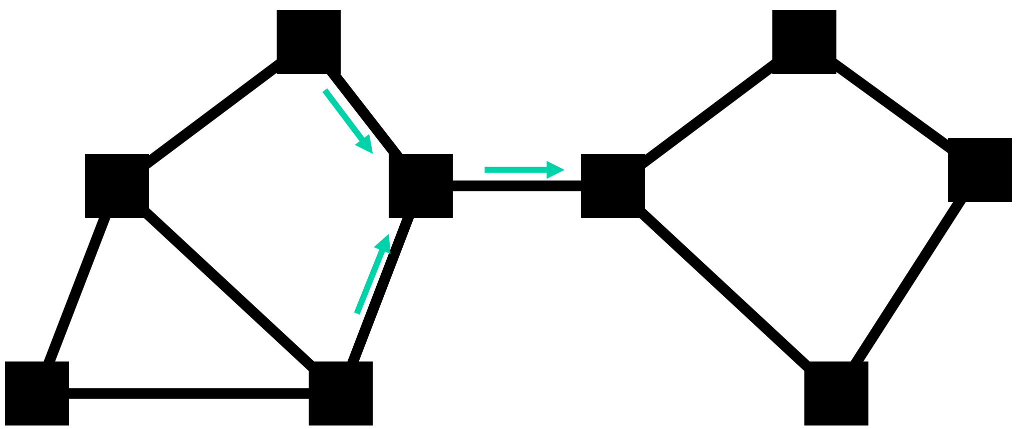
COMP/ELEC 429/556
Introduction to Computer Networks

Principles of Congestion Control

Some slides used with permissions from Edward W. Knightly, T. S. Eugene Ng, Ion Stoica, Hui Zhang

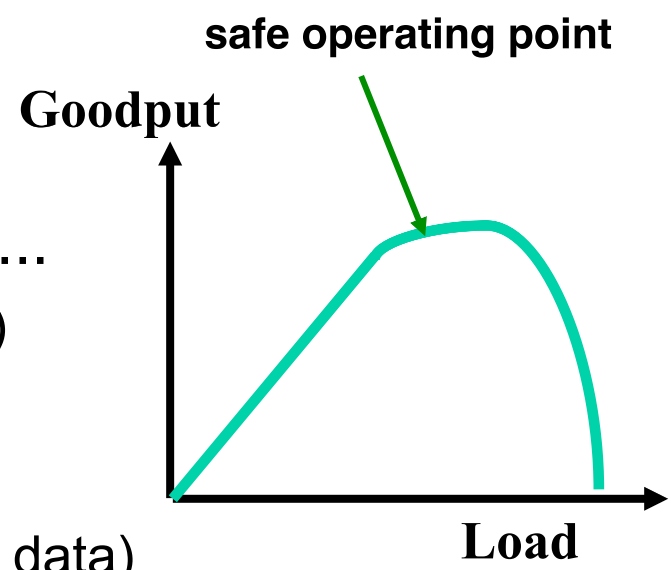
What is Congestion?

- The load placed on the network is higher than the capacity of the network
  - Not surprising: independent senders place load on network
- Results in packet loss: routers have no choice
  - Can only buffer finite amount of data

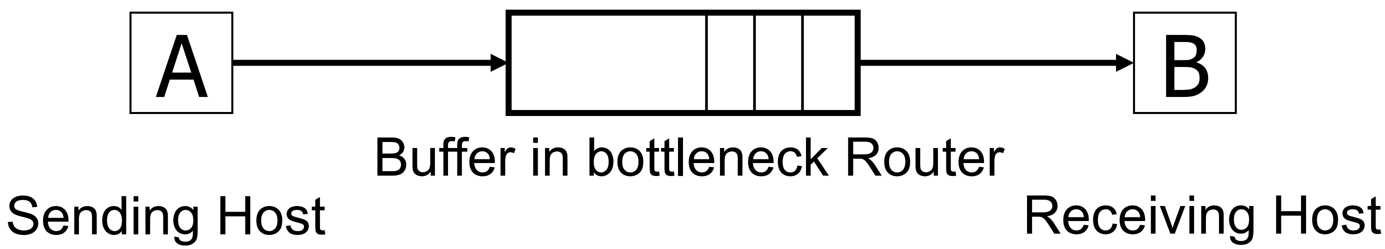


How Fast to Send? What's at Stake?

- Send too slow: link sits idle
  - wastes time
- Send too fast: link is kept busy but....
  - queue builds up in router buffer (delay)
  - overflow buffers in routers (loss)
  - Many retransmissions, many losses
  - Network goodput (throughput of useful data) goes down
  - "Congestion collapse"

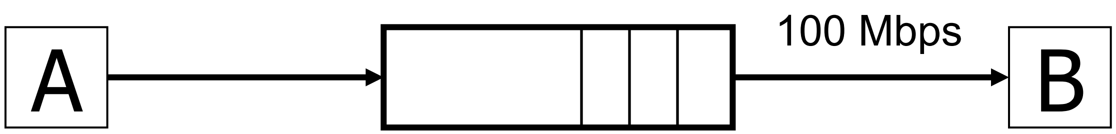


Abstract View



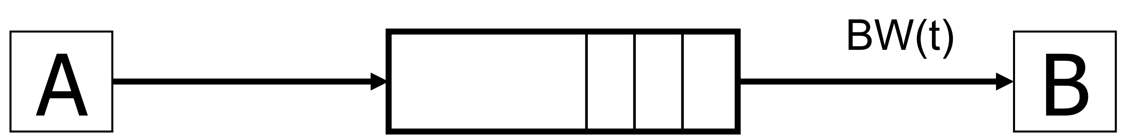
- We ignore internal structure of network and model it as having a single bottleneck link

Problem 1: Single Flow, Fixed Bandwidth



- Adjust rate to match bottleneck bandwidth
  - without any *a priori* knowledge
  - could be 40 Gbps link, could be a 32 Kbps link

Problem 2: Single Flow, Varying Bandwidth

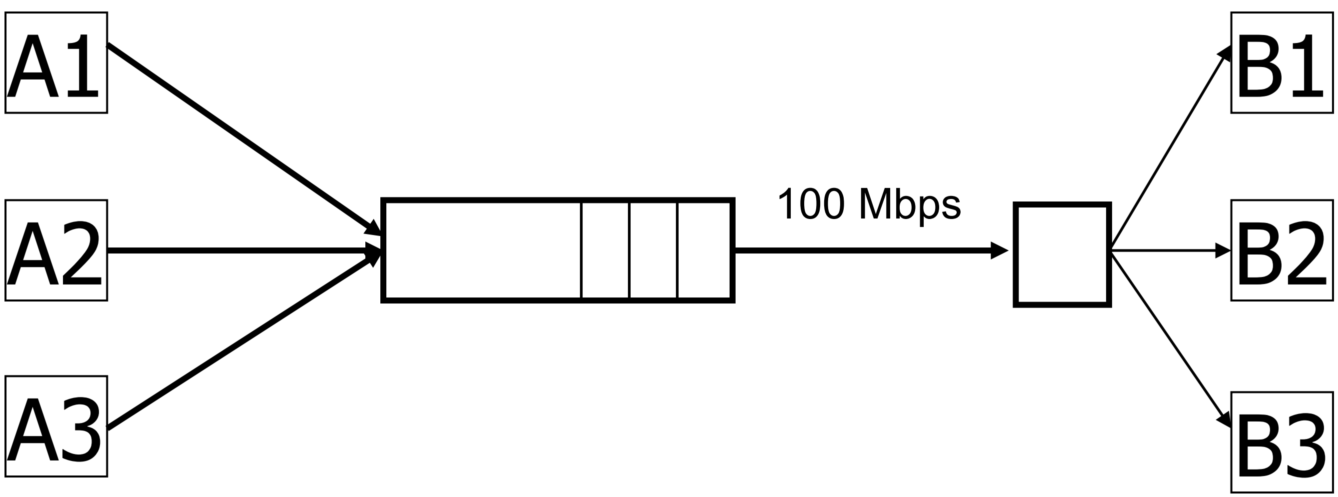


- Adjust rate to match instantaneous bandwidth
- Bottleneck can change because of a routing change

Problem 3: Multiple Flows

Two Issues:

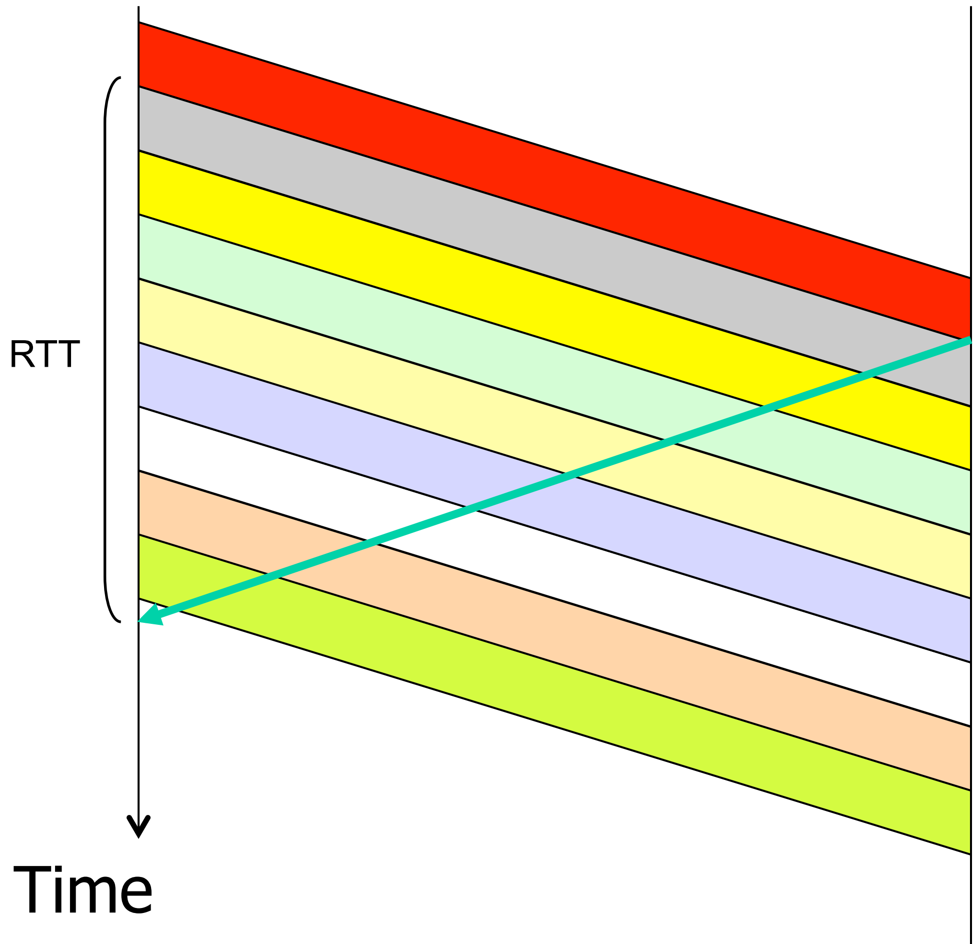
- Adjust total sending rate to match bottleneck bandwidth
- Allocation of bandwidth between flows



General Approaches

- Reservation
  - pre-arrange bandwidth allocations
  - requires negotiation before sending packets
  - requires router support

Sliding Window



Window size n = 9, i.e. 9 packets in one RTT

In general, sending rate proportional to n/RTT

General Approaches (cont'd)

- Dynamic sending rate adjustment
  - Every sender probe network to test level of congestion
  - speed up when no congestion
  - slow down when congestion
  - suboptimal, messy dynamics, but simple to implement
  - requires <u>no</u> router support

  - Distributed coordination problem!

Sliding Window Congestion Control

- Sender has a send window
  - controls amount of unacknowledged data in transit
- Sending rate proportional to: Send window size/RTT
- Vary send window size to control sending rate

Two Basic Components

- Detecting congestion
- Rate adjustment algorithm (change window size)
  - depends on congestion or not

Detecting Congestion

- Packet dropping is a plausible sign of congestion
  – delay-based methods are hard and risky
- How do you detect packet drops? ACKs
  – ACKs signal receipt of data
  – ACK denotes last contiguous byte received
- Two signs of packet drops
  – No ACK after certain time interval: time-out
  – Several duplicate ACKs for the same sequence number
- This heuristic may not work well for wireless networks, why?
  – Think whether packet drops are always due to congestion

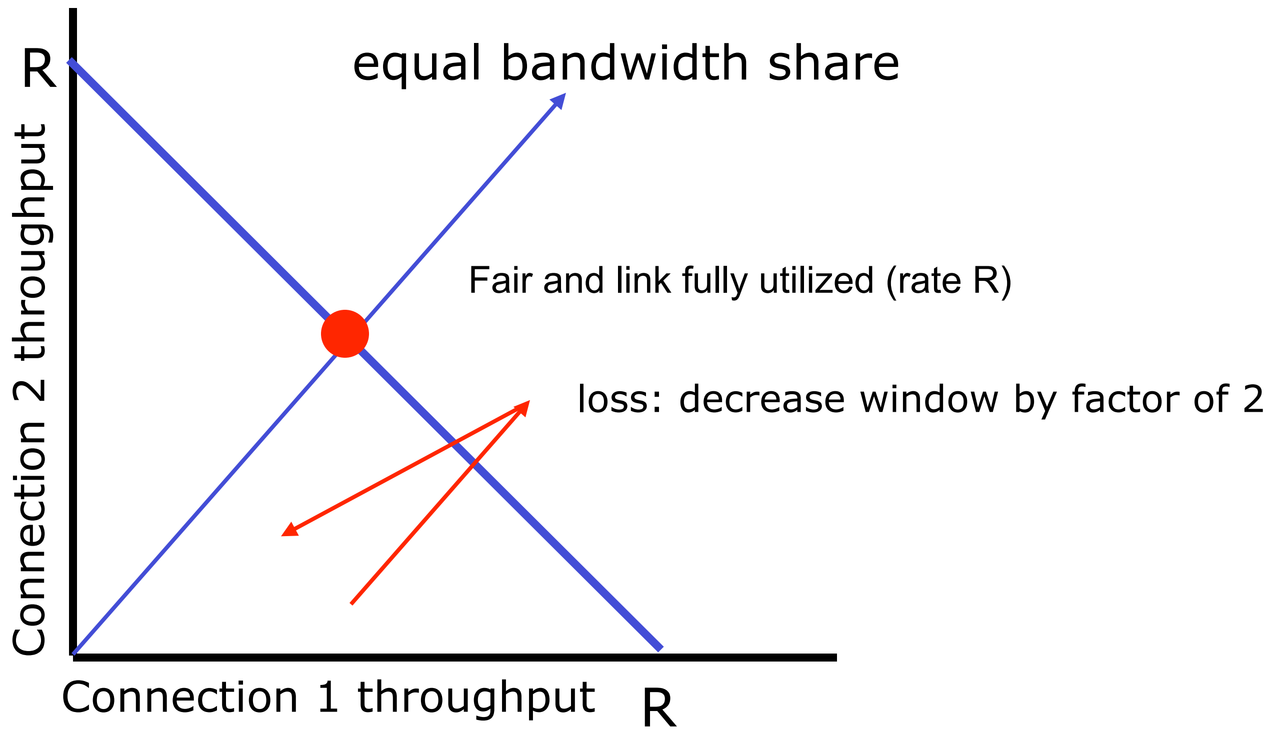
Rate Adjustment

- Basic idea:
  - Upon receipt of ACK (of new data): increase rate
    - Data successfully delivered, perhaps can send faster
  - Upon detection of loss: decrease rate
- But how much increase/decrease should be applied?
  - What outcomes do we want?
- For simplicity, restrict to “additive” and “multiplicative” increase/decrease
  - “additive” results in linearly change
  - “multiplicative” results in exponential change

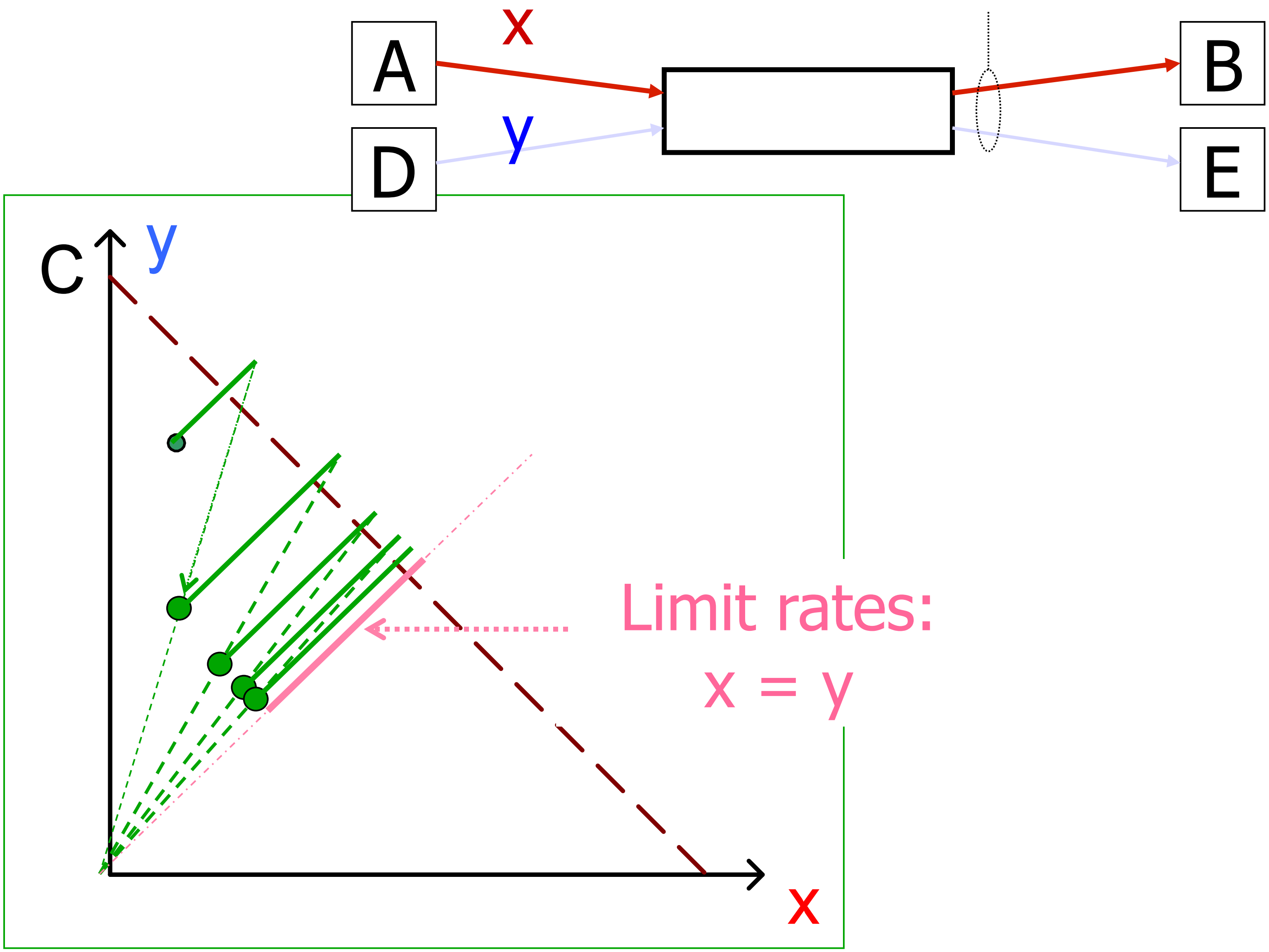
Fairness & Efficiency

Two competing sessions:

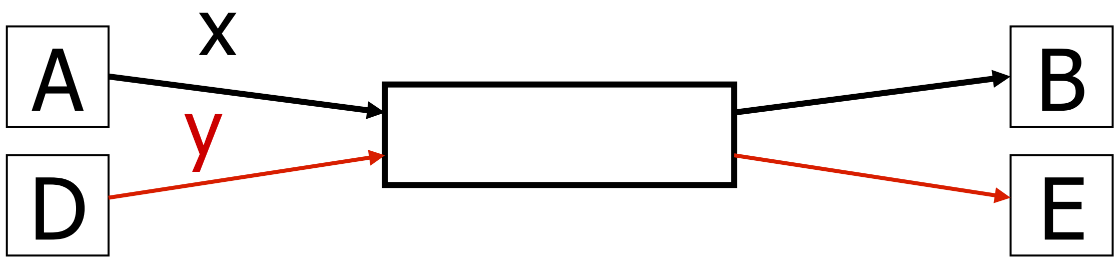
- Additive increase (AI) gives slope of 1, as throughout increases
- multiplicative decrease (MD) decreases throughput proportionally



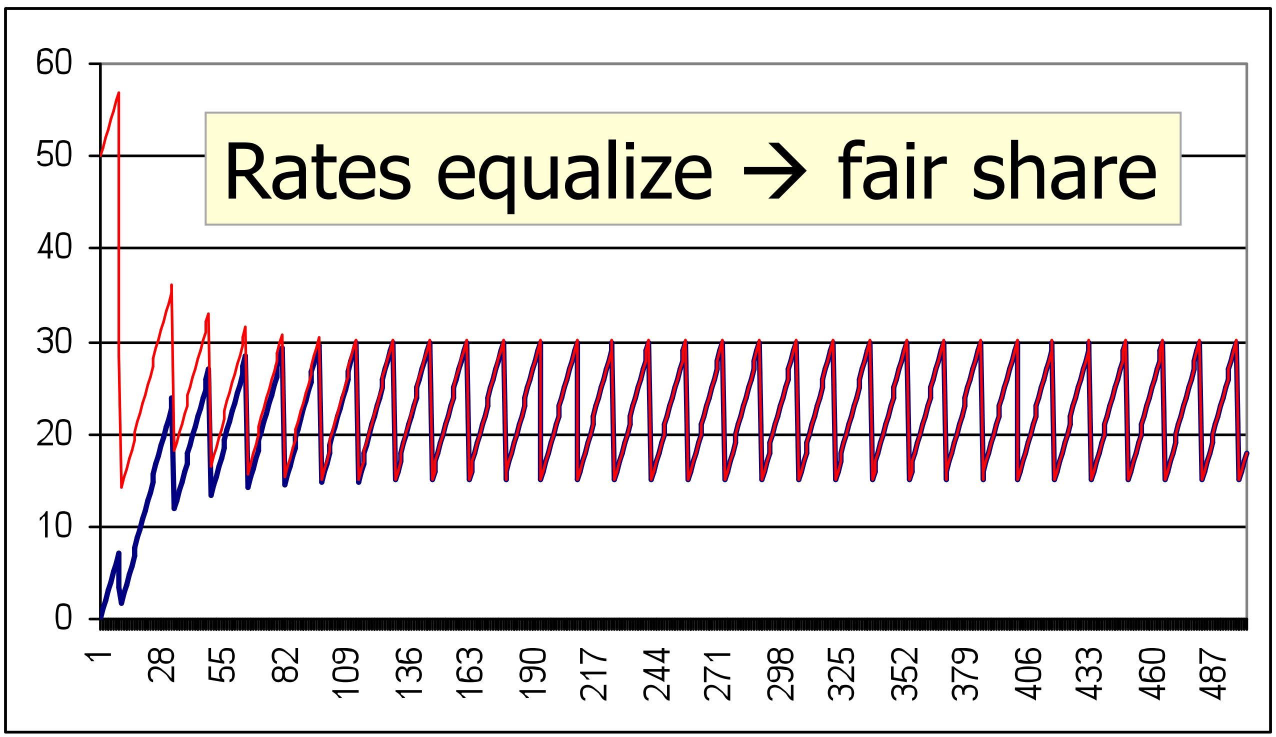
AIMD



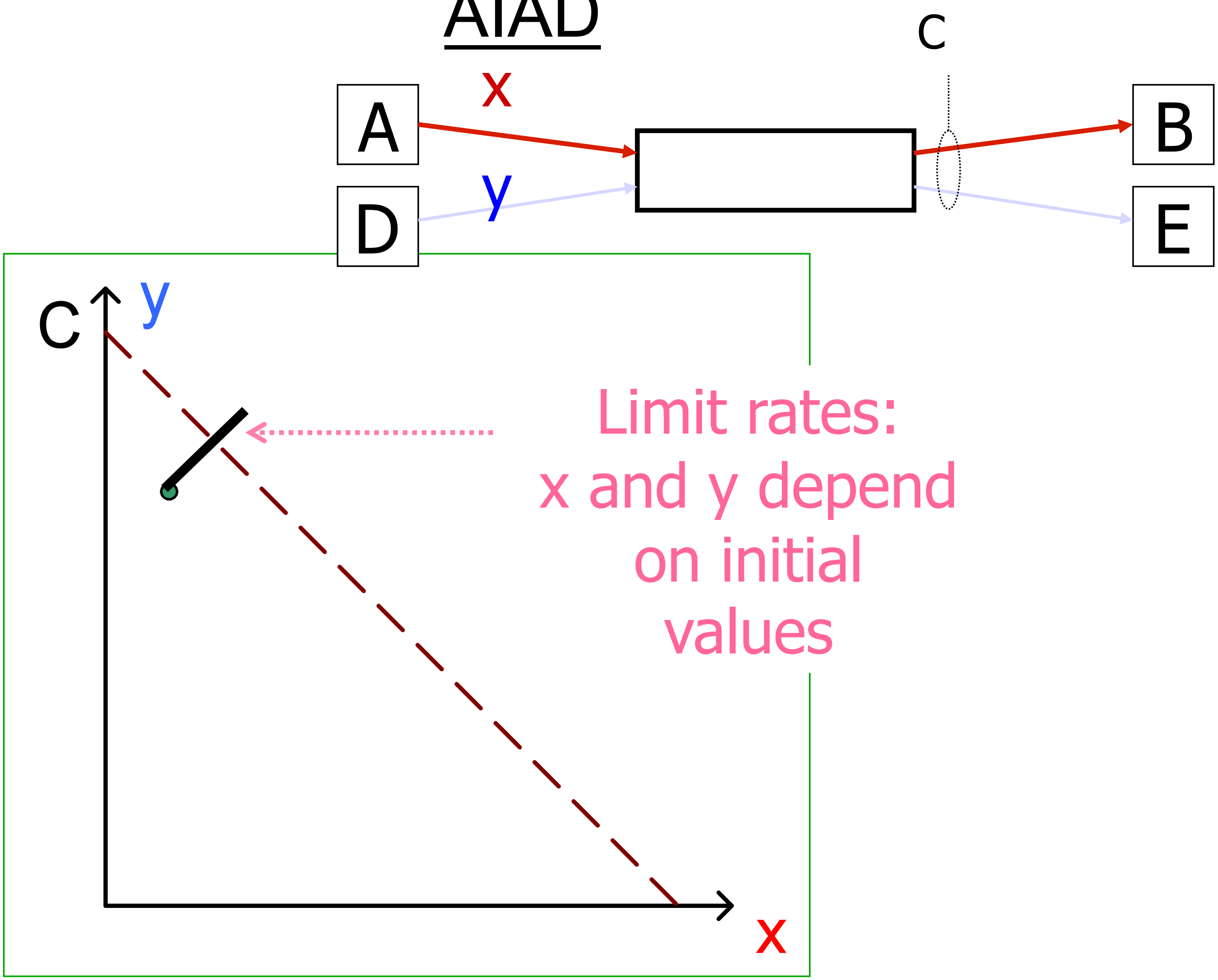
AIMD Sharing Dynamics



- No congestion → rate increases by one packet/RTT every RTT
- Congestion → decrease rate by factor 2

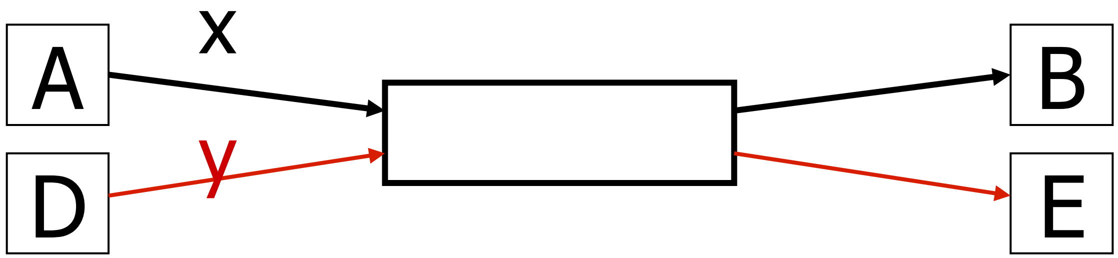


AIAD

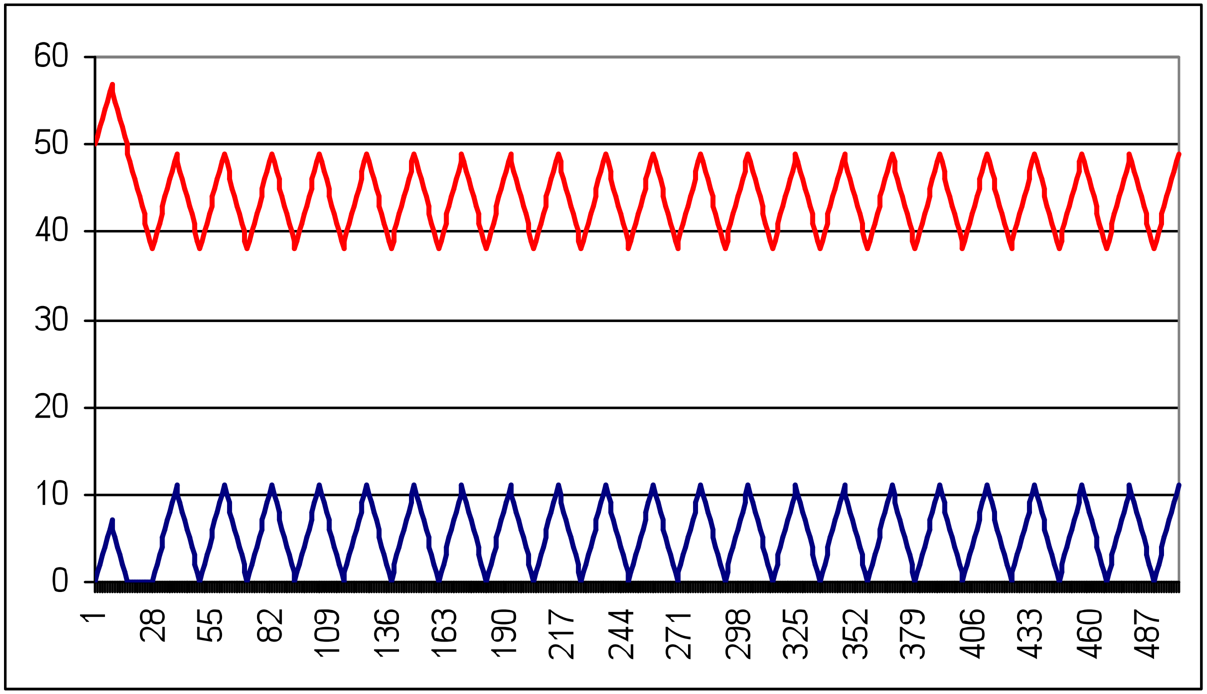


Limit rates:
x and y depend
on initial
values

AIAD Sharing Dynamics



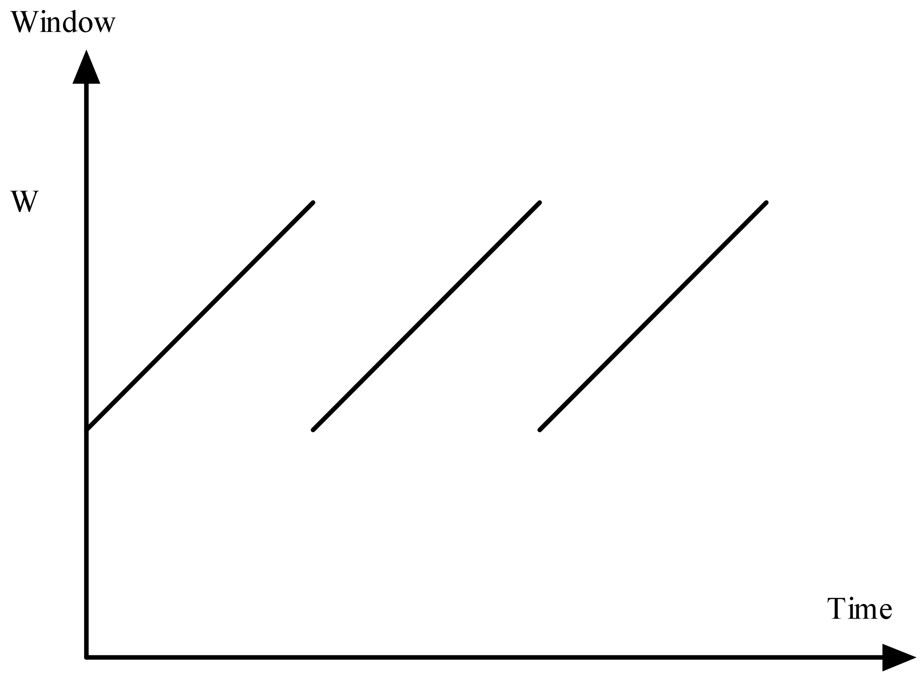
- No congestion → x increases by one packet/RTT every RTT
- Congestion → decrease x by 1



AIMD Model

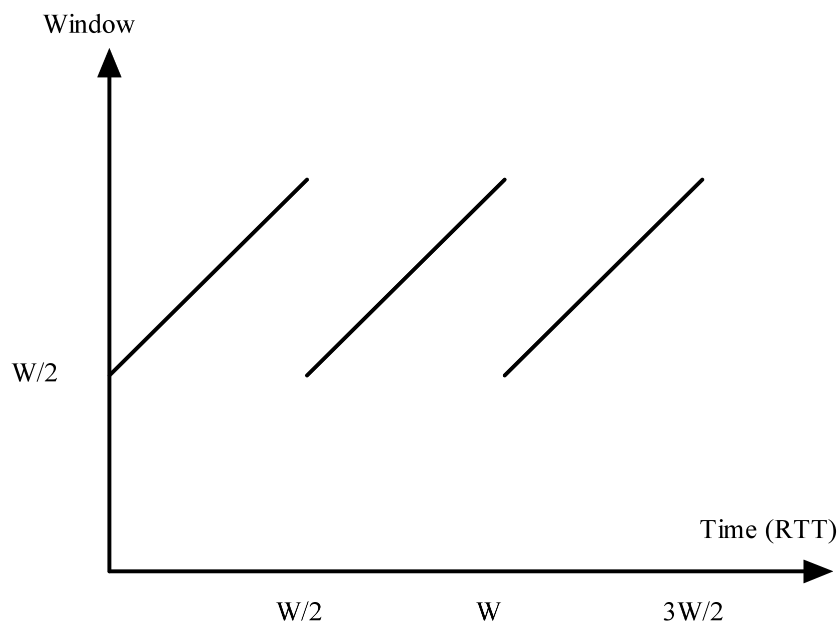
- Analyze the steady state throughput as a function of
  - RTT
  - Loss probability
- Assumptions
  - Each packet dropped with *iid* probability p
- Methodology: analyze "average" cycle in steady state
  - How many packets are transmitted per cycle?
  - What is the duration of a cycle?

Cycles in Steady State



- Denote *W* as the maximum achieved window
- What is the slope of the line?
- What are the key values on the time axis?

Cycle Analysis



W increase by 1 per RTT

$$\text{pkts xmitted/cycle} = \text{area} = \left(\frac{W}{2}\right)^2 + \frac{1}{2}\left(\frac{W}{2}\right)^2 = \frac{3}{8}W^2$$

Throughput

$$\text{throughput} = \frac{\text{pkts xmitted/cycle}}{\text{time/cycle}} = \frac{\frac{3}{8}W^2}{RTT\left(\frac{W}{2}\right)}$$

- What is W as a function of p?
  How long does a cycle last until a drop?

Cycle Length

Let α be the index of the lost packet that ends a cycle

$$P(\alpha = k) = P(k-1 \text{ pkts not lost, } k\text{th pkt lost})$$

$$= (1-p)^{k-1} p$$

$$\Rightarrow \quad E(\alpha) = \sum_{k=1}^{\infty} k(1-p)^{k-1} p = \frac{1}{p}$$

$$\Rightarrow \quad \frac{1}{p} = \frac{3}{8} W^2 \qquad\qquad \Rightarrow \quad W = \sqrt{\frac{8}{3p}}$$

AIMD Model

$$\text{throughput } T(p) = \frac{1/p}{RTT \cdot \frac{1}{2}\sqrt{\frac{8}{3p}}} = \frac{1}{RTT\sqrt{\frac{2}{3}p}}$$

- Note role of RTT. Is it "fair"?
- A "macroscopic" model