TCP Performance

- Effective over a wide range of capacities
- A lot of operational experience

$$\lambda_{TCP} = F(p, RTT)$$

$\frac{W}{RTT}$ sending rate (throughput)

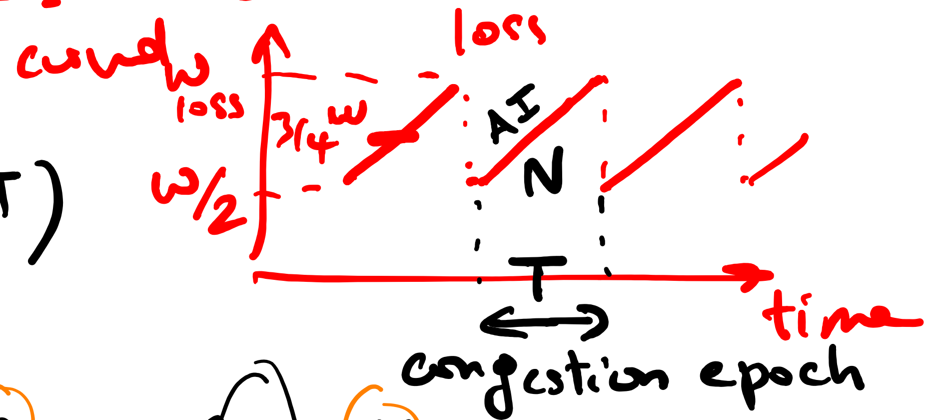
prob. of segment lost

- Periodic loss (macroscopic) model shows that throughput is inversely proportional to
  - square root of loss probability p
  - RTT
  - average sending rate = sqrt(1.5/p) / RTT

- Reno
- steady-state, static per RTT
- low/moderate congestion $\Rightarrow$ AIMD

$\lambda = ? = F(p, RTT)$

- periodic loss model (losing one segment every T)



$$\lambda = \frac{N}{T}$$

$$N = \frac{w}{2} + \left(\frac{w}{2}+1\right) + \left(\frac{w}{2}+2\right) + \dots + \left(\frac{w}{2}+\frac{w}{2}\right)$$

$1+2+3+\dots+x = \frac{x(x+1)}{2}$

$$= \frac{w}{2}\left(\frac{w}{2}+1\right) + \frac{\frac{w}{2}\left(\frac{w}{2}+1\right)}{2}$$

$$= \frac{w^2}{4} + \frac{w}{2} + \frac{w^2}{8} + \frac{w}{4} = \frac{3}{8}w^2 + \frac{3}{4}w = \frac{3}{4}w\left(\frac{w}{2}+1\right) \quad G(p)\ ?$$

$$T = \left(\frac{w}{2}+1\right)RTT \Rightarrow \lambda = \frac{N}{T} = \frac{\frac{3}{4}w\cancel{\left(\frac{w}{2}+1\right)}}{\cancel{\left(\frac{w}{2}+1\right)}RTT} = \boxed{\frac{3/4\, w}{RTT}}$$

$$N = \frac{3}{8}w^2 + \cancel{\frac{3}{4}w} \approx \frac{3}{8}w^2 \quad (w \gg 1)$$

$$p = \frac{1}{N} = 1 \Big/ \left(\frac{3}{8}w^2\right) \Rightarrow w^2 = \frac{8/3}{p} \Rightarrow w = \frac{\sqrt{8/3}}{\sqrt{p}}$$

$$\lambda = \frac{3/4\sqrt{8/3}}{\sqrt{p}\cdot RTT}$$

$$\lambda = \frac{\sqrt{3/2}}{\sqrt{p} \times RTT}$$

TCP Futures: TCP over "long, fat pipes"

$T = \left(\frac{W}{2} + 1\right) RTT$ $\quad B \times D \uparrow$

- Example: 10,000-bit segments, 100ms RTT, want 10 Gbps throughput

$W_{loss} \geq W_{ideal} = B \times D = \frac{\cancel{10}G \, (= 10^9) \times \cancel{100} \cancel{ms}}{10{,}000 = 10^4} = 10^5 = 100{,}000$ segments

$ms = 10^{-3}$

$\Rightarrow T = \left(\frac{W_{loss}}{2} + 1\right) RTT \geq \left(\frac{150{,}000}{2} + 1\right) * 100\,ms = 5{,}000$ seconds > 1 hour

TCP Futures: TCP over "long, fat pipes"

- Example: 10,000-bit segments, 100ms RTT, want 10 Gbps throughput

- Requires window size W = 100,000 in-flight segments
- Throughput in terms of loss rate:

$$10\text{Gbps} = \lambda = \frac{1.22 \cdot MSS}{RTT\sqrt{p}}$$

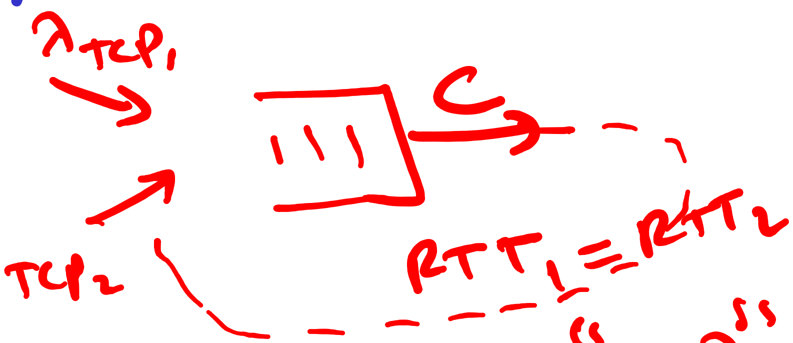
$\leftarrow 10{,}000$

$\leftarrow 100\text{ms}$

- p = 1.5 x 10^{-10} *Wow*
- New versions of TCP for high-speed

Why is TCP fair?

Reno AIMD



$\lambda = \frac{w}{RTT}$

AI: $w/R \leftarrow w/R + 1/R$ every RTT

MD: $w/R \leftarrow \frac{w}{2}/R$ upon loss

FAIRNESS: $\lambda_1^{ss} = \lambda_2^{ss}$

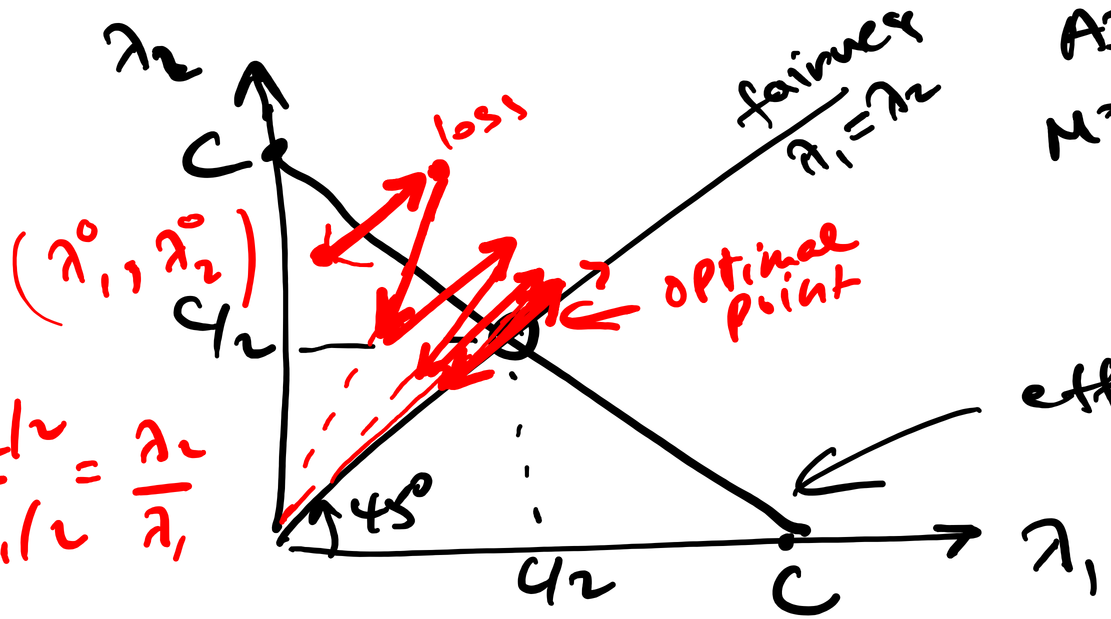
EFFICIENCY: $\lambda_1^{ss} + \lambda_2^{ss} = C$

$\left.\right\}$ OPTIMAL OPERATING POINT

AI: $\lambda \leftarrow \lambda + \frac{1}{RTT}$

MD: $\lambda \leftarrow \lambda/2$

synchronized update model



efficiency line

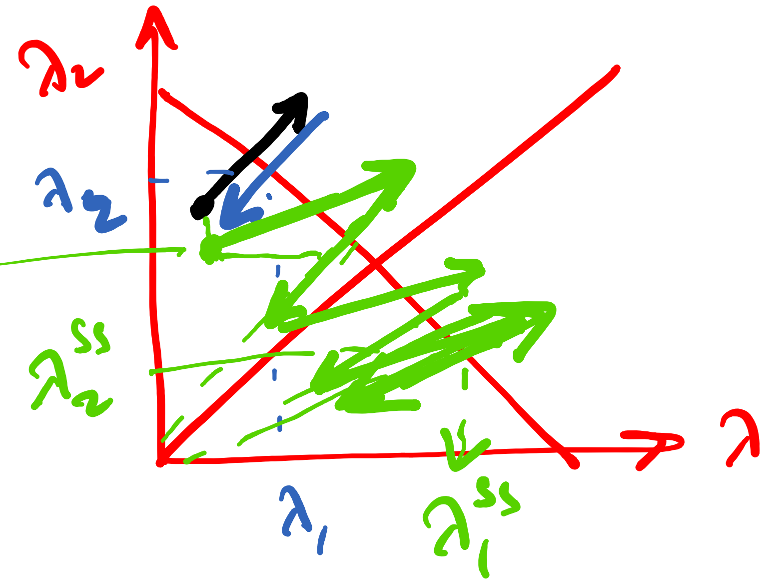
$\lambda_1 + \lambda_2 = C$

$\frac{\lambda_2/2}{\lambda_1/2} = \frac{\lambda_2}{\lambda_1}$

Why is TCP fair?

AIAD

$\lambda_1^0 < \lambda_2^0$



AIMD

$RTT_1 < RTT_2$

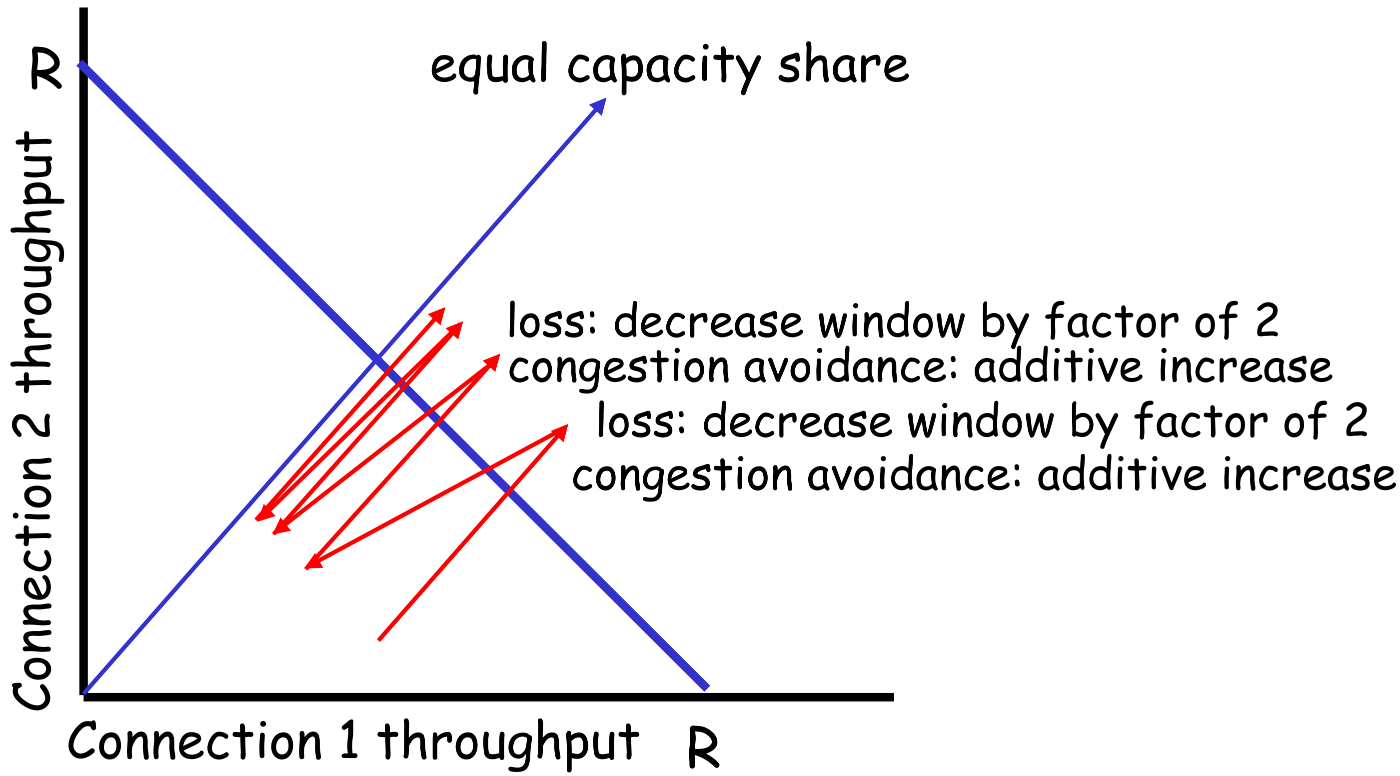
AI: $\lambda \leftarrow \lambda + \frac{1}{RTT} \Rightarrow \lambda_1^{ss} > \lambda_2^{ss}$

$$\lambda \propto \frac{1}{\sqrt{p} \cdot RTT}$$

Why is TCP fair?

Two competing sessions (with same RTT):

- Additive increase gives slope of 1, as throughout increases
- multiplicative decrease decreases throughput proportionally



Delay modeling

Q: How long does it take to receive an object from a Web server after sending a request?

Ignoring congestion, delay is influenced by:

- TCP connection establishment
- data transmission delay
- slow start

Notation, assumptions:

- Assume one link between client and server of rate C
- S: MSS (bits)
- O: object size (bits)
- no retransmissions (no loss, no corruption)

TCP Delay Modeling: Slow Start

Delay components:

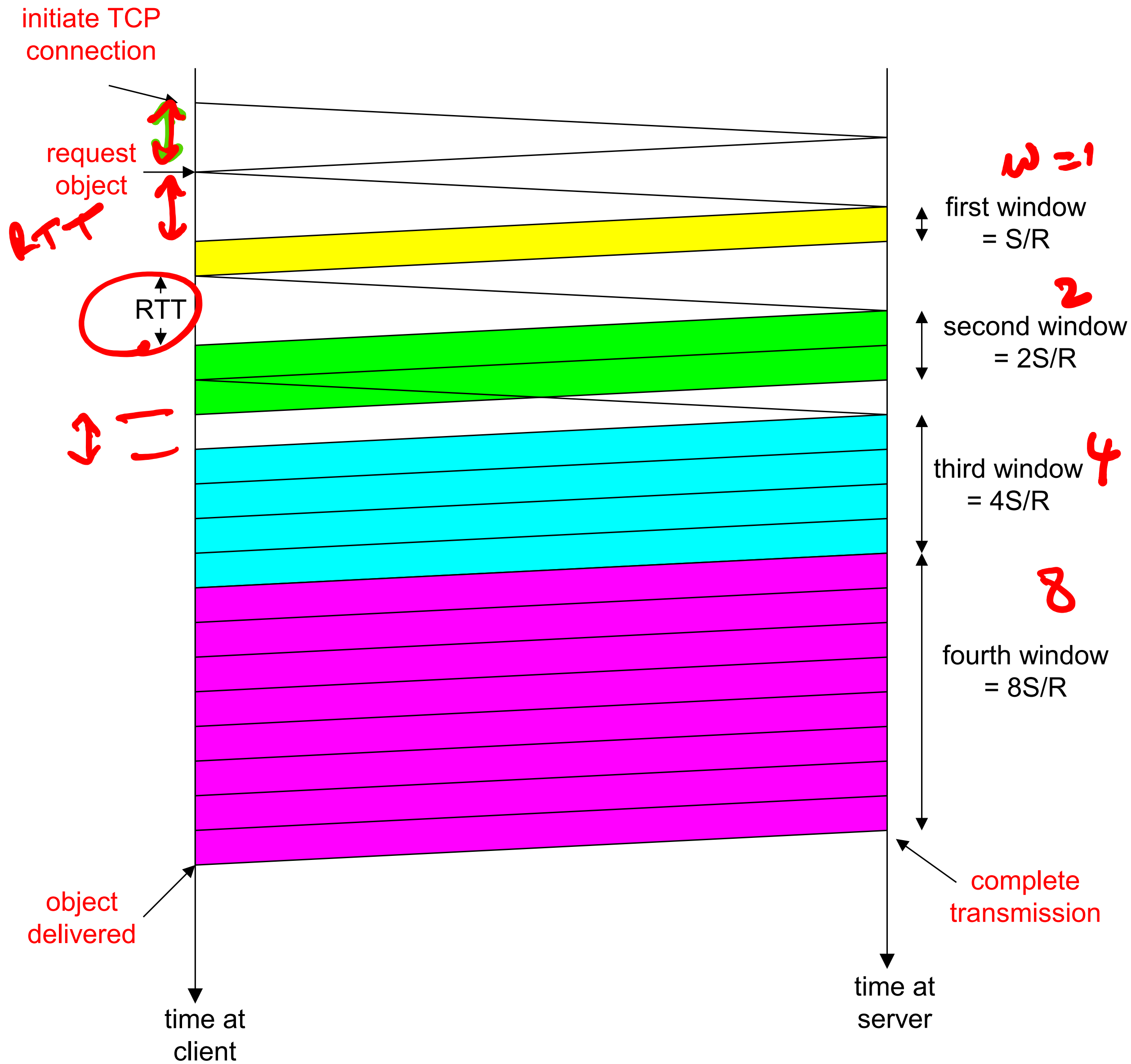
- 2 RTT for connection estab and request
- O/C to transmit object

[RTT - 2 S/C]

Example:

- O/S = 15 segments

Server idles 2 times due to slow start



Chapter 3: Summary

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

Next:

- leaving the network "edge" (application, transport layers)
- into the network "core"