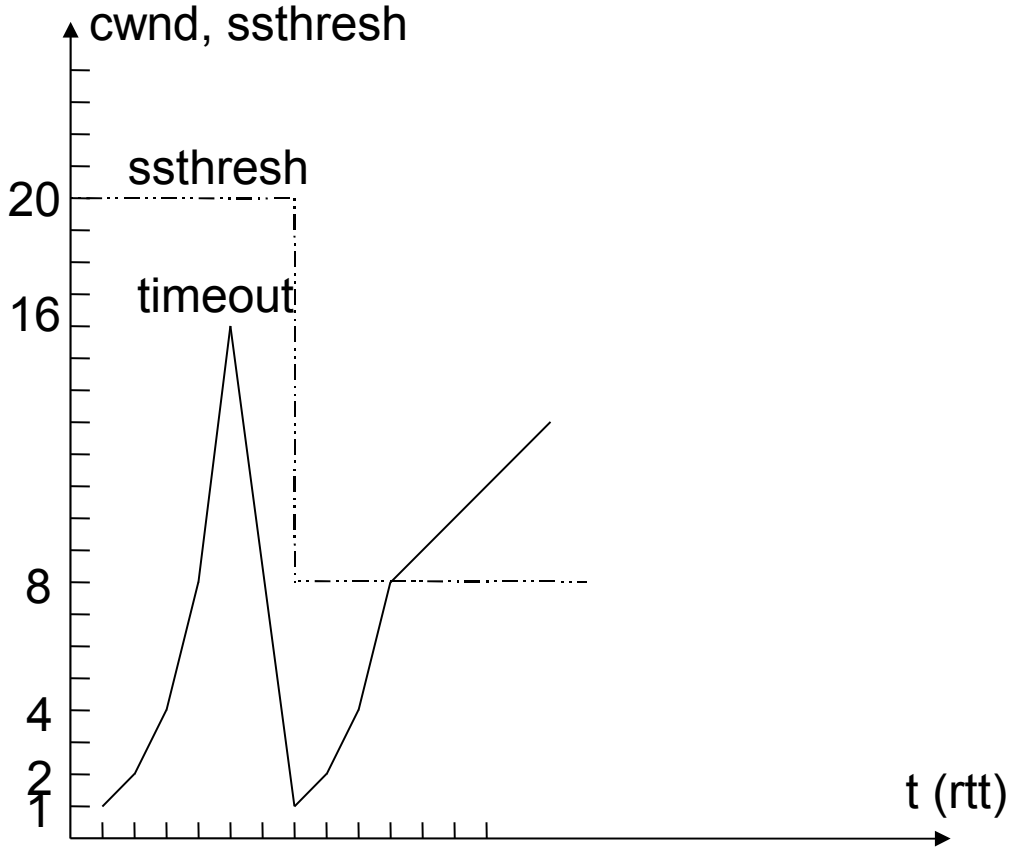
- Packet loss signals
  - timeout
  - 3 duplicate acknowledgments
    - TCP cumulative acknowledgment
- Timeout
  - srtt = srtt + g1 (rtt - srtt)
  - rttv = rttv + g2 (|rtt-srtt|-rttv)
  - rto = srtt + g3 rttv
  - g1: 0.125, g2: 0.25
  - g3: initially 2, now 4



Q: why g3 increased to 4?

Timeout retransmission

- Congestion control
  - ssthresh = cwnd / 2
  - cwnd = 1 MSS
  - followed by slow start
- Error control
  - retransmit packet
  - backoff timer
    - rto = rto * 2
    - until maxrto is reached

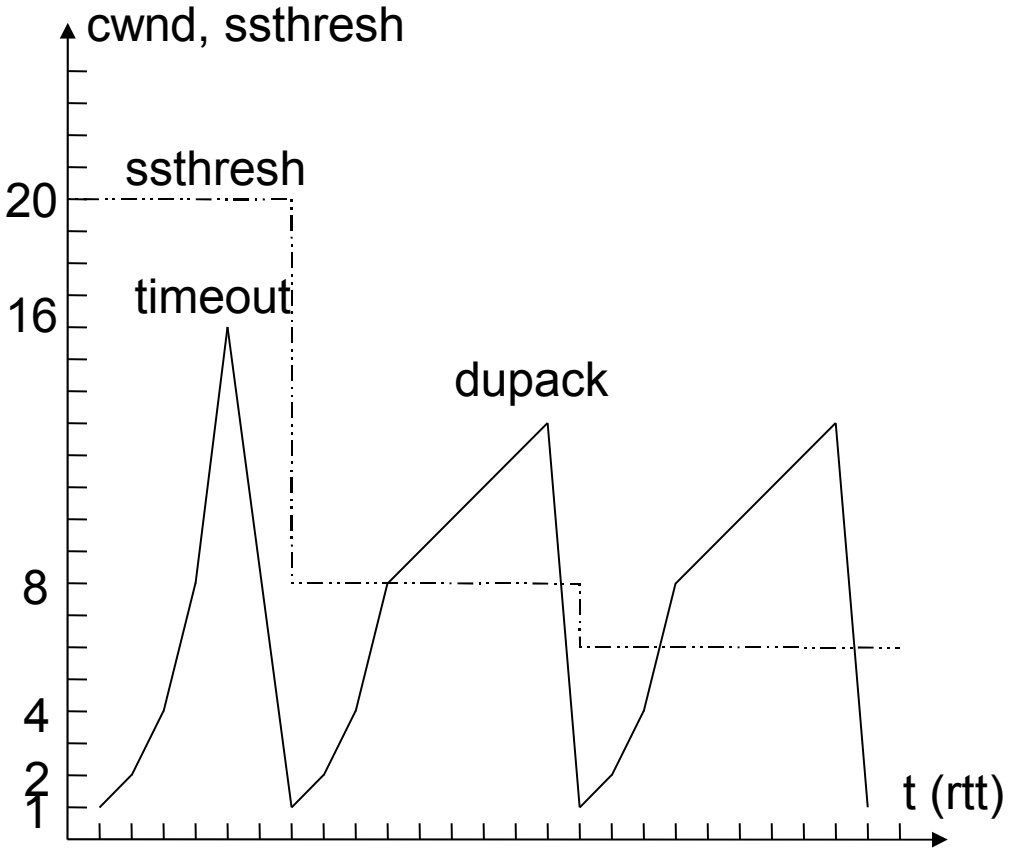


Fast retransmit

- 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
    - followed by slow start
- Error control
  - retransmit: [1000,1499]

Q: why fast retransmit is better?

Fast retransmit: cwnd

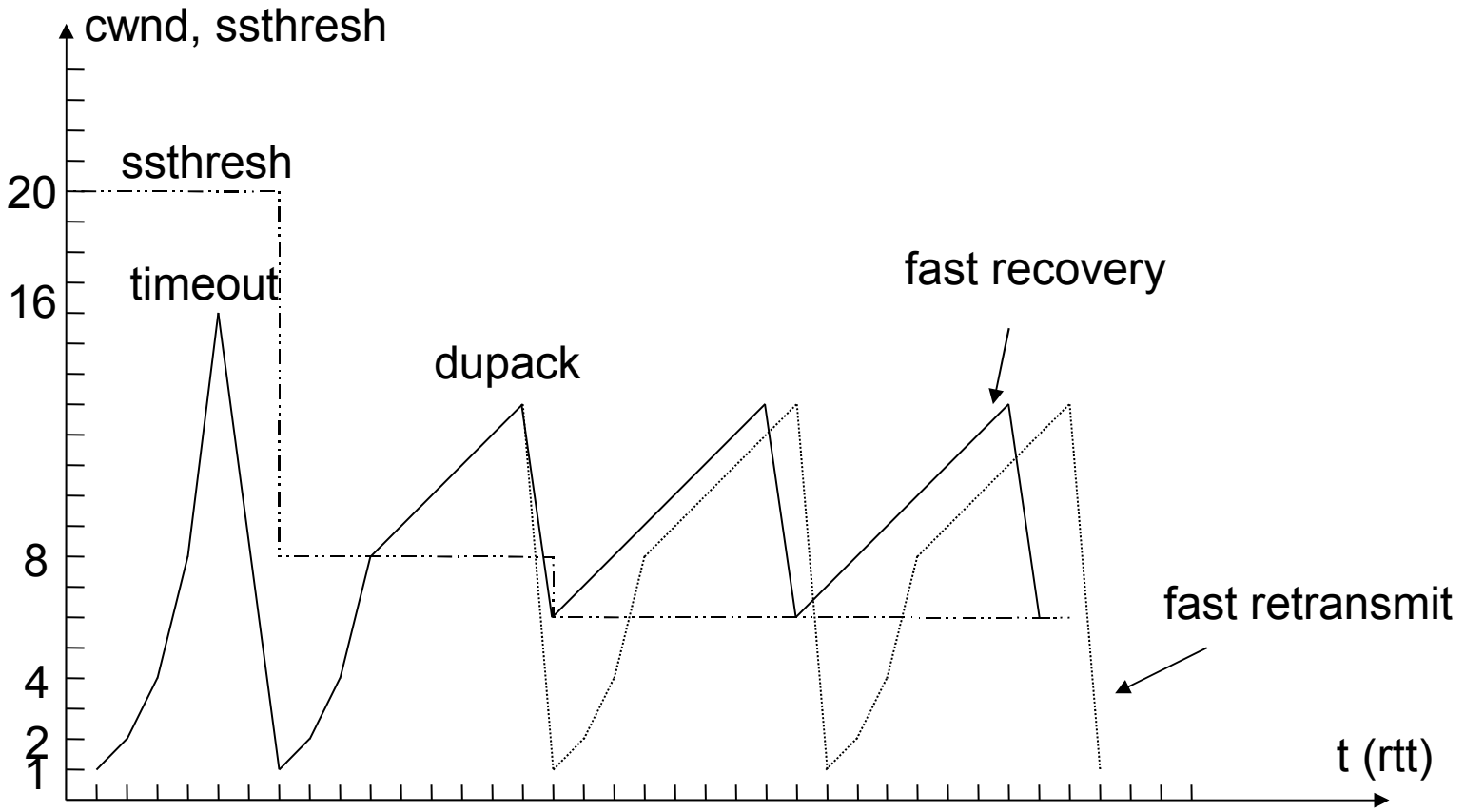


Fast recovery

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

Q: why treat dupack differently?

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

Challenges on TCP

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

Challenges on TCP: more

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

This lecture

- TCP congestion control
  - basic congestion control algorithms
    - slow-start
    - congestion avoidance
    - fast transmit
    - fast recovery
    - selective acknowledgment
- Explore further
    - [FJ93] S. Floyd and V. Jacobson, Random Early Detection Gateways for Congestion Avoidance, IEEE/ACM Transactions on Networking, Vol. 1, No. 4, pp. 397-413, August 1993. [RED]

Next lectures

- TCP Vegas
  - delay-based congestion control
- TCP-friendly congestion control
  - TCP throughput model
- XCP
  - explicit congestion control

- Bring up your course project web page by June 15