CPSC 619-600
Networks and Distributed Processing
Spring 2005

Congestion Control II

Dmitri Loguinov
Texas A&M University

February 8, 2005

Chapter 3: Roadmap

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

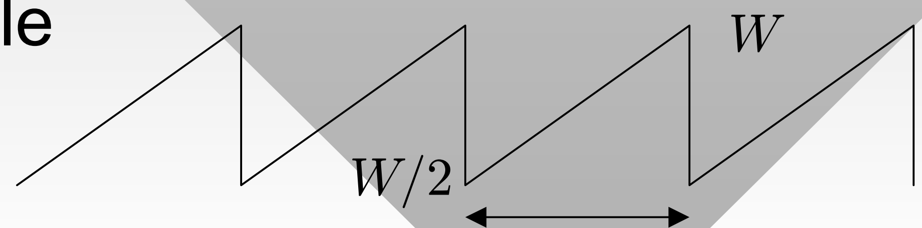
TCP Throughput

- What's the average throughout ot TCP as a function of window size W and RTT ?
  - Ignore slow start and assume perfect AIMD (no timeouts)
- Let W be the window size when loss occurs
- When window is W, throughput is $W * MSS/RTT$
- Just after loss, window drops to $W/2$, throughput to $W * MSS/(2RTT)$
- Average rate: $3/4$ * $W * MSS/RTT$

$$r_{av} = \frac{3}{4} \times \frac{W \times MSS}{RTT} = \frac{W_{av} \times MSS}{RTT}$$

TCP Model

- Example: 1500-byte segments, 100 ms RTT, want 10 gb/s average throughput
  - Requires window size $W = 111{,}111$ in-flight segments ($W_{av} = 83{,}333$ packets)
  - This is huge in terms of buffer space
- Next: derive average throughput in terms of loss rate
  - Assume packet loss probability is p
  - This means that one packet is lost out of every $1/p$ packets
- Step 1: derive the number of packets transmitted in one oscillation cycle



TCP Model

- Assume the window just before the loss is W
  - Then it is $W/2$ right after the loss
- The number of packets sent between two losses:

$$sent = W/2 + (W/2+1) + (W/2+2) + \ldots + W$$

- The above formula includes window size every RTT
- Q: How many terms in the summation?
- A: $W/2+1$

$$sent = W/2(W/2+1) + \sum_{i=1}^{W/2} i$$

TCP Model

- Thus we arrive at:

$$sent = \frac{3}{8}W^2 + \frac{3}{4}W$$

- Step 2: now notice that this number equals $1/p$
- Ignoring the linear term, we approximately get:

$$\frac{1}{p} \approx \frac{3}{8}W^2$$

- In other words:

$$W = \sqrt{\frac{8}{3p}}$$

TCP Model

- Step 3: writing in terms of average rate:

$$r_{av} = \frac{W_{av} \times MSS}{RTT} = \frac{\frac{3}{4}W \times MSS}{RTT} = \frac{\frac{3}{4}\sqrt{\frac{8}{3p}} \times MSS}{RTT}$$

- Simplifying:

$$r_{av} = \frac{\sqrt{3/2} \times MSS}{RTT\sqrt{p}} \approx \frac{1.22 \times MSS}{RTT\sqrt{p}}$$

- This is the famous formula of TCP throughput

TCP Model

- Q: What is the max packet loss rate allowed if we want to sustain 10 gb/s average throughput?
- A: $p = 2.1{\times}10^{-10}$ wow!
  - Such low rates are not likely (even corruption occurs more frequently in many networks)
- Q: In congestion avoidance, how long does it take TCP to go from 5 gb/s to 10 gb/s?
- Solution: at 5 gb/s, the window size is 41,666 pkts and at 10 gb/s it is 83,333 pkts
  - Then, we need $83{,}333 - 41{,}666$ RTTs to close this gap
  - This is $4{,}100 \text{ seconds} = 1.15$ hours

TCP Future

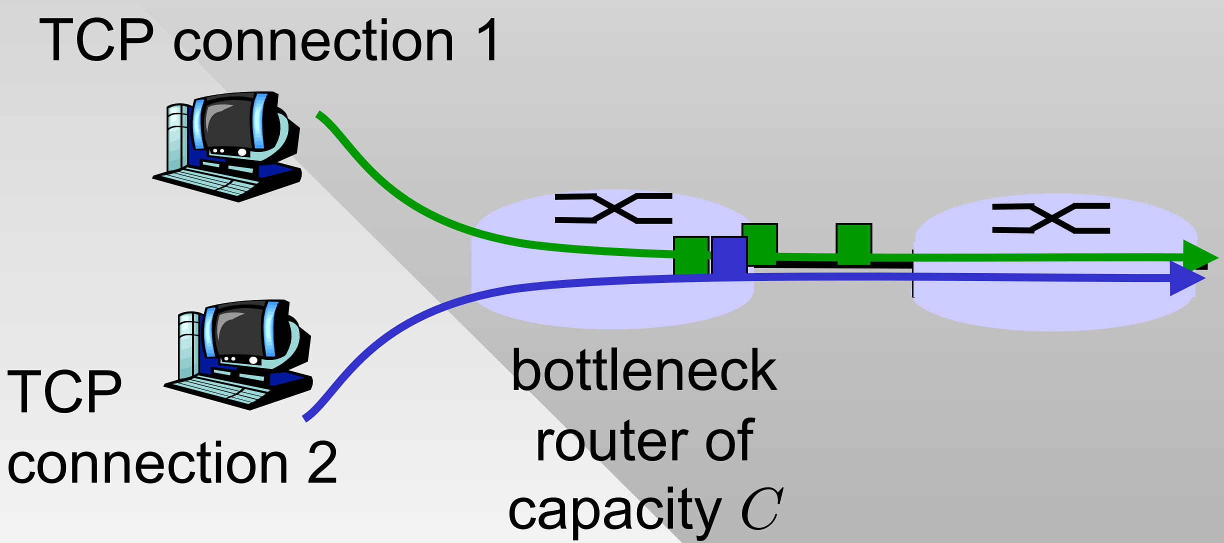
- TCP is slow, but what if most transfers are short?
  - How long before TCP reaches 10 gb/s in slow start?
- Idea: starting at $W = 1$ we need to reach $W = 83{,}333$ pkts at an exponential rate
- The time need to reach full capacity is $\text{ceil}(\log_2(83333))*RTT = 1.7$ seconds (17 steps)!
- How much data can we squeeze in slow start?

$$\text{pkts sent} = 1 + 2 + 4 + 8 + \ldots + 2^{17} = 2^{18}$$

- Total data transmitted ≈ 39.3 MB
  - Conclusion: short connects are perfectly fine

TCP Fairness

Fairness goal: if K TCP sessions share same bottleneck link of bandwidth C, each should have average rate of C/K



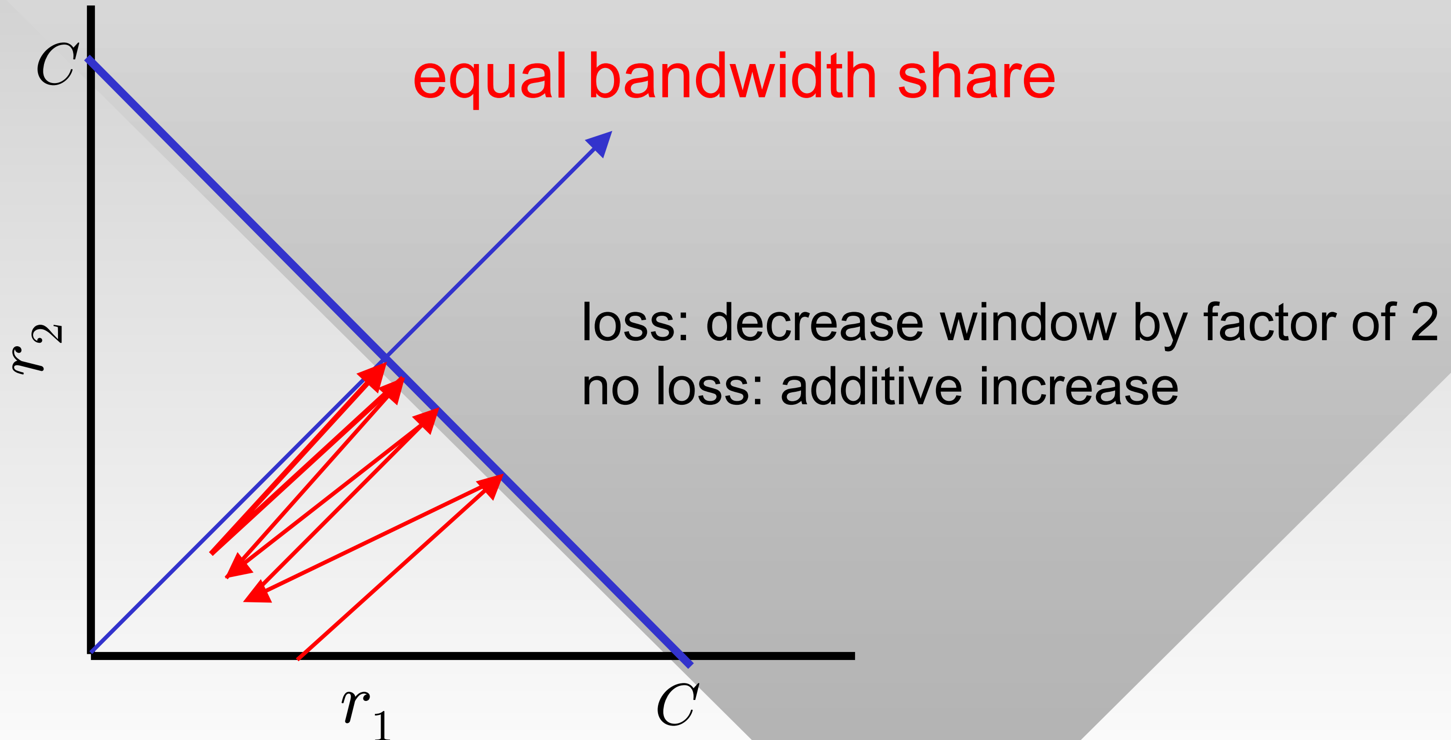
Why Is TCP Fair?

Consider any two competing sessions

- Assume one flow sends slower $r_1(0) < r_2(0)$:
  - First consider additive increase step
  - Old flow rates $(r_1(n), r_2(n))$, new rates $(r_1(n)+\alpha, r_2(n)+\alpha)$
  - Here, $\alpha = MSS/RTT$
- Multiplicative decrease reduces throughput proportionally
  - Old flow rates $(r_1(n), r_2(n))$, new rates $(r_1(n)/2, r_2(n)/2)$
- Assume we define fairness as $r_1(n)/r_2(n)$
  - Fairness of 1 is ideal since the rates are equal
- Q: How does fairness change during increase and decrease?

Why Is TCP Fair?

- A: fairness stays the same during decrease and improves during increase



Fairness (More)

Fairness and UDP

- Multimedia apps often do not use TCP
  - Do not want rate throttled by congestion control
- Instead use UDP:
  - Pump audio/video at constant rate, tolerate packet loss
- Research area: TCP friendly

Fairness and parallel TCP connections

- Nothing prevents app from opening parallel flows between 2 hosts
- Web browsers do this
- Example: link of rate C with 9 flows present:
  - New app asks for 1 TCP, gets rate $C/10$
  - New app asks for 11 TCPs, gets $C/2$!