Chapter 3 outline

- 3.1 Transport-layer services
- 3.2 Multiplexing and demultiplexing
- 3.3 Connectionless transport: UDP
- 3.4 Principles of reliable data transfer

- 3.5 Connection-oriented transport: TCP
  - segment structure
  - reliable data transfer
  - flow control
  - connection management

- 3.6 Principles of congestion control
- 3.7 TCP congestion control
TCP Congestion Control

- end-end control (no network assistance)
- sender limits transmission:
  \[ \text{LastByteSent} - \text{LastByteAcked} \leq \text{CongWin} \]
- Roughly,
  \[ \text{rate} = \frac{\text{CongWin}}{\text{RTT}} \text{ Bytes/sec} \]
- \text{CongWin} is dynamic, function of perceived network congestion

**How does sender perceive congestion?**

- loss event = timeout or 3 duplicate acks
- TCP sender reduces rate \((\text{CongWin})\) after loss event

**Three mechanisms:**
- AIMD
- slow start
- conservative after timeout events
TCP AIMD

**multiplicative decrease:**
cut CongWin in half after loss event

**additive increase:**
increase CongWin by 1 MSS every RTT in the absence of loss events:
probing

Long-lived TCP connection
TCP Slow Start

- When connection begins, $\text{CongWin} = 1 \text{ MSS}$
  - Example: MSS = 500 bytes & RTT = 200 msec
  - initial rate = 20 kbps
- available bandwidth may be >> MSS/RTT
  - desirable to quickly ramp up to respectable rate

- When connection begins, increase rate exponentially fast until first loss event
TCP Slow Start (more)

- When connection begins, increase rate exponentially until first loss event:
  - double CongWin every RTT
  - done by incrementing CongWin for every ACK received

- Summary: initial rate is slow but ramps up exponentially fast
Refinement

- After 3 dup ACKs:
  - CongWin is cut in half
  - window then grows linearly
- But after timeout event:
  - CongWin instead set to 1 MSS;
  - window then grows exponentially
  - to a threshold, then grows linearly

Philosophy:

- 3 dup ACKs indicates network capable of delivering some segments
- timeout before 3 dup ACKs is “more alarming”
Refinement (more)

Q: When should the exponential increase switch to linear?
A: When CongWin gets to 1/2 of its value before timeout.

Implementation:
- Variable Threshold
- At loss event, Threshold is set to 1/2 of CongWin just before loss event
Summary: TCP Congestion Control

- When $\text{CongWin}$ is below $\text{Threshold}$, sender in slow-start phase, window grows exponentially.

- When $\text{CongWin}$ is above $\text{Threshold}$, sender is in congestion-avoidance phase, window grows linearly.

- When a triple duplicate ACK occurs, Threshold set to $\text{CongWin}/2$ and $\text{CongWin}$ set to Threshold.

- When timeout occurs, Threshold set to $\text{CongWin}/2$ and $\text{CongWin}$ is set to 1 MSS.