

Lecture 22:

TCP CC, Links and Signaling

CSE 123: Computer Networks
Alex C. Snoeren



Lecture 22 Overview



- TCP Congestion Control
- Signaling
 - ♦ Channel characteristics
 - ♦ Types of physical media
- A lot of this material is **not in the book**
 - ♦ **Caveat:** I am not an EE Professor

TCP Bandwidth Probing



- TCP uses AIMD to perform congestion avoidance
 - ◆ Converges to fair share of bottleneck link bandwidth (BW)
 - ◆ Increases congestion window (one MSS per RTT) modestly in good times
 - ◆ Cuts drastically (by half) in bad times
- But what rate should a TCP flow use initially?
 - ◆ Need some initial congestion window
 - ◆ We'd like to TCP to work on all manner of links
 - ◆ Need to span 7+ orders of magnitude, e.g., 10 K to 100 Gbps.
 - ◆ Starting too fast is catastrophic!

**How long would it take
TCP to reach full speed
with AIMD?**

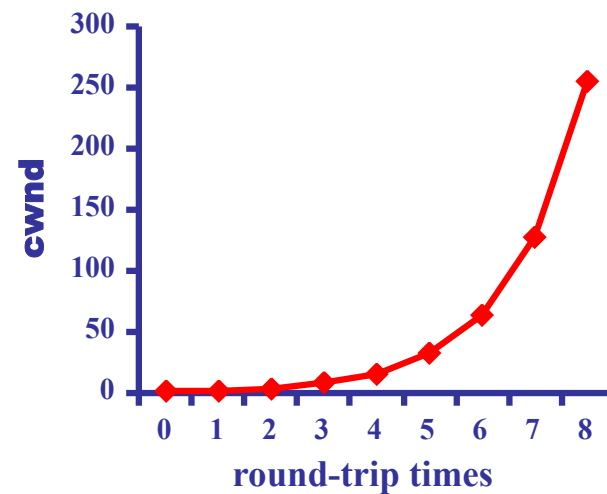
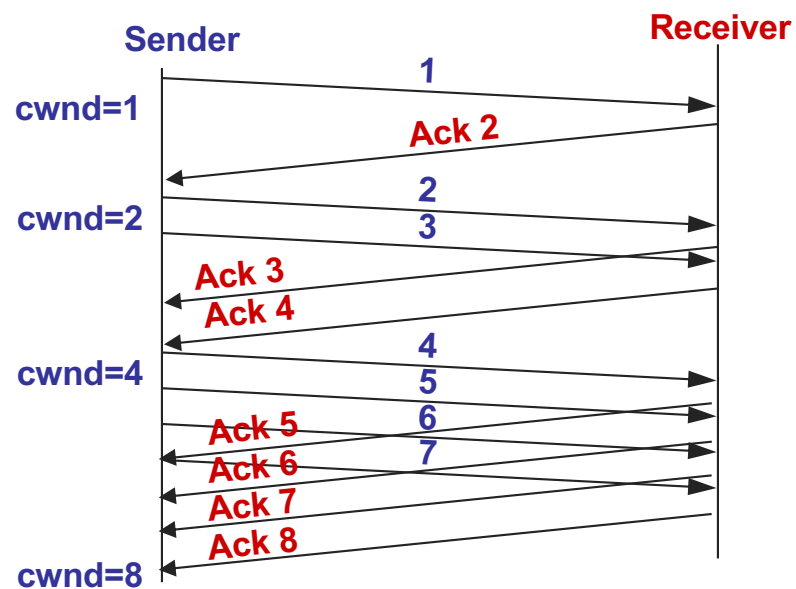
- A. RTT**
- B. BW/RTT**
- C. $BW/MSS * RTT$**
- D. $(BW*RTT)/MSS * RTT$**

Slow Start



- Goal: quickly find the equilibrium sending rate
- Quickly increase sending rate until congestion detected
 - ♦ Remember last rate that worked and don't overshoot it
- TCP Reno's Slow-Start Algorithm:
 - ♦ On new connection, or after timeout, set *cwnd*=1 MSS
 - ♦ For each segment acknowledged, increment *cwnd* by 1 MSS
 - ♦ If timeout then divide *cwnd* by 2, and set *ssthresh* = *cwnd*
 - ♦ If *cwnd* $\geq$ *ssthresh* then exit slow start
- Why called slow? Its exponential after all...

Slow Start Example



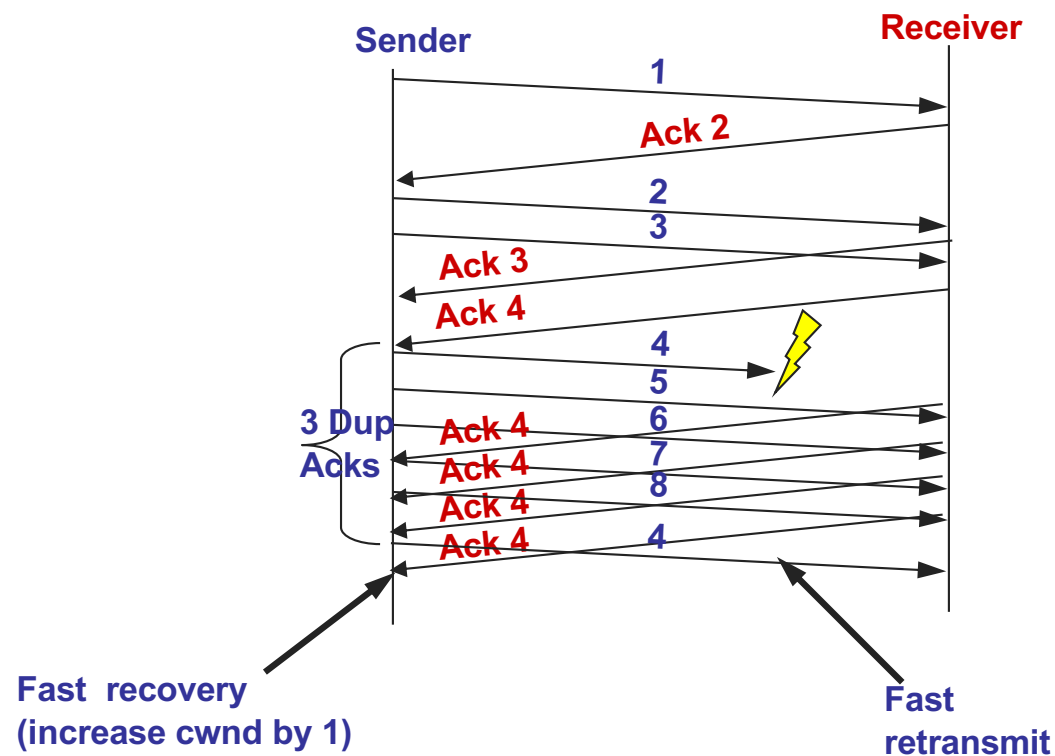




Fast Retransmit & Recovery

- Fast retransmit
 - ♦ Timeouts are slow (default often 200 ms or 1 second)
 - ♦ When packet is lost, receiver still ACKs last in-order packet
 - ♦ Use 3 duplicate ACKs to indicate a loss; detect losses quickly
 - » Why 3? When wouldn't this work?
- Fast recovery
 - ♦ Goal: avoid stalling after loss
 - ♦ If there are still ACKs coming in, then no need for slow start
 - ♦ If a packet has made it through -> we can send another one
 - ♦ Divide *cwnd* by 2 after fast retransmit
 - ♦ Increment *cwnd* by 1 MSS for each additional duplicate ACK

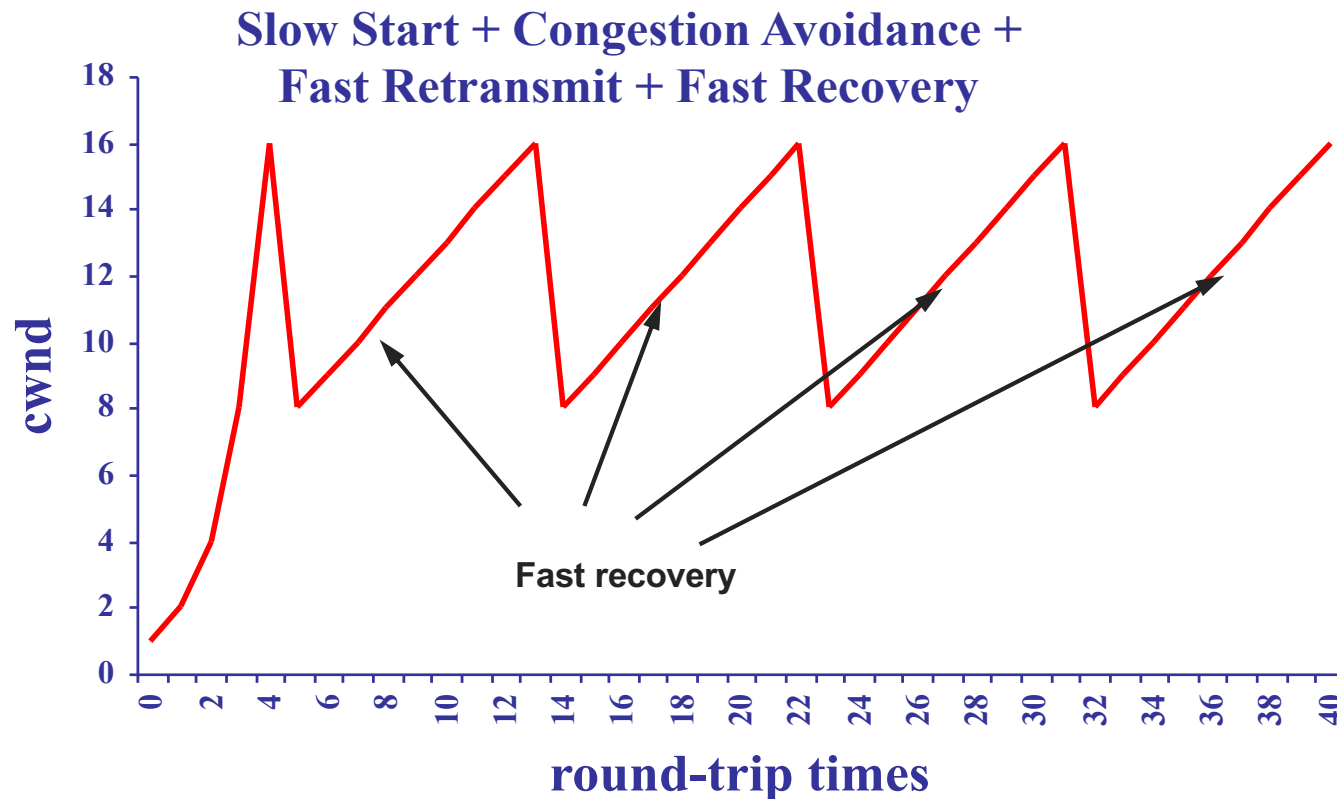
Fast Retransmit Example



What segment will be sent next?

- A. 4
- B. 5
- C. 8
- D. 9

More Sophistication



Short Connections



- Short connection: only contains a few pkts
- How do short connections and Slow-Start interact?
 - ♦ What happens when a packet is lost during Slow-Start?
 - ♦ What happens when the SYN is dropped?
- Bottom line: Which packet gets dropped matters a lot
 - ♦ SYN
 - ♦ Slow-Start
 - ♦ Congestion avoidance

**Are most flows on the Internet long or short?
Are most bytes on the Internet in long or short flows?**

- A. Long / Long**
- B. Long / Short**
- C. Short / Long**
- D. Short / Short**

Open Issues



- TCP is designed around the premise of cooperation
 - ♦ What happens to TCP if it competes with a UDP flow?
 - ♦ What if we divide *cwnd* by 3 instead of 2 after a loss?
- There are a bunch of magic numbers
 - ♦ Decrease by 2x, increase by $1/cwnd$, 3 duplicate acks, initial timeout = 3 seconds, etc.
- But overall it works surprisingly well!
 - ♦ Still being constantly tweaked...

TCP CC Summary

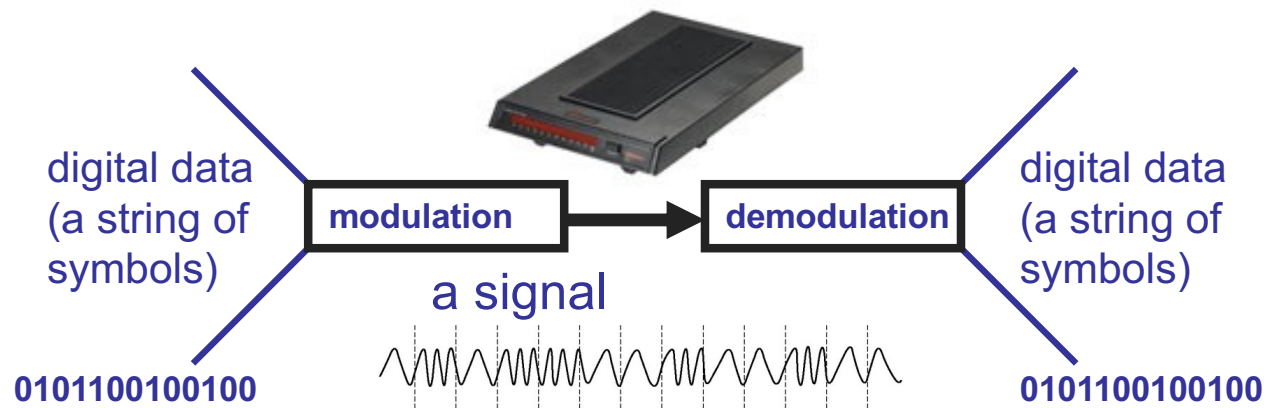


- TCP Probes the network for bandwidth, assuming that loss signals congestion
- The congestion window is managed with an additive increase/multiplicative decrease policy
 - ◆ It took fast retransmit and fast recovery to get there
 - ◆ Fast recovery keeps pipe “full” while recovering from a loss
- Slow start is used to avoid lengthy initial delays
 - ◆ Ramp up to near target rate, then switch to AIMD

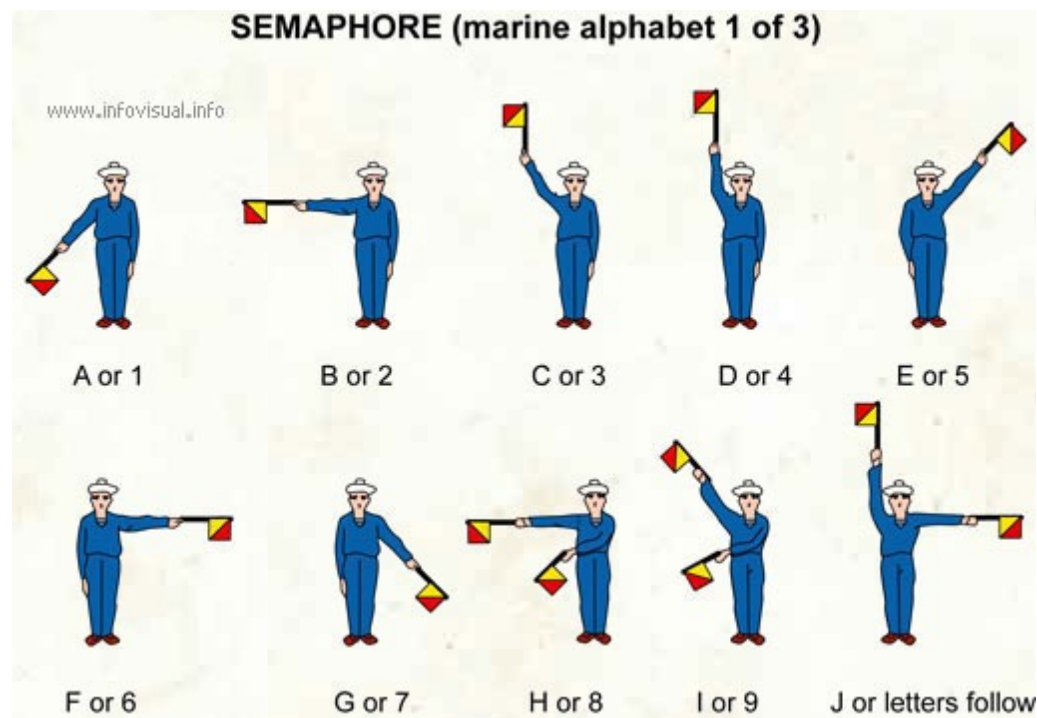


Underneath it all: Sending bits

- A three-step process
 - ◆ Take an input stream of bits (digital data)
 - ◆ Modulate some physical media to send data (analog)
 - ◆ Demodulate the signal to retrieve bits (digital again)
 - ◆ Anybody heard of a **modem** (Modulator-demodulator)?



A Simple Signaling System



Morse Code



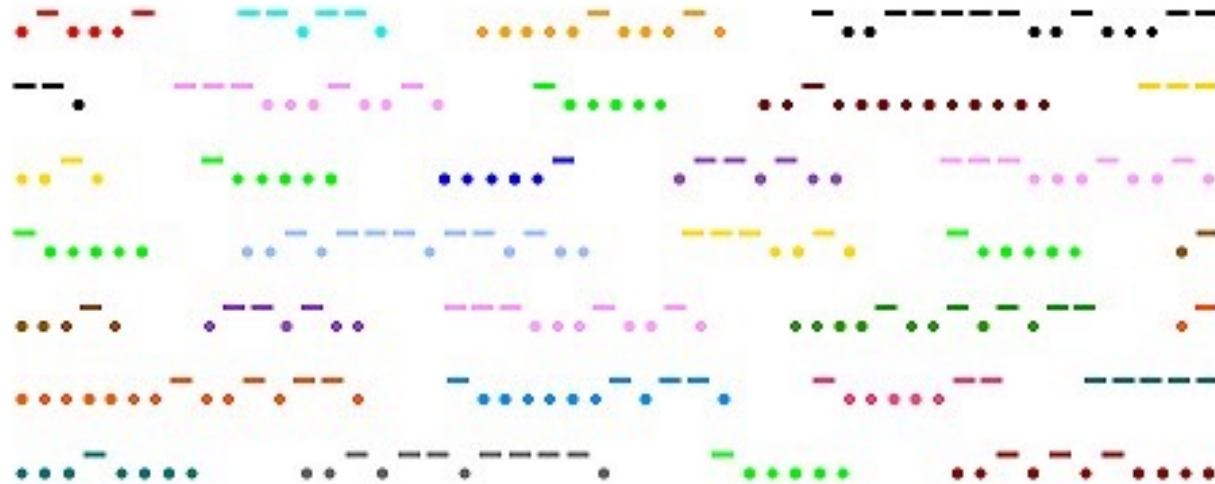
A	• —	V	• • • —
B	— • • •	W	— — •
C	— • — •	X	— • • —
D	— • •	Y	— • — —
E	•	Z	— — • •
F	• • — •	,	• — • — • —
G	— — •	.	— — • • — —
H	• • • •	?	• — — • • •
I	• •	/	— • • — •
J	• — — —	@	• — — • • •
K	— • —	1	• — — — —
L	• — • •	2	• • — — —
M	— —	3	• • • — —
N	— •	4	• • • • —
O	— — —	5	• • • • •
P	• — — •	6	— • • • •
Q	— — • —	7	— — — • •
R	• — • •	8	— — — — •
S	• • •	9	— — — — •
T	—	0	— — — — —
U	• • —		



Morse Code Message



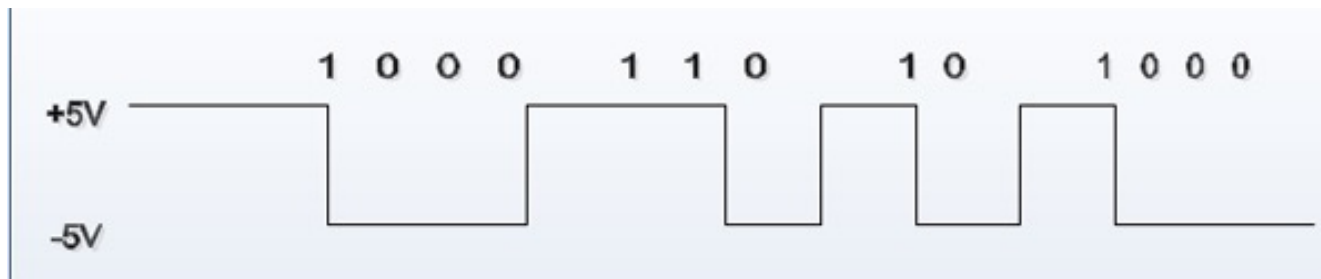
Let man have dominion over the fish of the sea and over the fowl of the
air and over every living thing that moves upon the earth



Binary signaling with Voltage



- Encode 1's and 0's on a wire
 - ◆ +5 volts = 1
 - ◆ -5 volts = 0



Signals and Channels

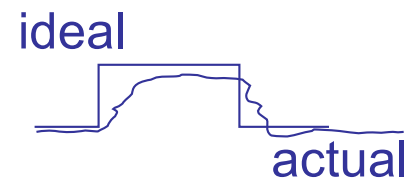
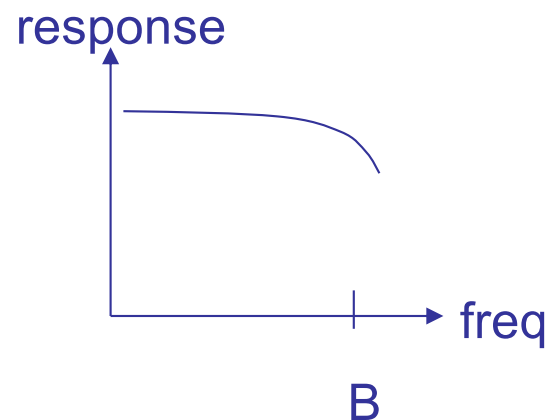


- A **signal** is some form of energy (light, voltage, etc)
 - ◆ Varies with time (on/off, high/low, etc.)
 - ◆ Can be continuous or discrete
- A **channel** is a physical medium that conveys energy
 - ◆ Any real channel will distort the input signal as it does so
 - ◆ How it distorts the signal depends on the signal

Channel Challenges



- Every channel degrades a signal
 - ♦ Distortion impacts how the receiver will interpret signal



For Next Time



- Read 2.2 in P&D
- Homework 4 due Friday
- Project 2b due next Monday