CS 330: Network Applications & Protocols

Transport Layer

Galin Zhelezov
Department of Physical Sciences
York College of Pennsylvania

Overview of Transport Layer

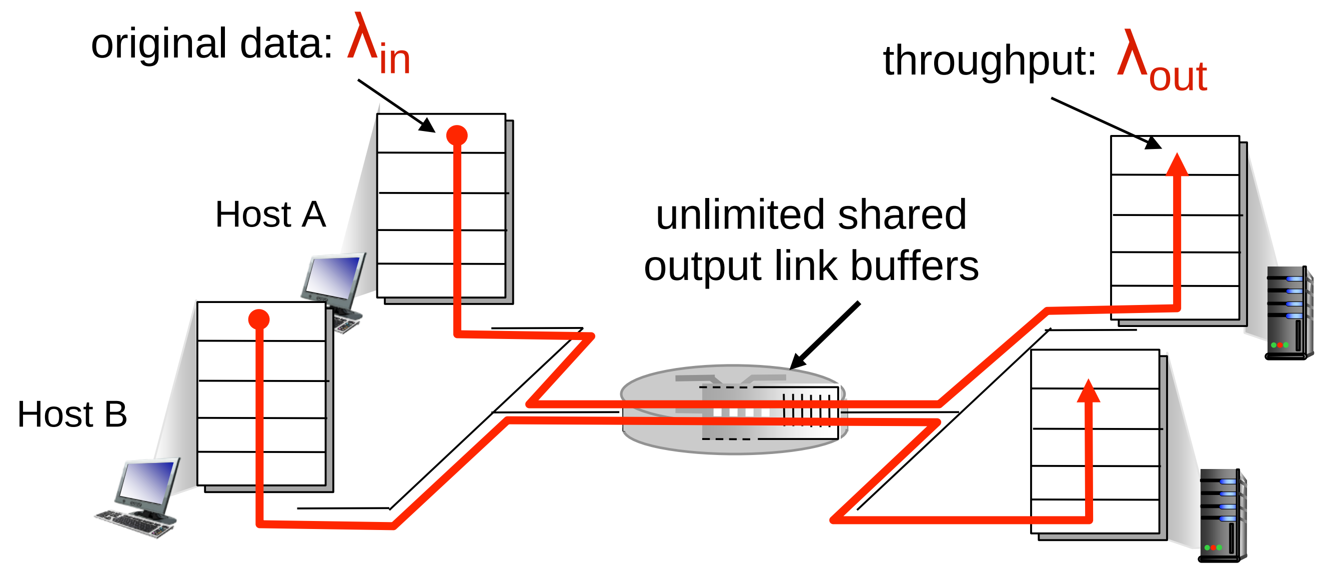
- **Transport-layer Services**
- **Multiplexing and Demultiplexing**
- **Connectionless Transport: UDP**
- **Principles of Reliable Data Transfer**
- **Connection-oriented Transport: TCP**
  - Segment Structure
  - Reliable Data Transfer
  - Flow Control
  - Connection Management
- **Principles of Congestion Control**
- **TCP Congestion Control**

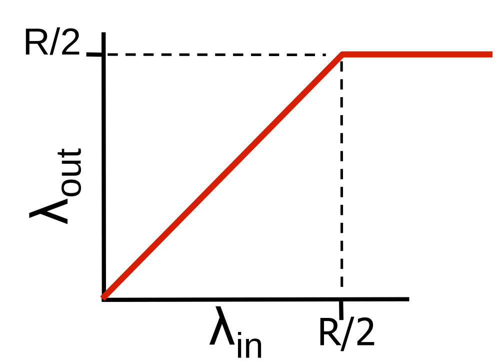
Principles of Congestion Control

- **What is congestion?**
  - Informally: "too many sources sending too much data too fast for the ***network*** to handle"
  - Different from flow control
    - Flow control used to ensure buffers at receiver do not overflow
    - Flow control does nothing to prevent router buffers from overflowing
- **What problems can congestion cause?**
  - Lost packets (buffer overflow at routers)
  - Long delays (queueing in router buffers)

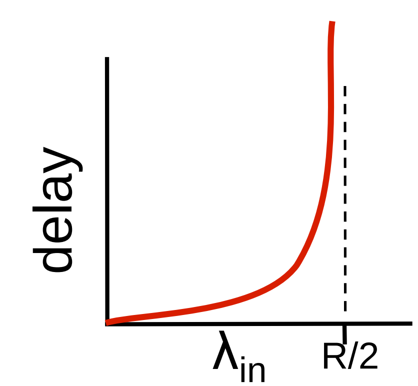
Causes/Costs of Congestion: Scenario #1

- **Consider the following scenario**
  - Two senders, two receivers
  - One router with infinite buffers
  - Single output link capacity shared by senders with capacity R
  - Assume no retransmission necessary
- **Senders cannot send at a rate higher than R/2 since they are sharing single link**
- **As senders max out the output link, the delay between source and destination increases**





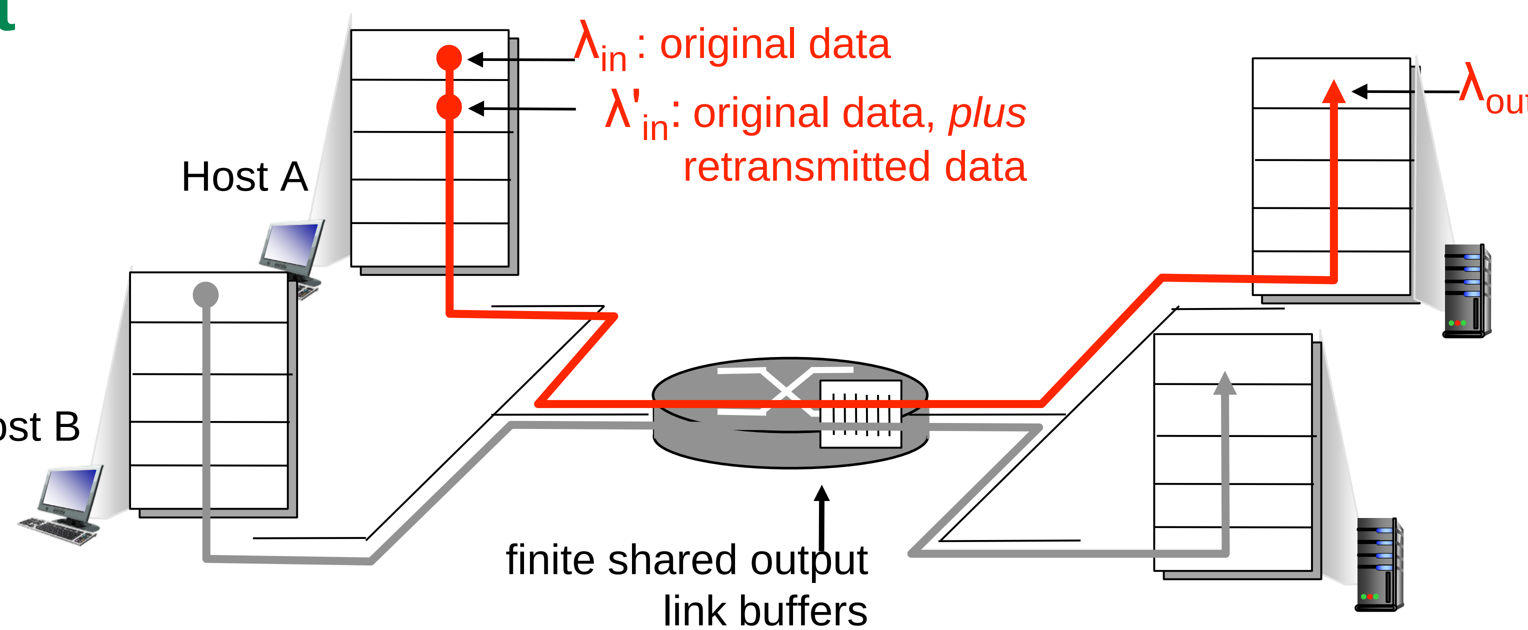
maximum per-connection throughput is R/2

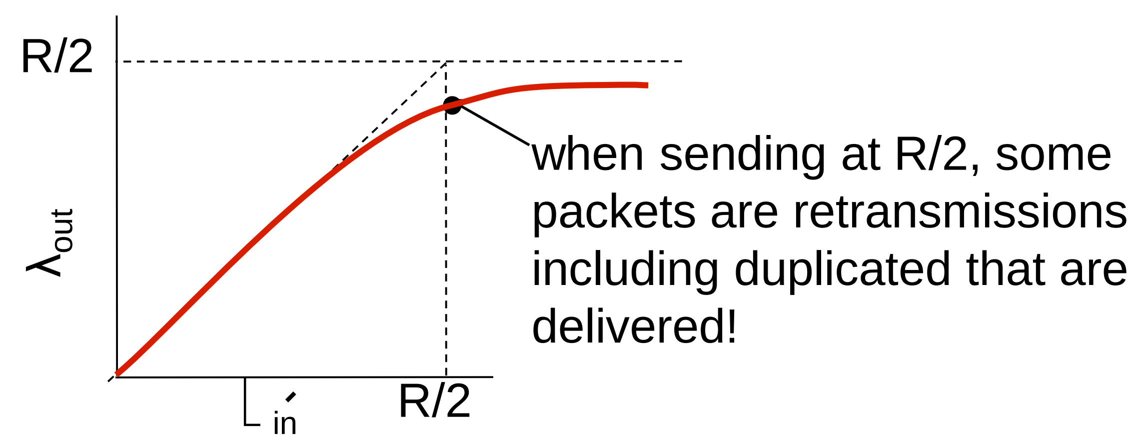


large delays as arrival rate, λ_{in}, approaches capacity

Causes/Costs of Congestion: Scenario #2

- **Modify scenario #1 such that**
  - The router has finite shared buffers
  - The sender may retransmit packets
- **Retransmission reduce the throughput**
  - Packets can be dropped at router if buffers are full
  - Sender may timeout prematurely due to delay in router; send multiple copies of same data





Two Approaches Towards Congestion Control

- **End-to-end congestion control**
  - No explicit feedback from network
  - Congestion inferred from end-system observed loss, delay
  - Approach taken by TCP

- **Network-assisted congestion control**
  - Routers provide feedback to end systems
  - Single bit indicating congestion (SNA, DECbit, TCP/IP ECN, ATM)
  - Explicit rate for sender to send at

Overview of Transport Layer

- **Transport-layer Services**
- **Multiplexing and Demultiplexing**
- **Connectionless Transport: UDP**
- **Principles of Reliable Data Transfer**
- **Connection-oriented Transport: TCP**
  - Segment Structure
  - Reliable Data Transfer
  - Flow Control
  - Connection Management
- **Principles of Congestion Control**
- **TCP Congestion Control**

TCP Congestion Control

- **Must have each sender limit the rate at which it sends traffic as a function of the network congestion**
  - Too much congestion? Send less data.
  - Not much congestion? Full speed ahead!

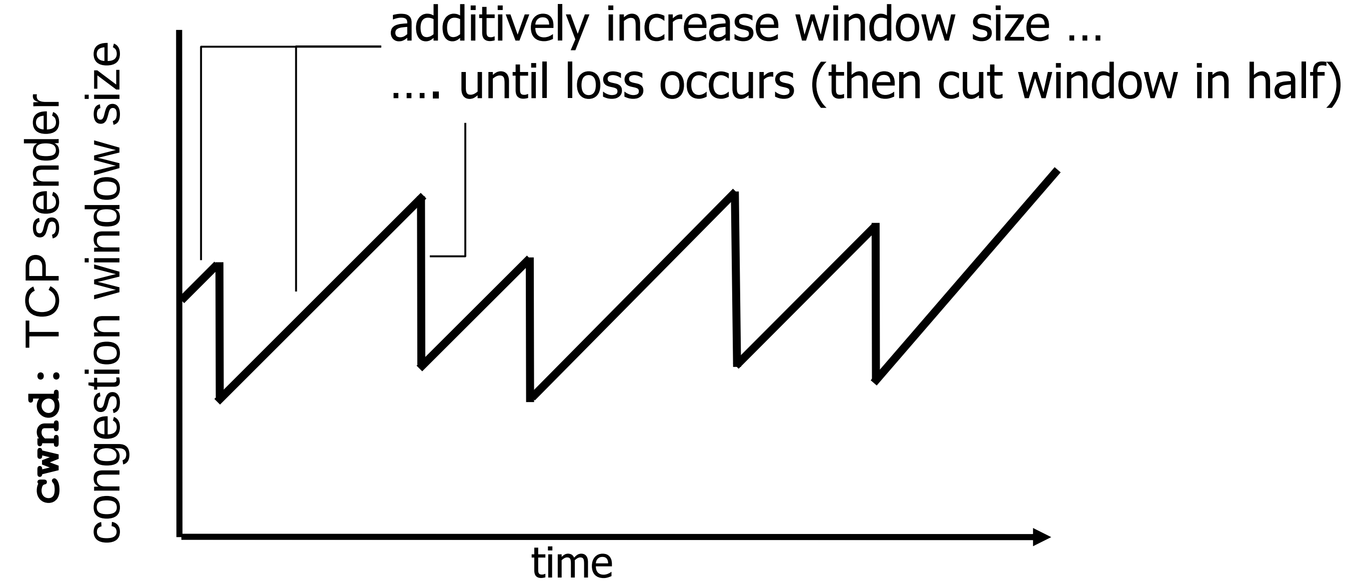
- **How does a TCP sender limit the rate at which it sends data?**

- **How does a TCP sender detect congestion between itself and the destination?**

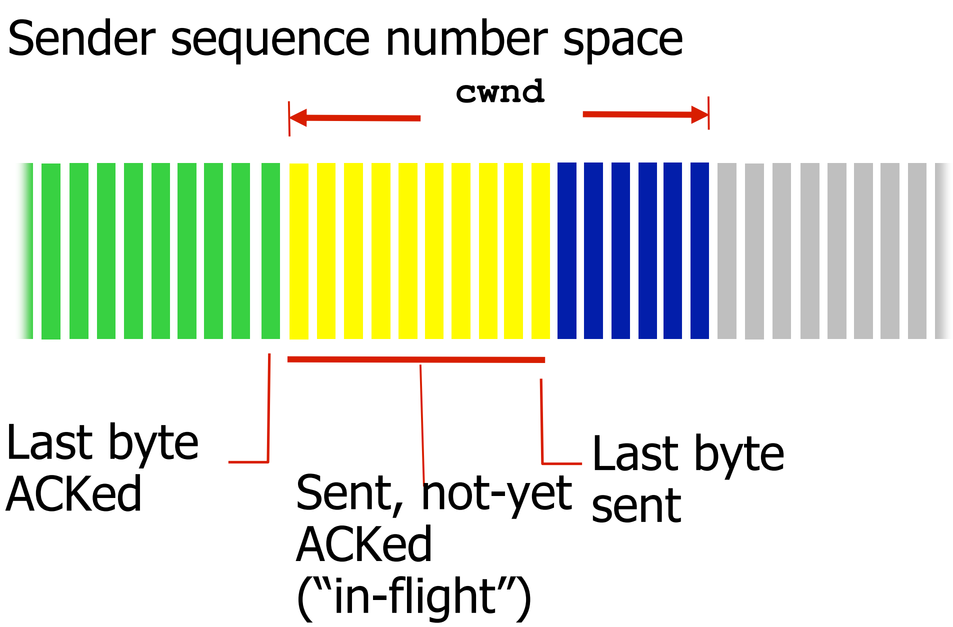
- **How should the sender change the rate at which it sends data based on the network congestion?**

TCP Congestion Control: AIMD

- **Additive Increase Multiplicative Decrease (AIMD)**
  - **Approach**: sender increases transmission rate (window size), probing for usable bandwidth, until loss occurs
    - Additive increase: increase congestion window (**`cwnd`**) by 1 MSS (Maximum Segment Size) every RTT until loss detected
      - ACKs arriving at sender signal the sender to increase its window
    - Multiplicative decrease: cut congestion window in half after a packet loss occurs



TCP Congestion Control (Cont.)



- **TCP sending rate (assuming no limit on receiver's buffer (`rwnd`))**
  - Send **`cwnd`** bytes, wait RTT for ACKS, then send more bytes

$$\boxed{\texttt{rate} \approx \frac{\texttt{cwnd}}{\texttt{RTT}} \texttt{bytes/sec}}$$

- **Congestion window (`cwnd`) is a dynamic, function of perceived network congestion**
  - Sender is limited by both **`cwnd`** and **`rwnd`**
- **Amount of unacked data at sender may not exceed the minimum of `cwnd` and `rwnd`**

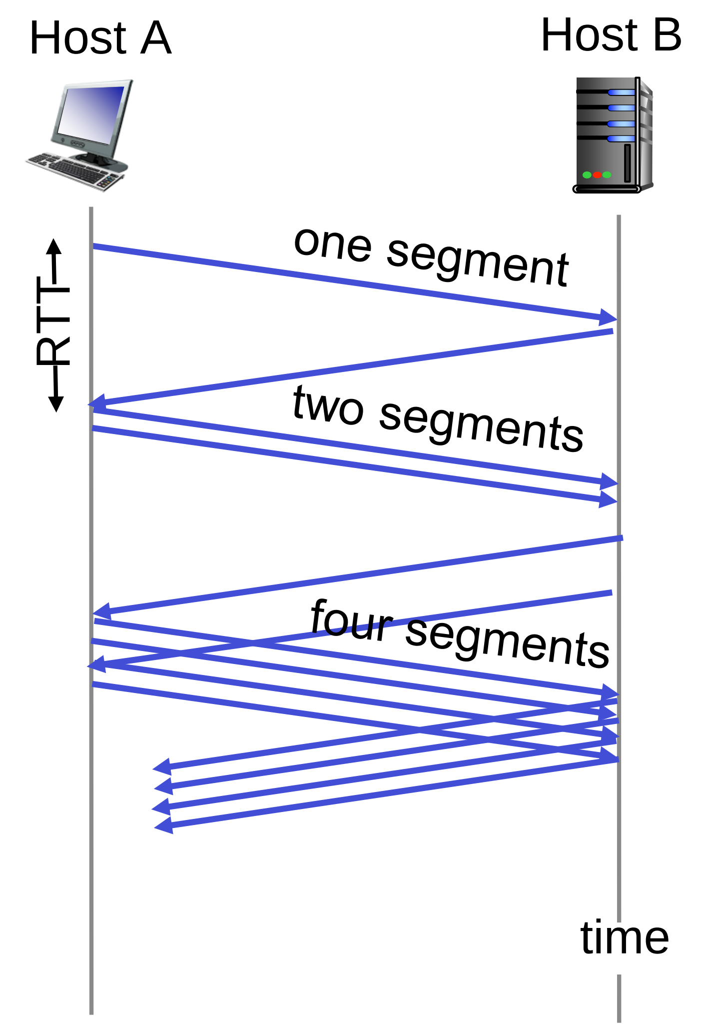
$$\boxed{\texttt{LastByteSent - LastByteAcked} \leq \texttt{min\{cwnd, rwnd\}}}$$

TCP Congestion-Control Algorithm

- **TCPs congestion control algorithm contain three main components**
  - Slow-start
  - Congestion Avoidance
  - Fast Recovery

TCP Slow-Start

- **When connection begins, increase rate exponentially until first loss event:**
  - Initially, **`cwnd`** = 1 MSS
  - Double **`cwnd`** every RTT
    - Done by incrementing **`cwnd`** for every ACK received
  - Initial rate is slow but ramps up exponentially fast
- **When first loss occurs, store (`.5*cwnd`) as `SSThresh` and restart slow-start**
- **When `cwnd` reaches `SSThresh`, switch from slow-start mode to congestion avoidance mode**

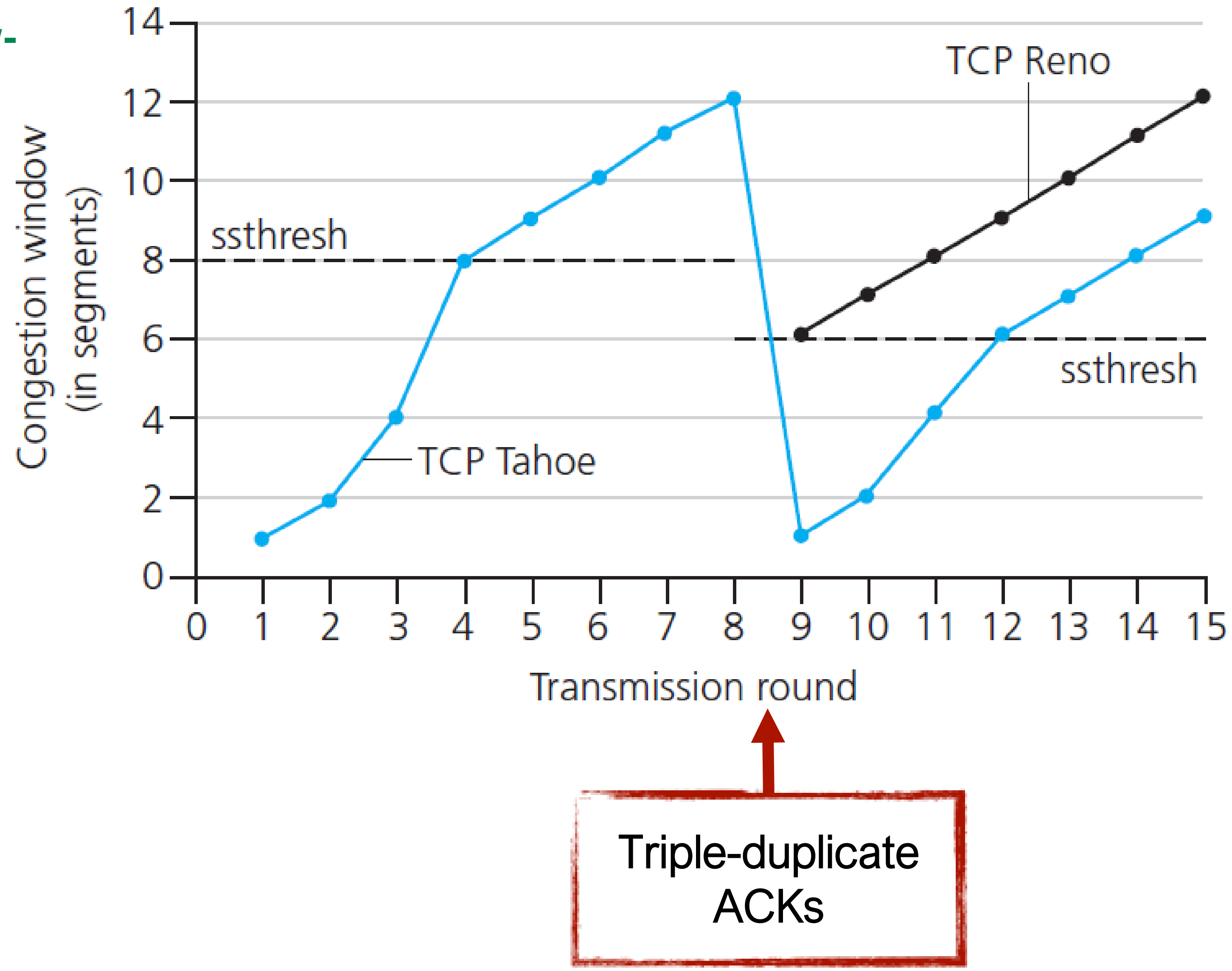


Loss During Slow-Start

- **If loss is indicated by a timeout**
  - Store (**`.5*cwnd`**) as **`SSThresh`** and restart slow-start
  - Set **`cwnd`** set to 1 MSS
  - **`cwnd`** then grows exponentially (as in slow start) to threshold value **`SSThresh`**, then grows linearly in congestion avoidance phase
- **If loss is indicated by 3 duplicate ACKs (only in TCP Reno)**
  - Duplicate ACKs indicate network capable of delivering some segments, so don't drop **`cwnd`** all the way down to 1 MSS
  - Store **`(.5*cwnd)`** as **`SSThresh`**
  - **`cwnd`** is also set to **`(.5*cwnd)`**, but will be increment for each duplicate ACK
- **TCP Tahoe always sets `cwnd` to 1 (for either timeout or 3 duplicate acks)**

Switch from Slow Start to Congestion Avoidance

- **Exponential growth phase shows TCP slow-start**
- **Linear phase after crossing over `SSThresh` shows the congestion avoidance phase**
- **TCP Tahoe**
  - Set **`cwnd = 1`** for both a timeout and for triple duplicate ACKs
  - Set **`SSThresh = cwnd/2`**
  - Re-enters slow-start phase
- **TCP Reno**
  - Implements Fast Recovery
  - Retransmits missing segment
  - Set **`SSThresh = cwnd/2`**
  - Set **`cwnd = SSThresh + 3`**
  - In congestion avoidance phase



Chapter 3: summary

- **principles behind transport layer services:**
  - multiplexing, demultiplexing
  - reliable data transfer
  - flow control
  - congestion control
- **instantiation, implementation in the Internet**
  - UDP
  - TCP

next:

- **leaving the network "edge" (application, transport layers)**
- **into the network "core"**
- **two network layer chapters:**
  - data plane
  - control plane