CSc 450/550
Computer Networks
Congestion Control

Jianping Pan

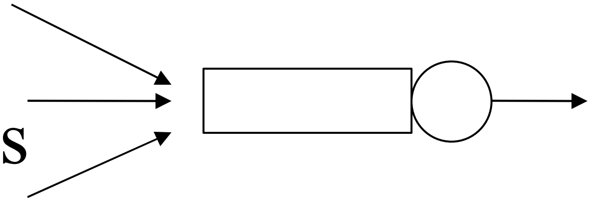
Summer 2007

Review: TCP mechanisms

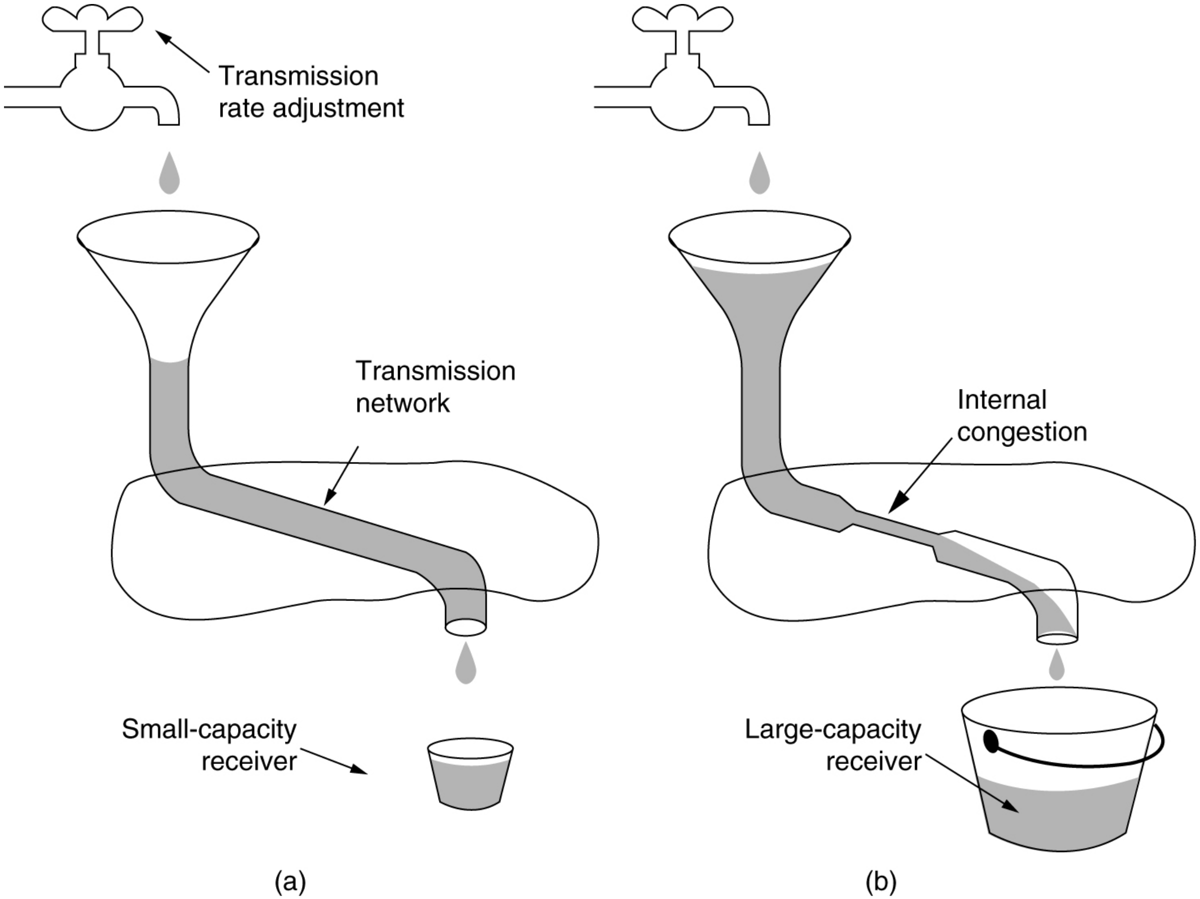
- Connection management
  - packet handshake
- Flow control
  - sliding variable window
- Error control
  - error detection
  - error recovery

Why congestion control?

- Flow control
  - coordinate sender and receiver (buffer)
- Network congestion
  - coordination between the sender and network
    - avoid a sender to overflow a router
  - coordination among many senders
    - traffic aggregation from many senders
  - congestion syndrome
    - increasing queuing delay, packet drop



Flow vs congestion control



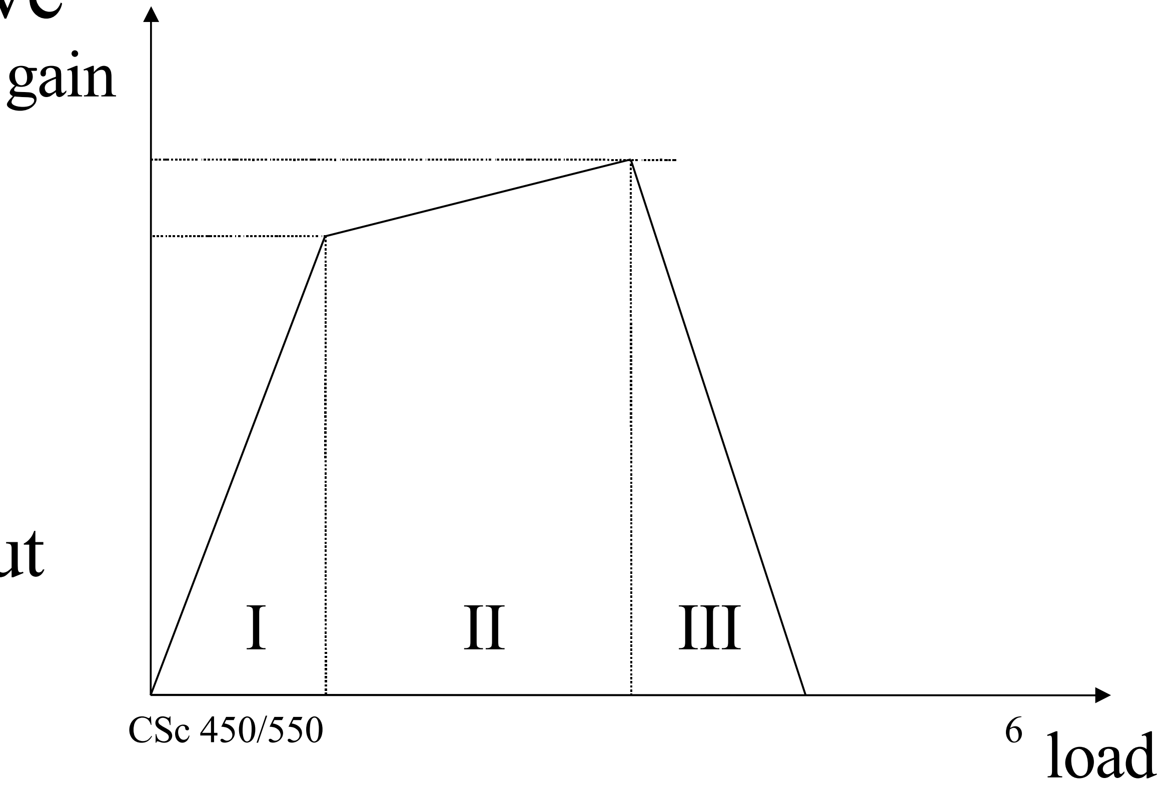
Congestion control approaches

- End-to-end approach
  - congestion indicators for the endpoint
    - packet loss
      - also can be caused by transmission error
    - increasing round-trip time
      - also can be caused by alternative routes
- Network-assisted approach
  - ICMP source quenching
  - explicit congestion notification (ECN)

Q: why packet loss and increasing rtt?

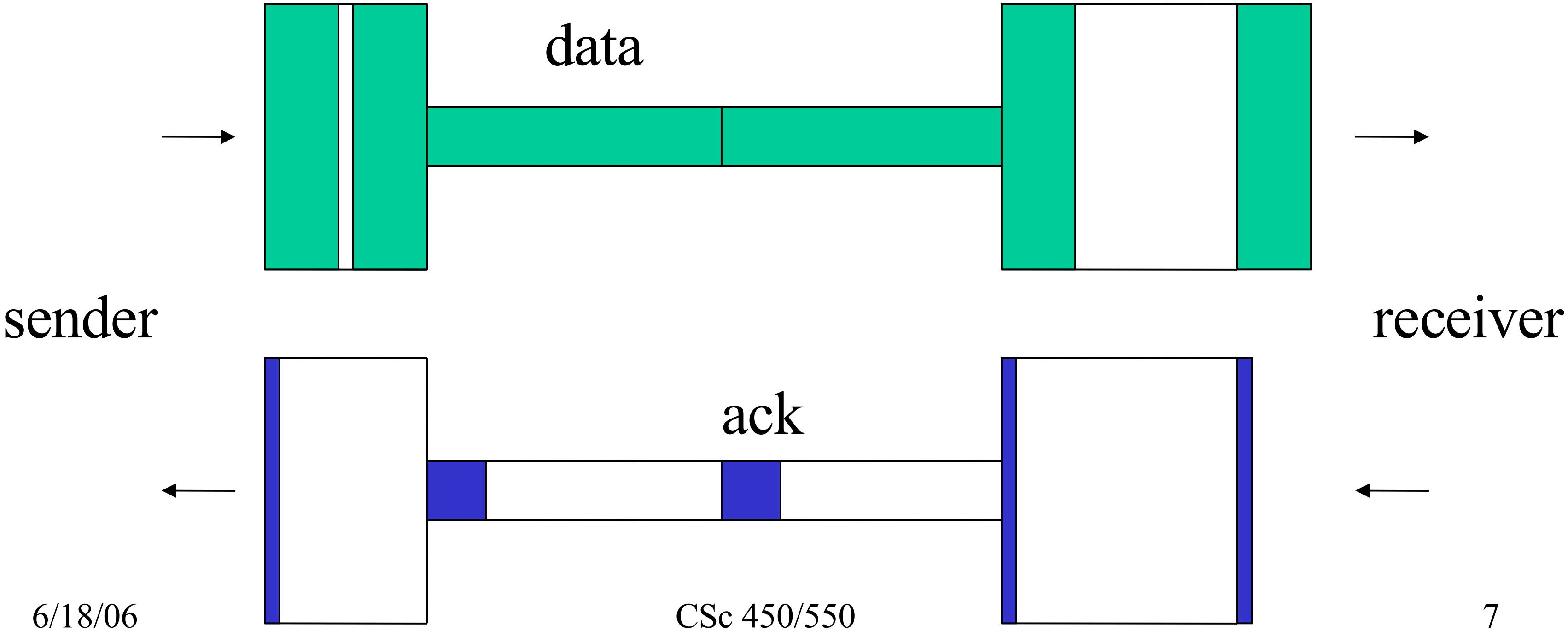
Load-gain curve

- Gain: ~ throughput/delay
- Load-gain curve
  - low load
  - medium load
  - high load
- Congestion
  - low throughput
  - high delay
  - very low gain



Congestion control principles

- Principle: packet conservation in steady state
  - acknowledgment self-clocking



Q: what does ack spacing mean?

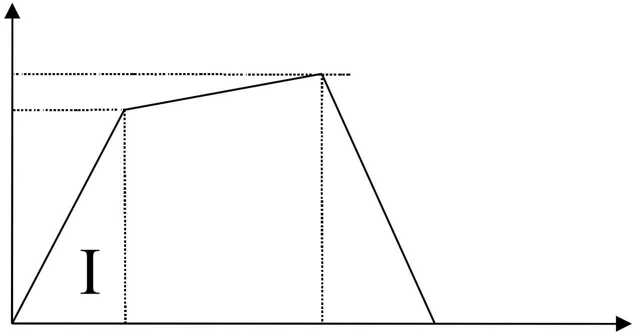
TCP congestion control

- Changes at sender only
  - an add-on to TCP flow control
- Sender variables
  - congestion window (cwnd)
    - sender window = min {…, …, cwnd}
    - initially, cwnd = 1 MSS (maximal segment size)
  - slow-start threshold (ssthresh)
    - initial ssthresh

Q: why sender-only is better?

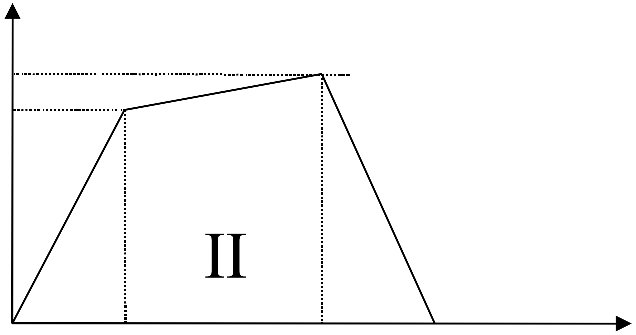
Slow start

- Slow start
  - when cwnd < ssthresh
  - on each new ack
    - cwnd += 1 MSS
    - effectively, doubling cwnd every RTT
  - “start small, but grow really fast”
    - Q: why?



Congestion avoidance

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



Q: why increase cwnd to “avoid” congestion?

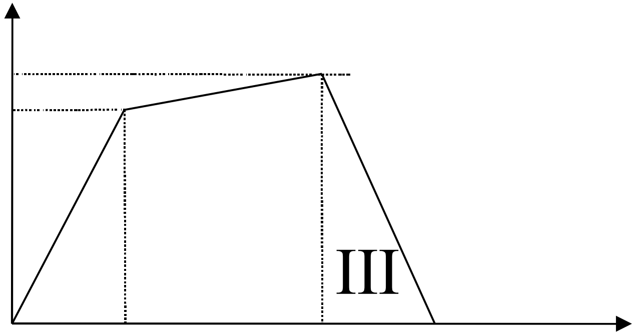
Network congestion

- cwnd is always increased in slow-start and congestion avoidance
  - network congestion is inevitable
- Network congestion indicator
  - TCP treats packet loss as network congestion
- Packet loss indicators
  - acknowledgment timeout
  - 3 duplicate acknowledgments

Q: packet loss == network congestion?

Timeout retransmit

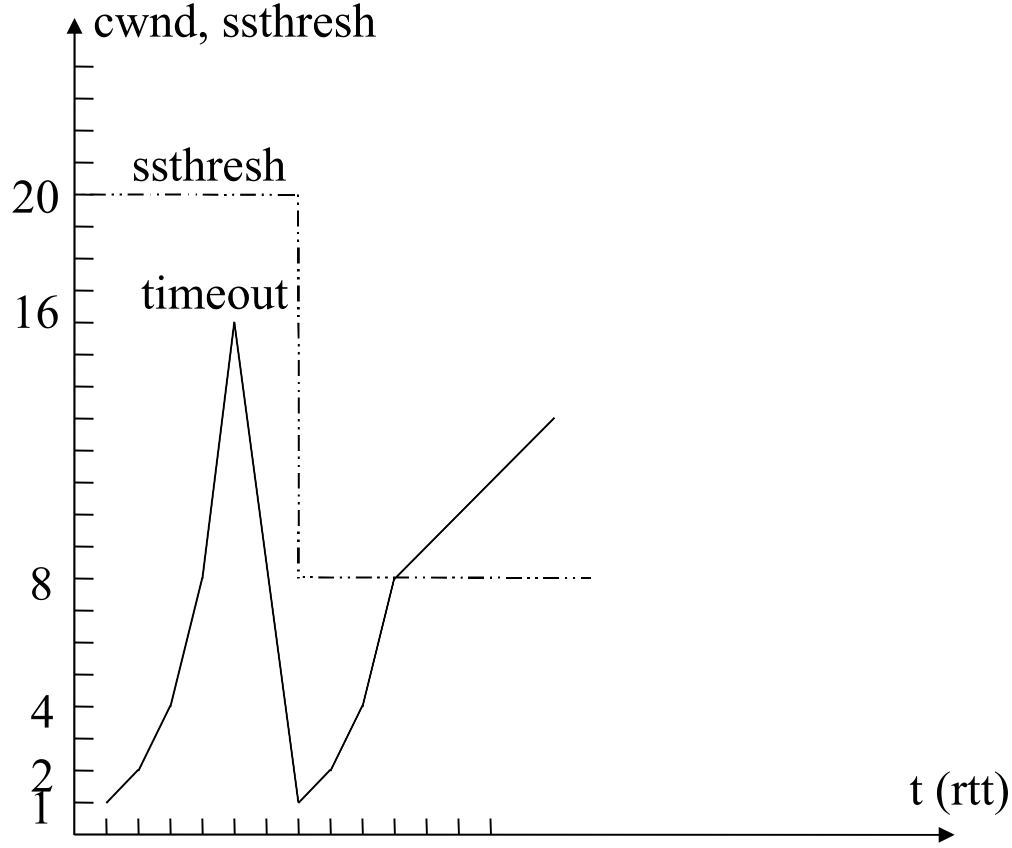
- Timeout
  - RTO = *d* (SRTT + *c* RTTV)
- Congestion control
  - ssthresh = cwnd / 2
  - cwnd = 1 MSS
  - followed by slow-start
- Error control
  - retransmit packet
  - backoff timer



Q: how to calculate SRTT and RTTV?

Congestion window

- Slow-start
- Congestion avoidance
- Timeout retransmission
  - TCP timeout is quite conservative
  - pay attention to how ssthresh is adjusted!

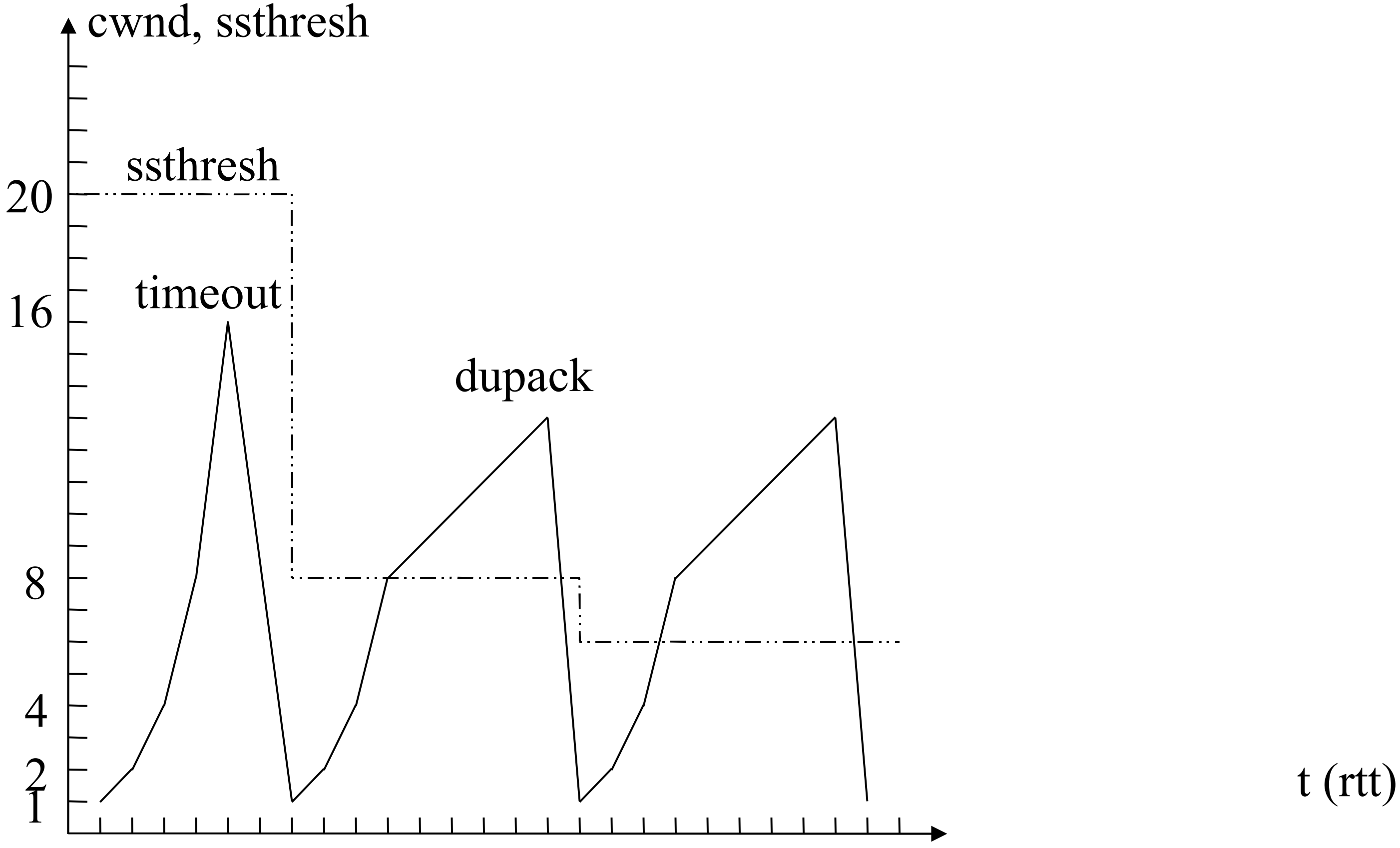


Fast retransmit

- 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: dupack == loss?

Fast retransmit: cwnd



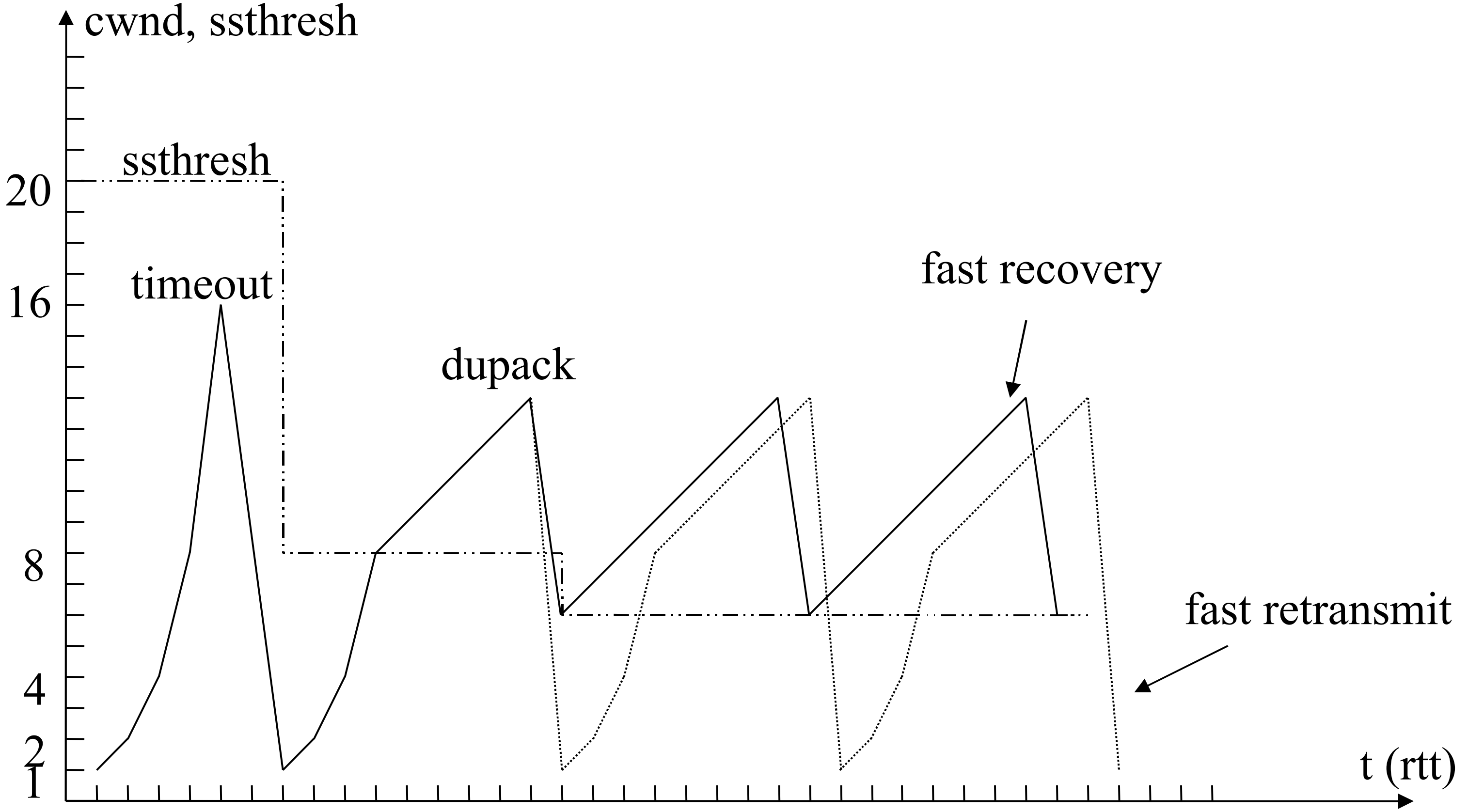
Q: why “fast” retransmit?

Fast recovery

- 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 recovery)
  - on 3rd dupack: cwnd=ssthresh=cwnd/2
  - followed by congestion avoidance
- Error control
  - retransmit: [1000,1499]

Q: fast transmit vs recovery?

Fast recovery: cwnd



Q: why fast recovery?

This lecture

- Congestion control
  - purpose and approach
  - TCP congestion control
    - slow-start, congestion avoidance
    - timeout retransmit
    - fast retransmit, fast recovery
- Explore further
  - TCP congestion control [RFC2581]
  - http://www.cs.uvic.ca/~pan/csc485

Next lectures

- May 21: more on TCP congestion control
- May 25: UDP
- May 27: extra before-exam office hours
  - regular office hours: MR
  - use the google group: get help and help others
- May 28: 2nd in-class midterm exam