CSCE 463/612
Networks and Distributed Processing
Fall 2024

Transport Layer VI

Dmitri Loguinov
Texas A&M University

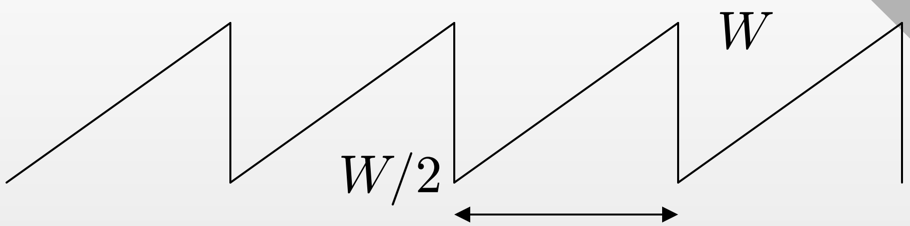
October 24, 2024

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

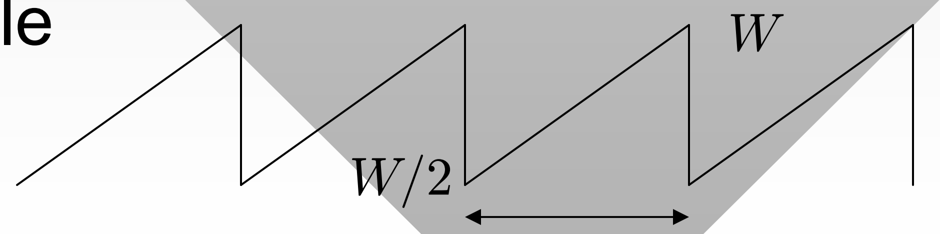


- What's the average throughout of TCP as a function of max window size W and RTT?
  - Ignore slow start and assume perfect AIMD (no timeouts)
- Let W be the window size when loss occurs
  - At that time, throughput is $W * MSS/RTT$
  - Just after loss, window drops to $W/2$, throughput is halved
- Average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

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

TCP Model

- Example: 1500-byte segments, 100 ms RTT, want 10 Gbps average throughput r_{av}
  - Requires max window size $W = 111{,}111$ in-flight segments, 166 MB of buffer space ($W_{av} = 83{,}333$ packets)
  - But there are bigger issues as discussed below
- Next: derive average throughput in terms of loss rate
  - Assume packet loss probability is p
  - Roughly one packet lost for every $1/p$ sent packets
- Step 1: derive the number of packets transmitted in one oscillation cycle



TCP Model

- Examine time in terms of RTT units
  - At each step, window increases by 1 packet
- The number of packets sent between two losses:

$$sent = \frac{W}{2} + \left(\frac{W}{2} + 1\right) + \left(\frac{W}{2} + 2\right) + \ldots + W$$

- Combining $W/2$ terms, we have:

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

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 AIMD throughput
  - Note: homework #3 does not use congestion control and its rate is a different function of p

TCP Model (Discussion)

$$\frac{1.22 \times MSS}{RTT\sqrt{p}}$$

- Example: What is the required packet loss for 100-ms RTT, 1500-byte MSS, and 10 Gbps average rate?
  - Turns out, $p = 2.1{\times}10^{-10}$, which is almost impossible (even in wired networks corruption occurs more frequently)
  - Backbone loss $p = 10^{-4}$ (and even 10^{-3}) is considered great
- Example: In AIMD, how long does it take for TCP to go from 5 Gbps to 10 Gbps?
  - Window must grow from 41,666 pkts to 83,333
  - TCP needs (83,333 - 41,666) RTTs to close this gap
  - This is 4,166 seconds = 1 hour 9 minutes
- Over long-distance links (RTT > 50 ms), AIMD typically maxes out around 200 Mbps

TCP Future

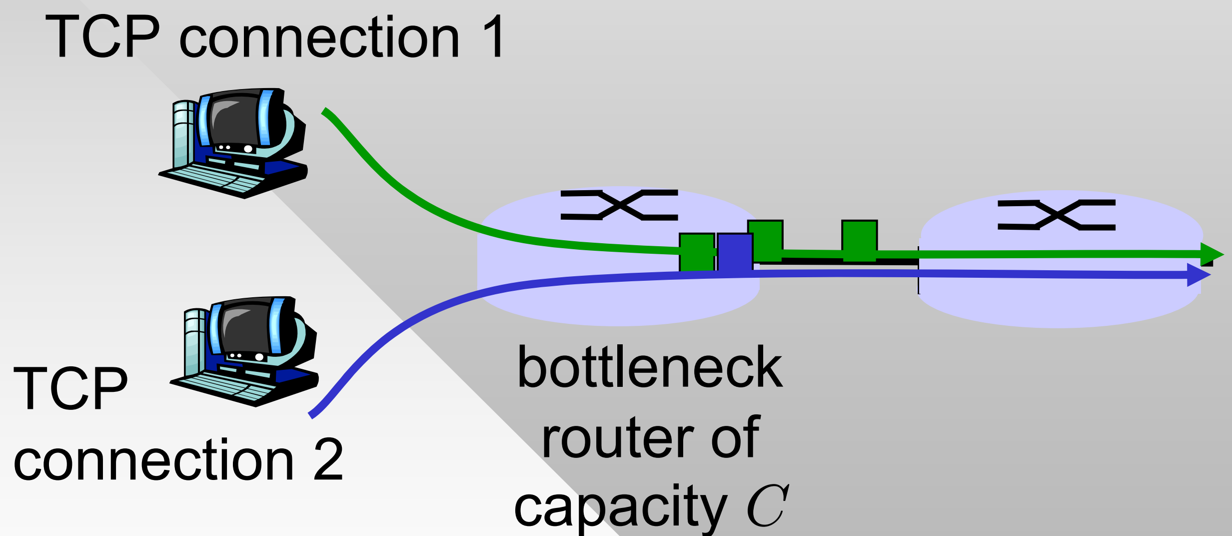
- TCP is slow, but what if most transfers are short?
  - How long before TCP reaches 10 Gbps in slow start?
- Idea: starting at $W = 1$ we need to hit $W = 83{,}333$ pkts, doubling the window each RTT
- The time needed to reach full capacity is $\mathrm{ceil}(\log_2(83333))^*RTT = 1.7$ seconds (17 steps)!
- How much data can we squeeze in slow start?

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

- Total data transmitted (pkt size 1500) ≈ 393 MB
  - Conclusion: short connects are fine with original TCP

TCP Fairness

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

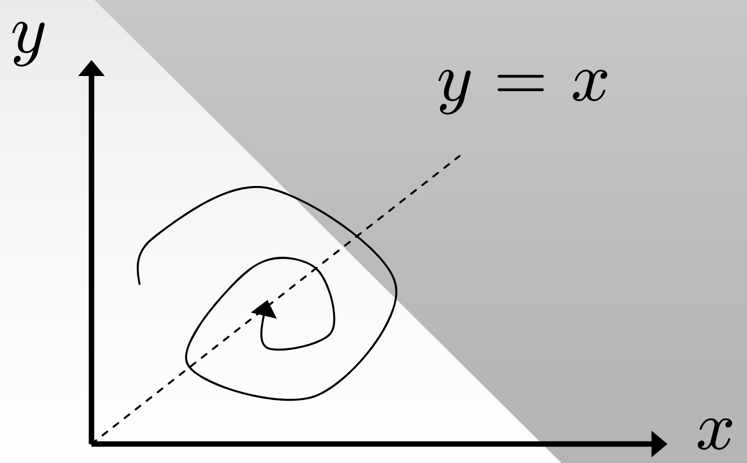


Fairness index of two flows:

$$\Phi = \frac{\min(x, y)}{\max(x, y)}$$

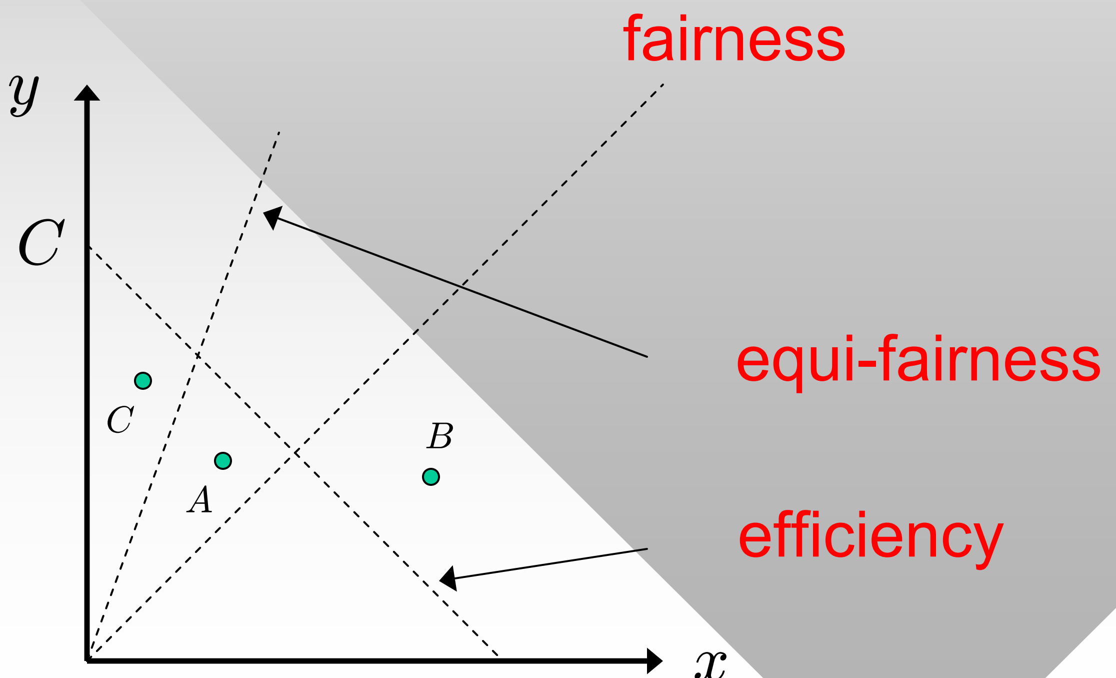
TCP Fairness 2

- Fairness index $= 1$ is ideal since the rates are equal
- Fairness index $= 0$ means maximally unfair conditions
- Analysis using the system trajectory plot
  - Trajectory follows rates of flows x and y on a 2D plane
  - The plot connects points $(x(t),\, y(t))$, where t is time in RTT steps, $x(t)$ and $y(t)$ are the rates of the two flows



TCP Fairness 3

- Useful lines on this 2D plane
  - Fairness: $y = x$
  - Efficiency: $x+y = C$
  - Equi-fairness: $y = mx$ (infinitely many, one for each m)
- Visual analysis
  - Which point(s) have packet loss?
  - Which point is more fair A or C?

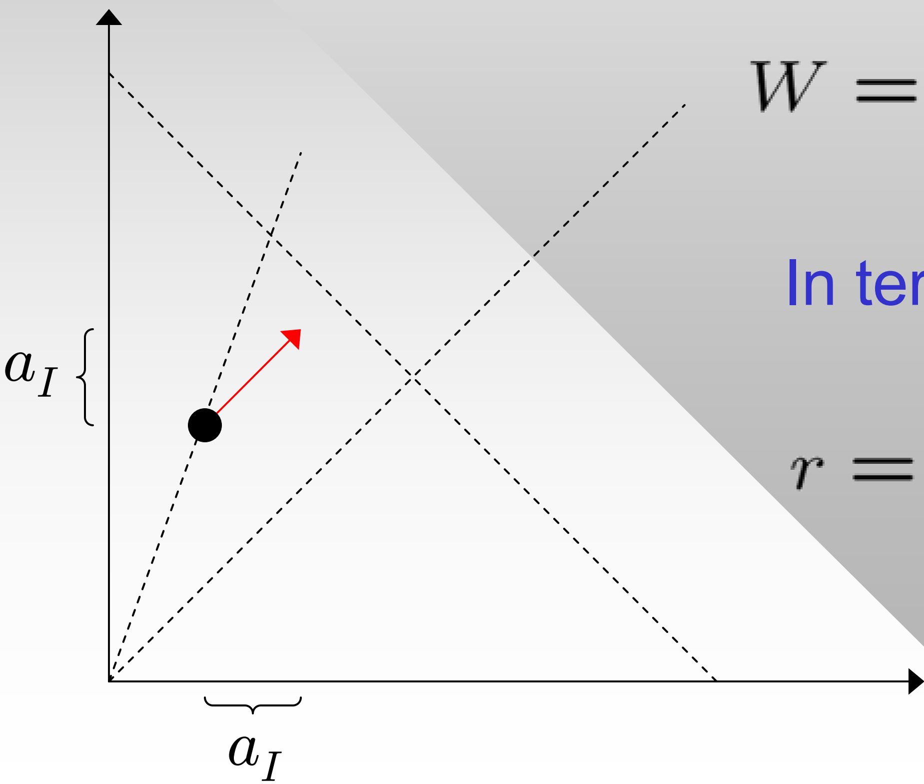


TCP Fairness 4

- The fairness line is where flow rates are equal
  - Hence, the goal is to converge the system to this line
- The efficiency line intersects both axes at C
  - When flows cross the efficiency line, they have loss
  - In uncongested cases, the system is below this line
- All points along the equi-fairness line have the same fairness index
  - Given initial flow rates (x,y), rates (αx,αy) have the same fairness index for any $\alpha > 0$

TCP Fairness 5

- Now examine what AIMD does (fixed MSS and RTT)
  - Start with additive increase
  - Why is this move parallel to the fairness line?
  - What happens to fairness?



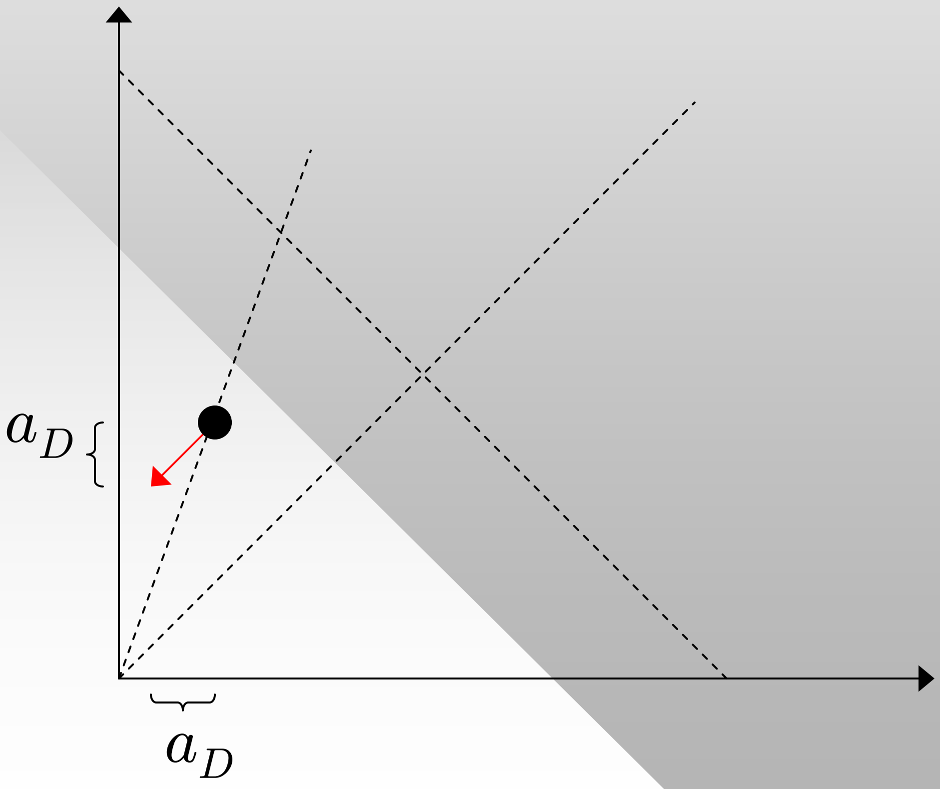
$$W = \begin{cases} W + 1 & \text{per RTT} \\ W/2 & \text{per loss} \end{cases}$$

In terms of rate:

$$r = \begin{cases} r + \underbrace{\frac{MSS}{RTT}}_{a_I} & \text{per RTT} \\ r/2 & \text{per loss} \end{cases}$$

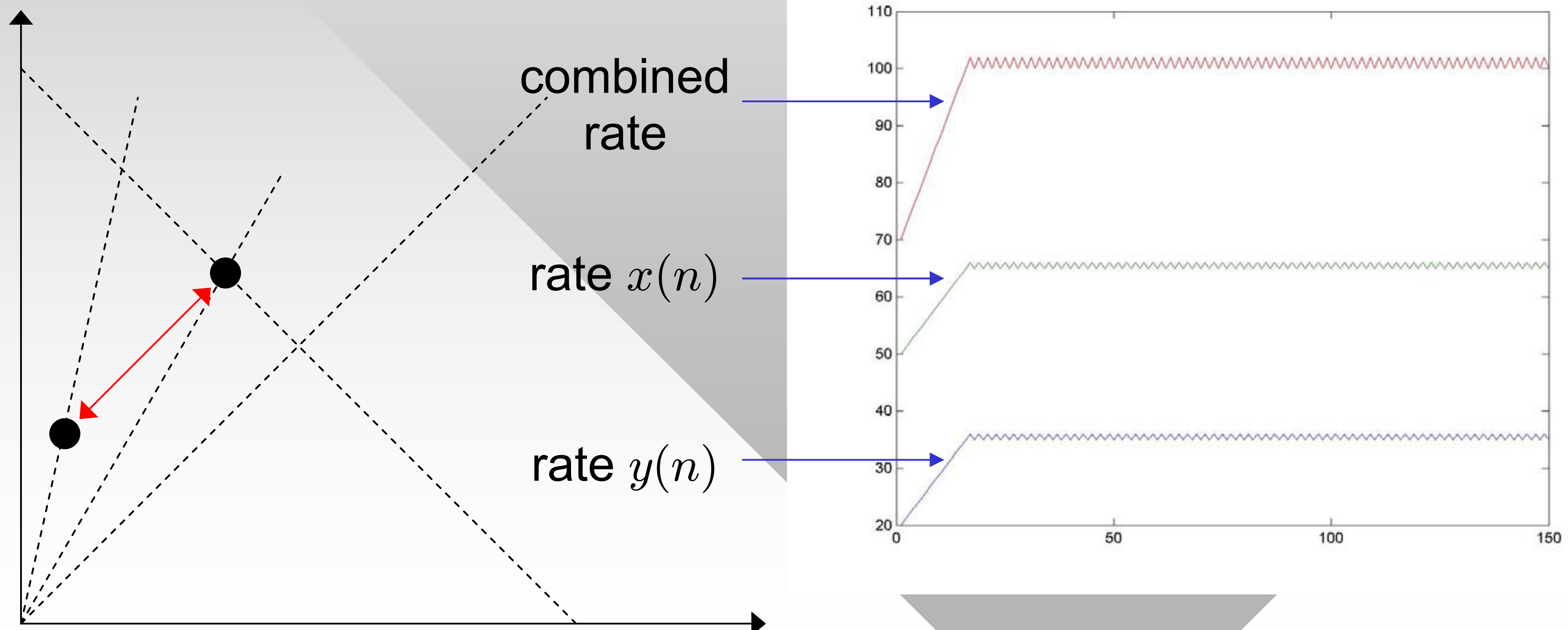
TCP Fairness 6

- Now consider additive decrease
  - Additive constant in the decrease step reduces fairness



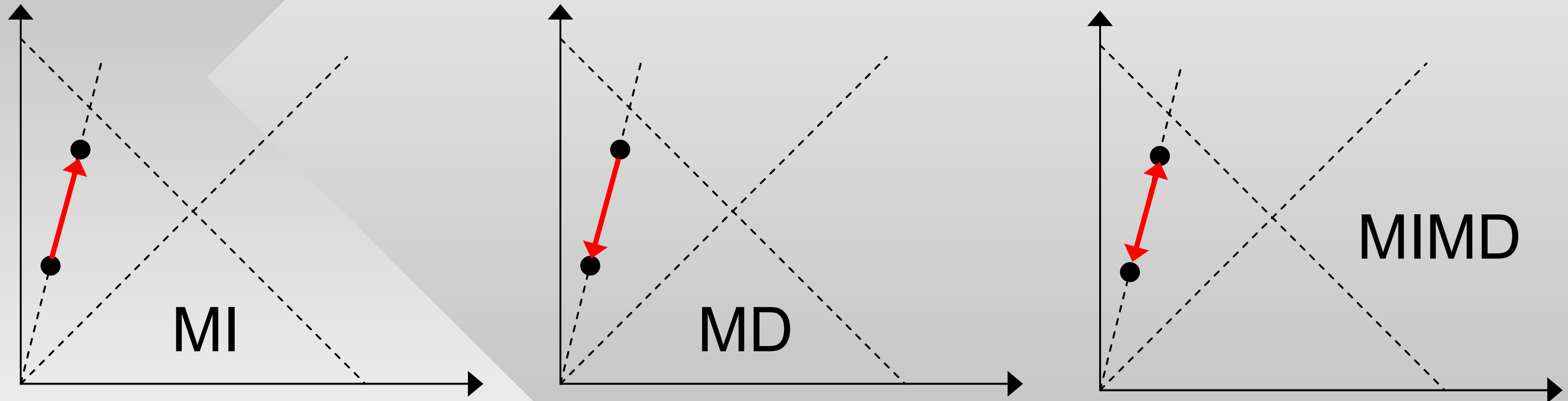
TCP Fairness 7

- Now examine a combination of additive increase and additive decrease (AIAD):
  - The system fluctuates between two unfair states without convergence to the fairness line



TCP Fairness 8

- Now examine MI (multiplicative increase) and MD (multiplicative decrease)



- What happens to fairness in each case?
- MIMD moves the system along the corresponding equi-fairness line
  - Does not converge to fairness either

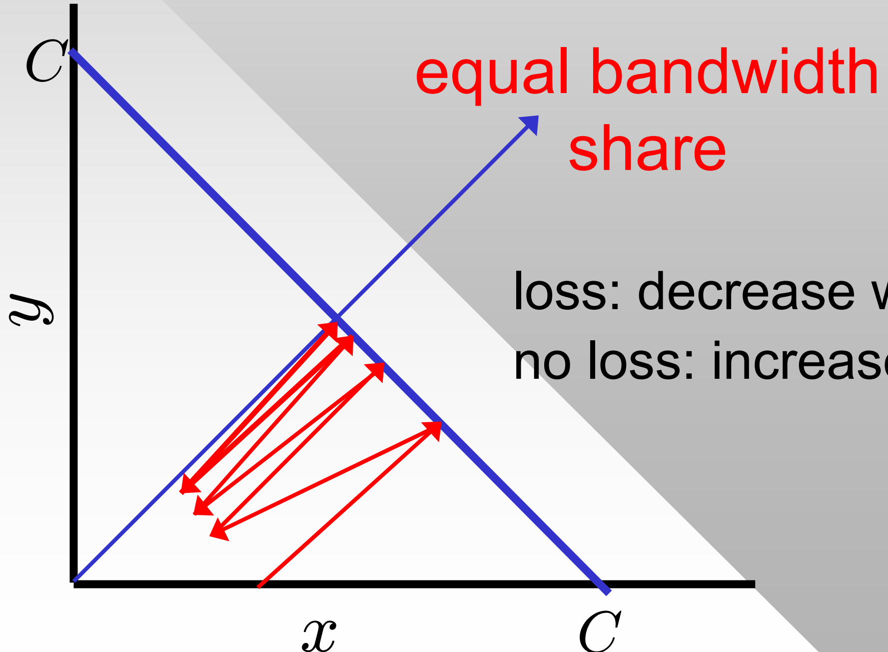
Why Is TCP Fair?

Two competing sessions

- Assume initial rate y is lower, i.e., $x(0) > y(0)$:
- First consider the additive increase (AI) step
  - New rates:
    $x(n+1) = x(n) + MSS/RTT$,
    $y(n+1) = y(n) + MSS/RTT$
  - Prove that $\Phi(n+1) > \Phi(n)$
- Multiplicative decrease (MD)
  - New rates $x(n+1) = x(n)/2$, $y(n+1) = y(n)/2$
  - Prove that $\Phi(n+1) = \Phi(n)$

Why Is TCP Fair?

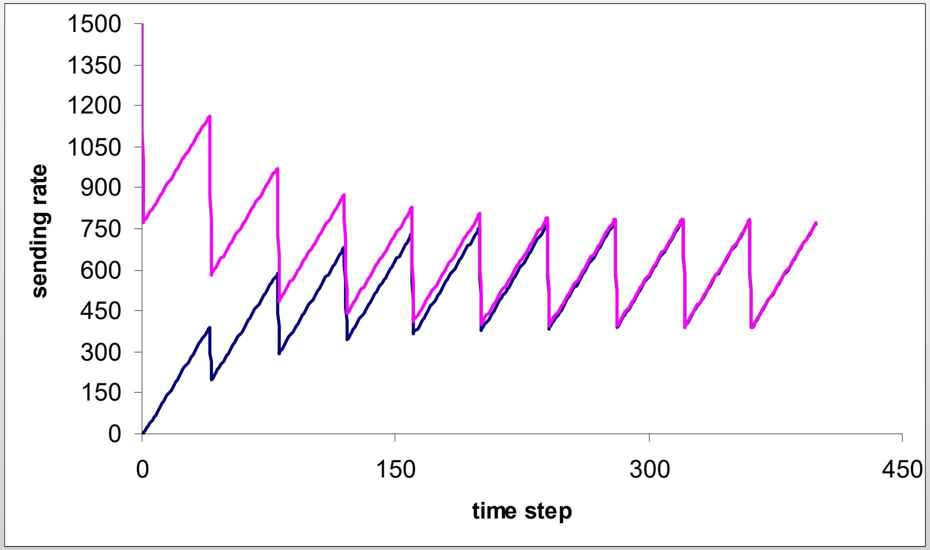
- Fairness stays the same during MD and improves during AI, eventually converging to 1
  - Intuitive reasoning: during increase, both flows gain bandwidth at the same rate; however, during decrease, the faster flow releases more

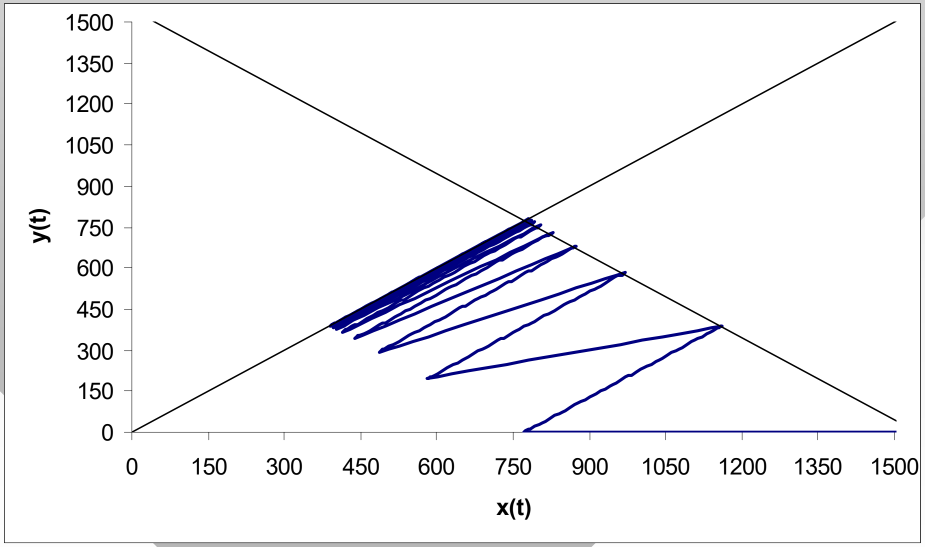


loss: decrease window by factor of 2
no loss: increase by MSS/RTT

Fairness Example

- AIMD example
  - $C =$ 1544 Kbps, 2 flows
- Start in the maximally unfair state
  - $x(0) = 1544, y(0) = 0$
- Eventually converge to fairness
- Caveat: fairness in TCP is achievable only when flows have the same RTT and MSS





Fairness (Final Thoughts)

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 transport over UDP (e.g., QUIC)

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 10 flows present:
  - New app asks for 1 TCP connection, gets rate $C/11$
  - New app asks for 10 TCPs, gets $C/2$

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"