Congestion Control

Prof. Anja Feldmann, Ph.D.

(Based on slide deck of Computer Networking, 7th ed., Jim Kurose and Keith Ross.)

Outline

- *Connection-oriented* transport: TCP
  - Reliable data transfer
  - Flow control
  - Connection management
- Congestion control
  - Principles
  - Mechanism

Congestion?

Congestion

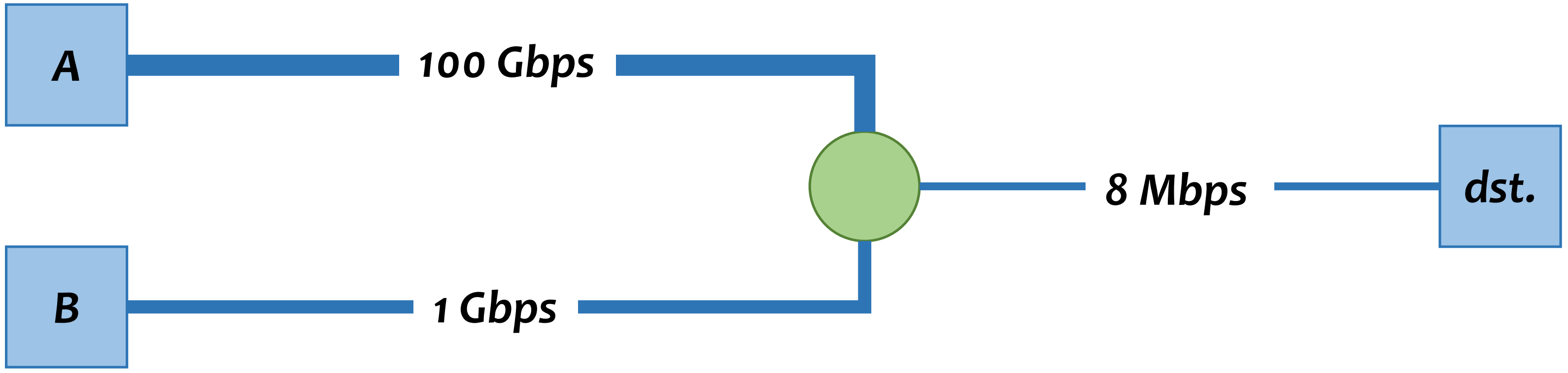
- Informally: *"too many sources sending too much data too fast for network to handle"*
- *Different from flow control!*

How does it manifest?

- ***Lost packets*** (*buffer overflow* at routers)
- ***Long delays*** (*queueing* in router buffers)

A top-10 problem!

Congestion: Problem



Different sources compete for resources inside network

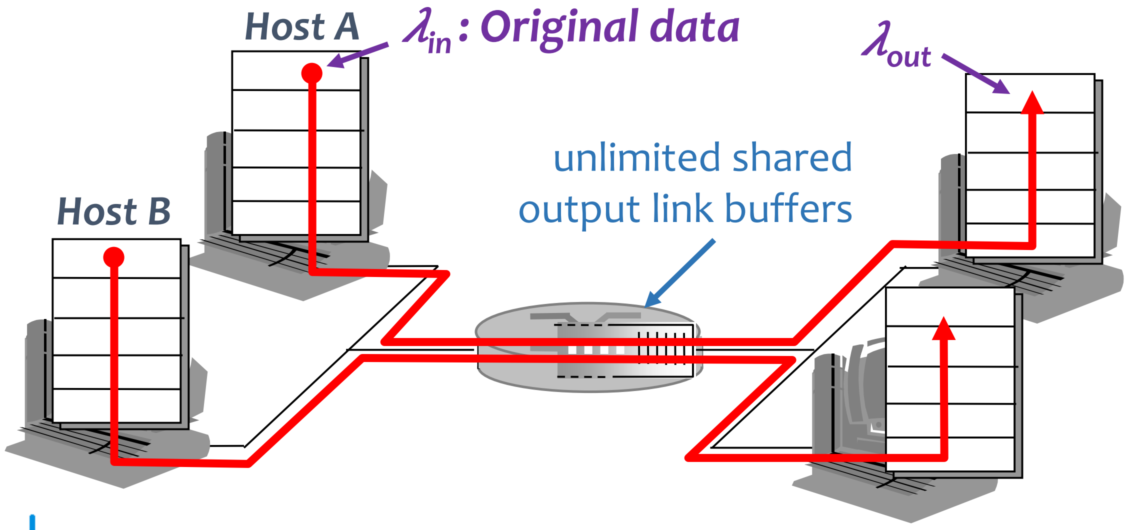
— Why is it a problem?

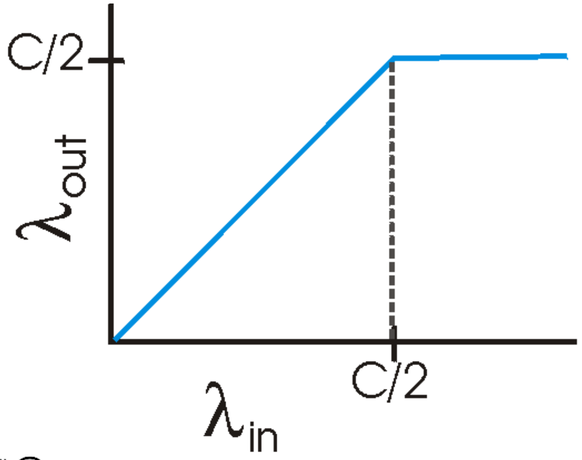
- Sources are *unaware* of
  - *Current state of resource*
  - *Each other*

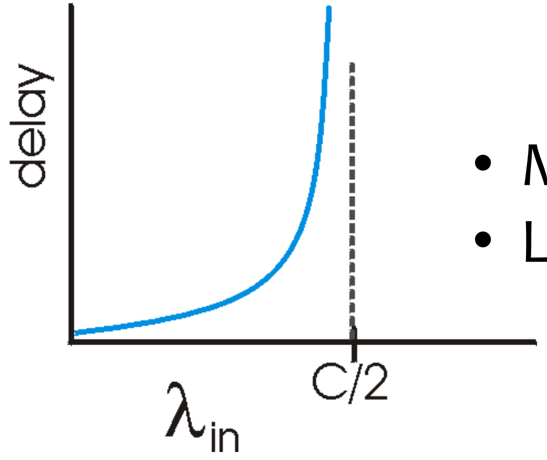
Often results in ***< 8 Mbps*** of *throughput* (***congestion collapse***)

Causes & Costs of Congestion: Scenario 1

- Two senders, two receivers
- One router, ***infinite*** buffers
- No retransmission



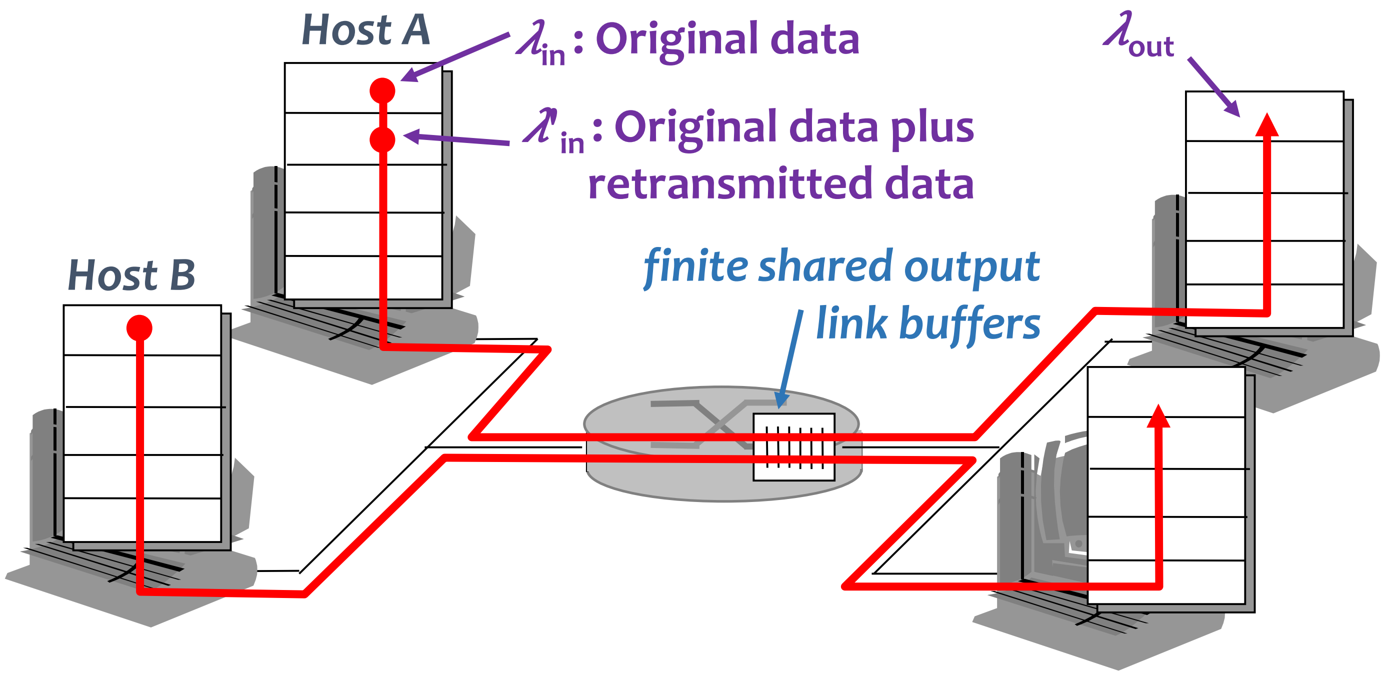




- Maximum achievable throughput
- Large delays when congested

Causes & Costs of Congestion: Scenario 2

- One router, *finite* buffers
- Sender retransmits lost packet

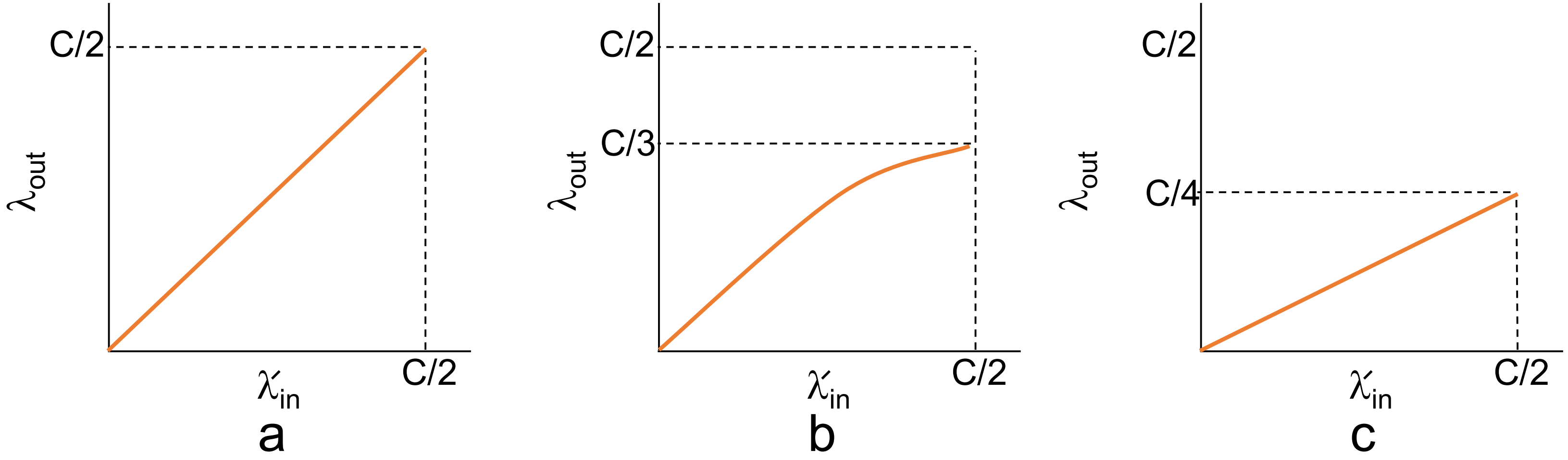


Causes & Costs of Congestion: Scenario 2

a) Always: $\lambda_{in} = \lambda_{out}$ (**goodput**)

b) *"Perfect"* retransmission only when loss: $\lambda'_{in} > \lambda_{out}$

c) Retransmission of delayed (not lost) packet makes λ'_{in} larger (than perfect case) for same λ_{out}

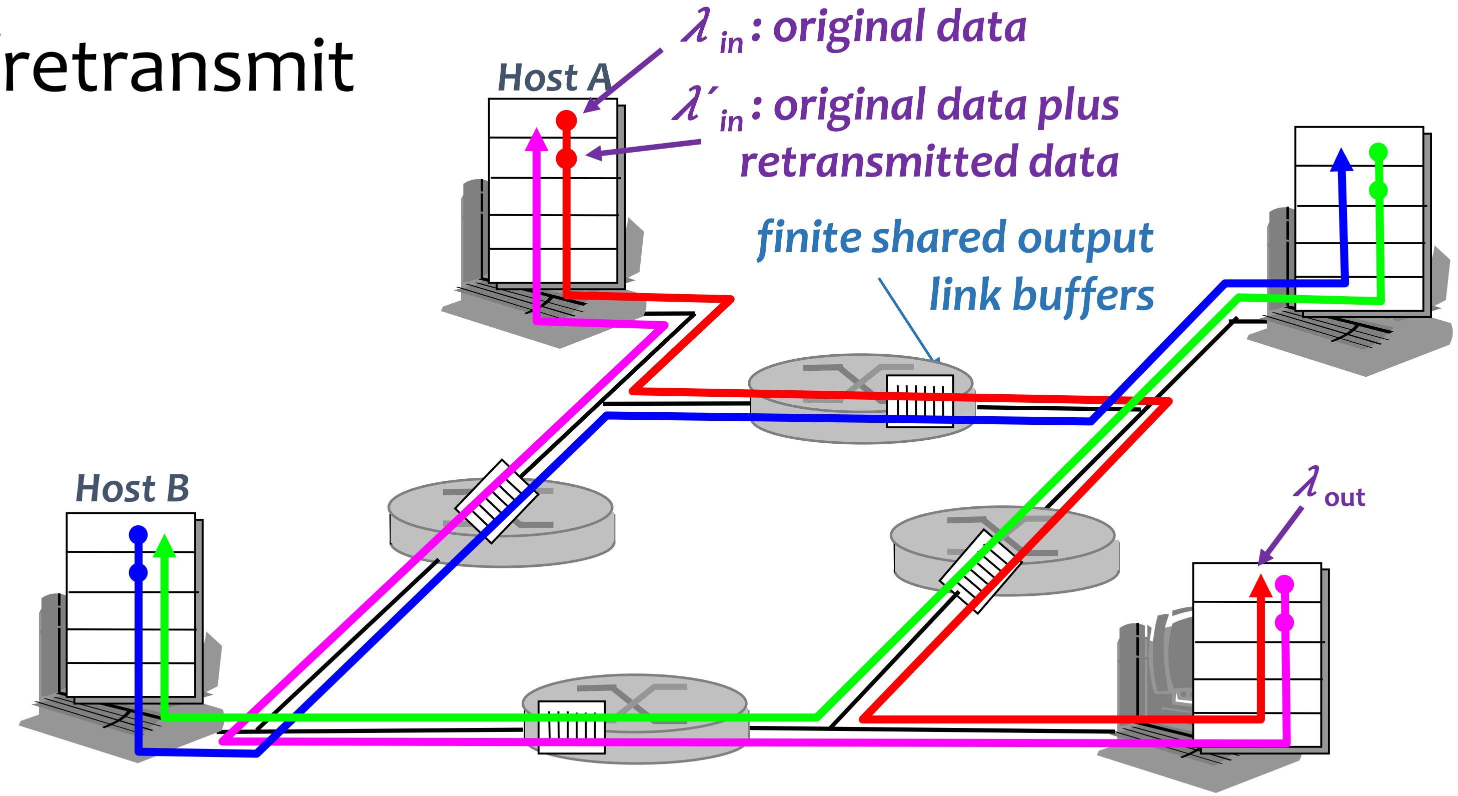


"Costs" of congestion:

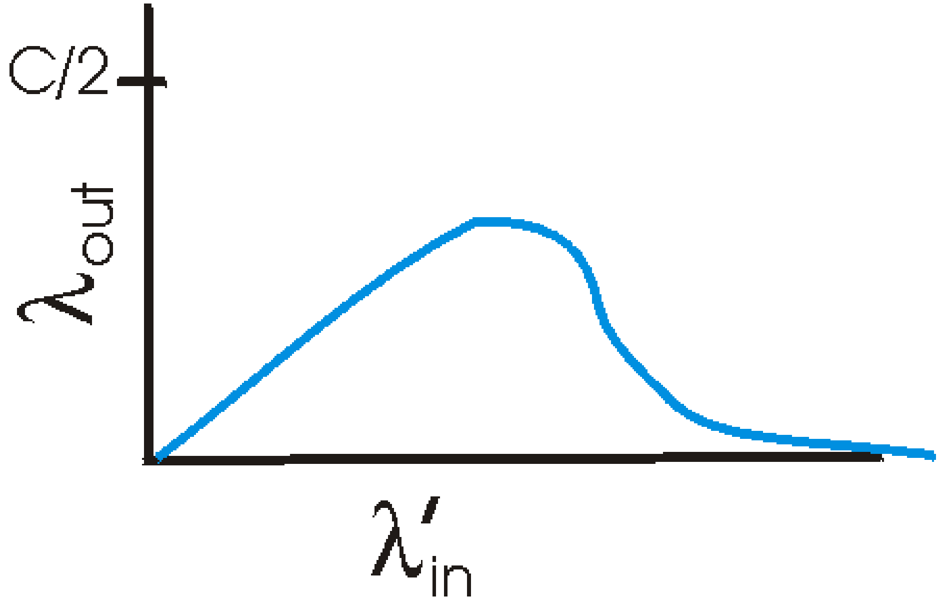
- More work (retransmissions) for given *"goodput"*
- Unnecessary retransmissions: Link carries multiple copies of the **same** packet

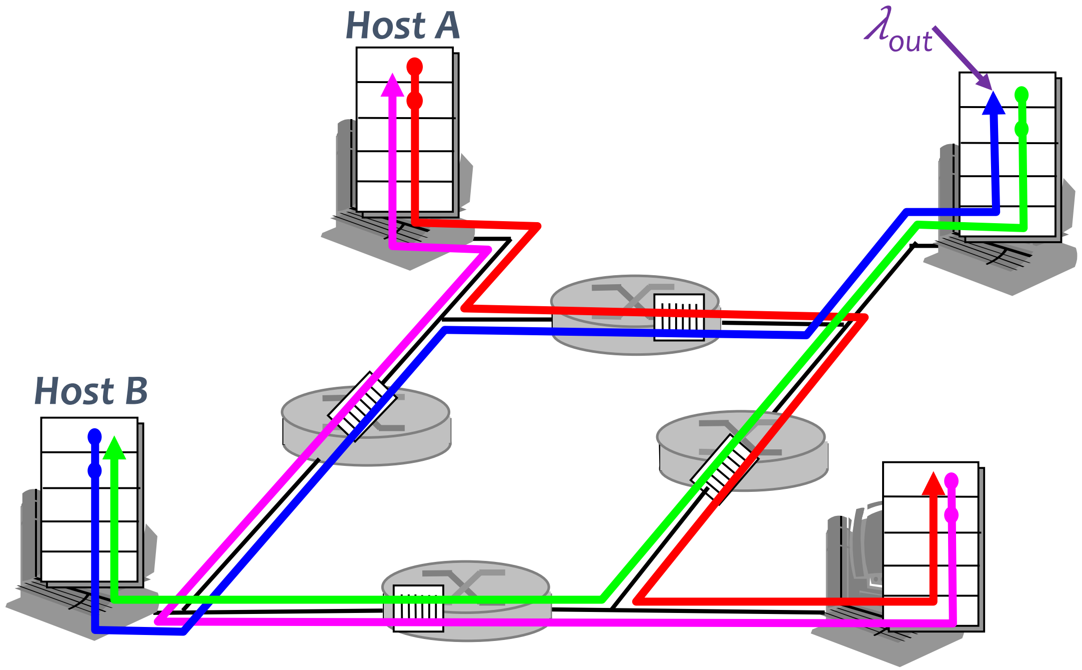
Causes & Costs of Congestion: Scenario 3

- Four senders
- Multi-hop paths
- Timeout/retransmit



Causes & Costs of Congestion: Scenario 3





Another "cost" of congestion:

- When packet dropped, any "*upstream*" transmission capacity used for that packet was wasted!

Congestion Collapse

Defn.: ***Increase in network load results in decrease of useful work done***

Many possible causes

- ***Spurious retransmissions of packets still in flight***
  - Classical congestion collapse
  - How can this happen with packet conservation
  - *Solution*: Better timers and TCP congestion control
- ***Undelivered packets***
  - Packets consume resources and are dropped elsewhere in network
  - *Solution*: Congestion control for **ALL** traffic

Congestion Collapse: Other Causes

Fragments

- Mismatch of transmission and retransmission units
- Solutions: (a) Make network drop all fragments of a packet;
  (b) Do path MTU discovery

Control traffic

- Large percentage of traffic is for control
  (Headers, routing messages, DNS, etc.)

Stale or unwanted packets

- Packets that are delayed on long queues
- "Push" data that is never used

Where to prevent collapse?

Can end hosts prevent problem?

- Yes, but must trust end hosts to do right thing

  (e.g., sending host must adjust amount of data it puts in the network based on detected congestion)

Can routers prevent collapse?

- No, not all forms of collapse; does not mean they can not help!
- Sending accurate congestion signals
- Isolating *well-behaved* from *ill-behaved* sources

Congestion Control and Avoidance

A mechanism which

- Uses network resources efficiently
- Preserves fair network resource allocation
- Prevents or avoids collapse

Congestion collapse is not just a theory

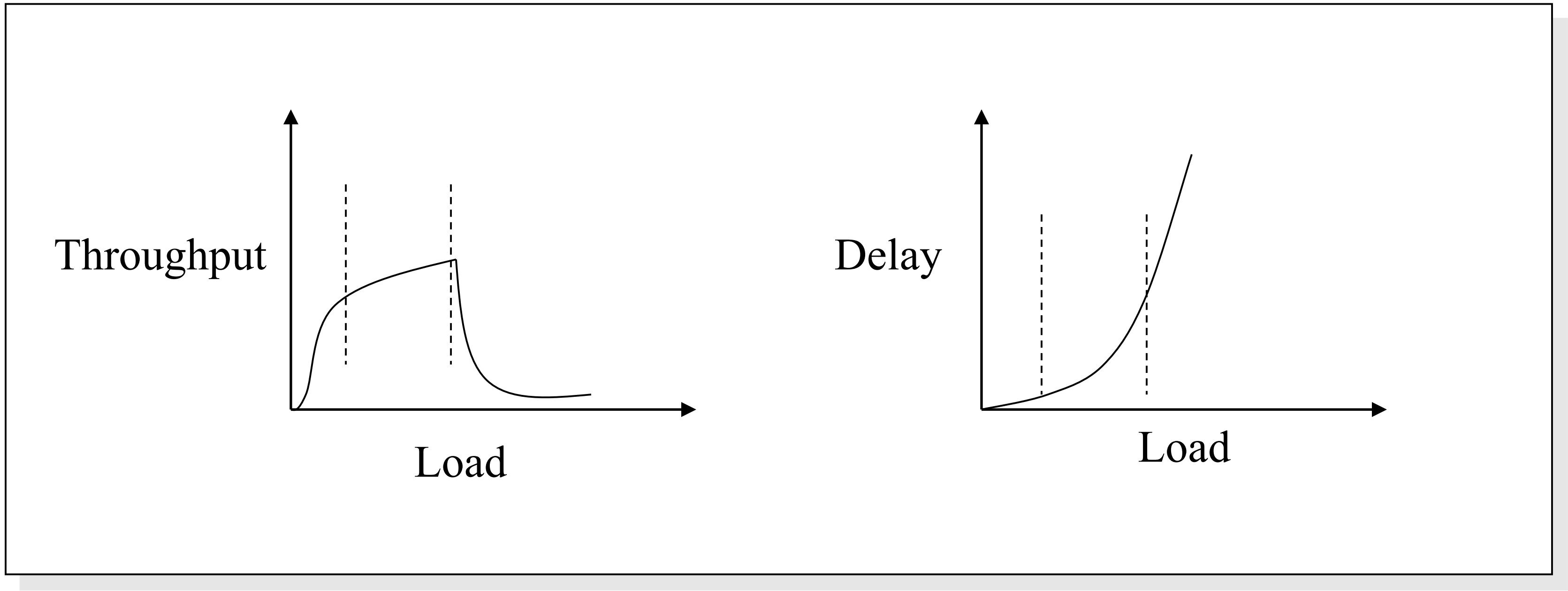
- Has been frequently observed in many networks

Congestion Control

Congestion collapse was first observed on the early Internet in October 1986, when the **NSFnet** phase-I backbone dropped three orders of magnitude from its capacity of ***32 kbit/s*** to ***40 bit/s***, and continued to occur until end nodes started implementing *Van Jacobson's* **congestion control** between 1987 and 1988.

Congestion Control vs. Avoidance

- Avoidance keeps the system performing at the knee
- Control kicks in once the system has reached a congested state



Congestion Control: What to do?

Two broad approaches to congestion control

- ***End-to-end*** Congestion Control
- ***Network-assisted*** Congestion Control

Congestion Control: Approaches

End-to-end cong. control:

- No explicit feedback from network
- Congestion inferred from end-system observed loss, delay
- Approach taken by TCP

Network-assisted cong. control:

- Routers provide feedback to end systems
- Choke packet from router to sender
- Single bit indicating congestion *(e.g., SNA, DECbit, TCP/IP ECN, ATM)*
- Explicit rate sender should send at

End-to-end Cong. Control: Objectives

- Simple router behavior
- Distributed-ness
- Efficiency: $X_{knee} = \sum x_i(t)$
- Fairness: $(\sum x_i)^2 / n(\sum x_i^2)$
- Power: $throughput^{\alpha}/delay$
- Convergence: Control system must be stable

Basic Control Model

Let's assume window-based control

Reduce window when congestion is perceived

- How is congestion signaled?
  - Either mark or drop packets
- When is a router congested?
  - Drop tail queues – when queue is full
  - Average queue length – at some threshold

Increase window otherwise

- Probe for available bandwidth – how?

Linear Control

Many diff. possibilities for reaction to congestion and probing

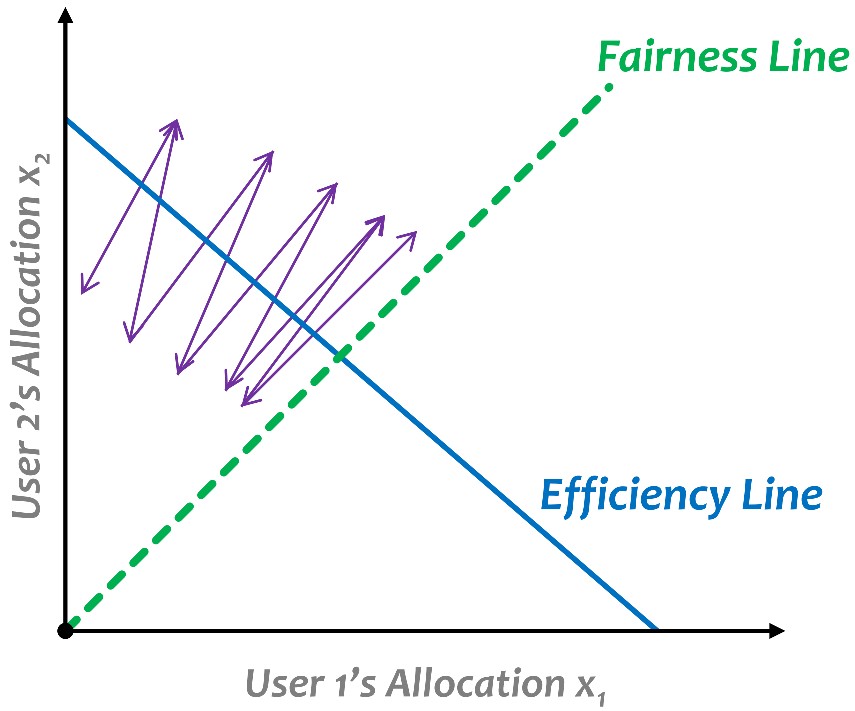
- Simple linear control
- *Window(t + 1) = a + b Window(t)*
- Different a_i/b_i for increase and a_d/b_d for decrease

Supports various reaction to signals

- Increase/decrease additively
- Increased/decrease multiplicatively
- Which combination is optimal?

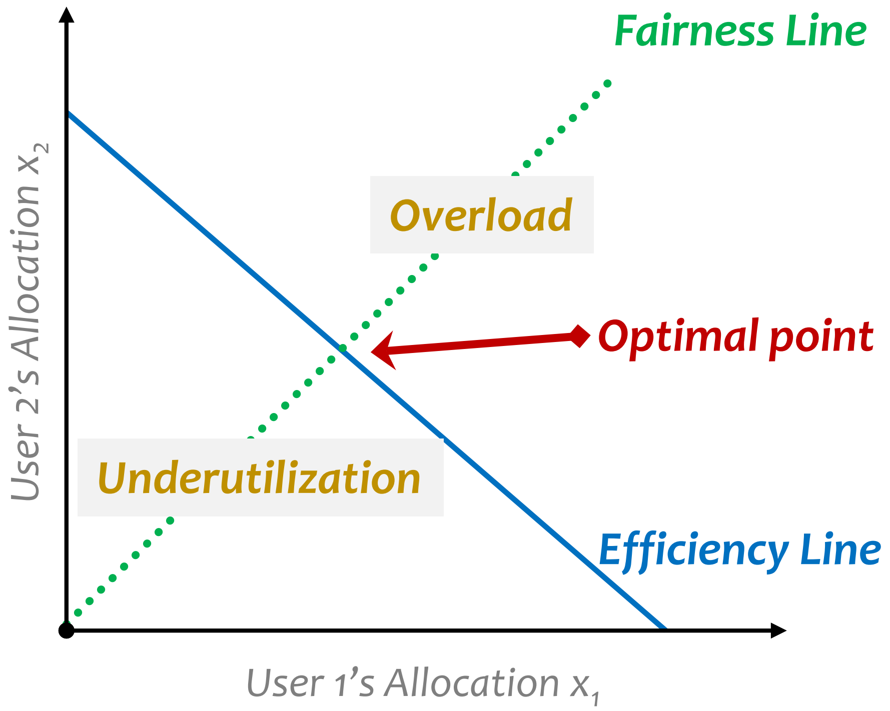
Phase Plot

Simple way to visualize behavior of two competing connections over time



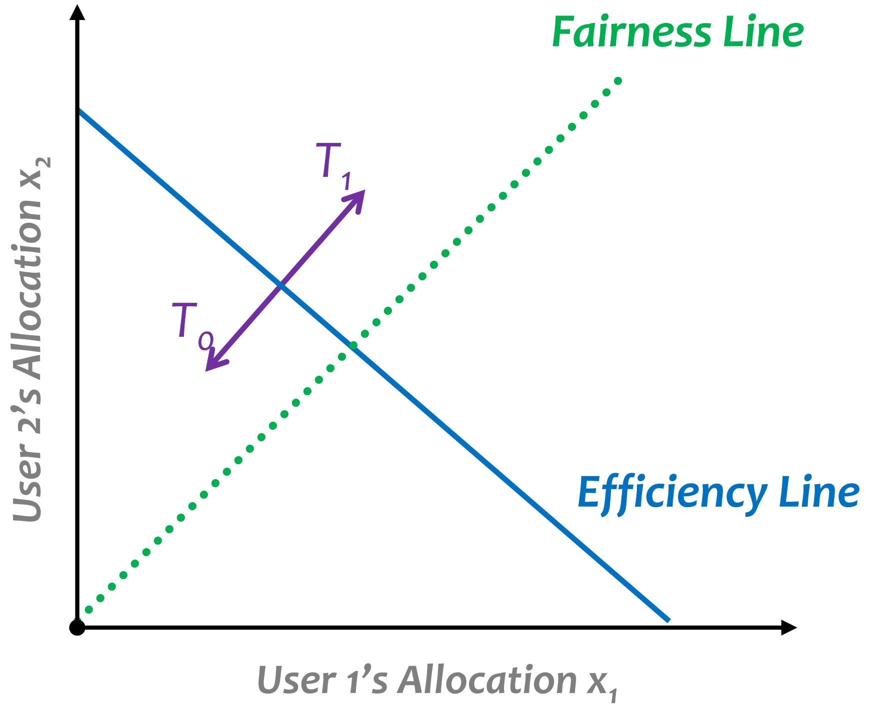
Phase Plot

- *What are desirable properties?*
- *What if flows are not equal?*



Additive Increase or Decrease

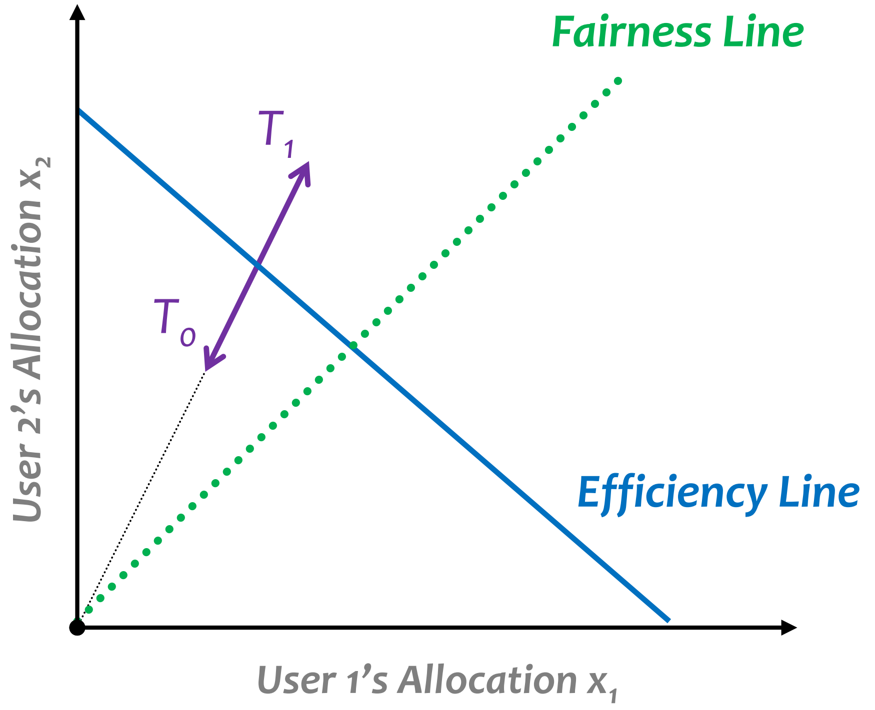
X_1 and X_2 increase or decrease by same amount over time



Multiplicative Increase or Decrease

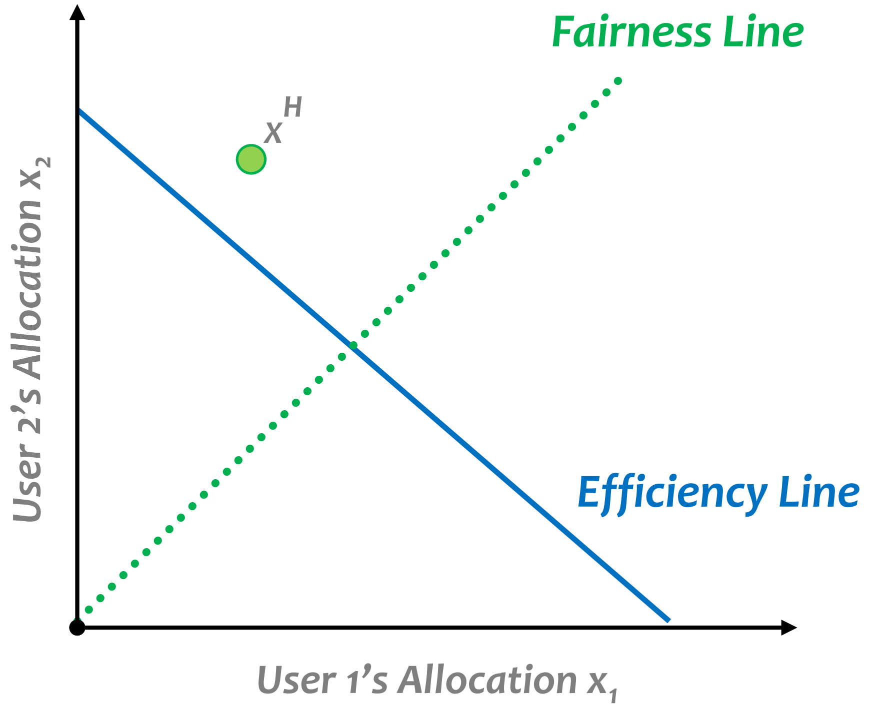
X_1 and X_2 increase or decrease by the same factor

- Extension from origin

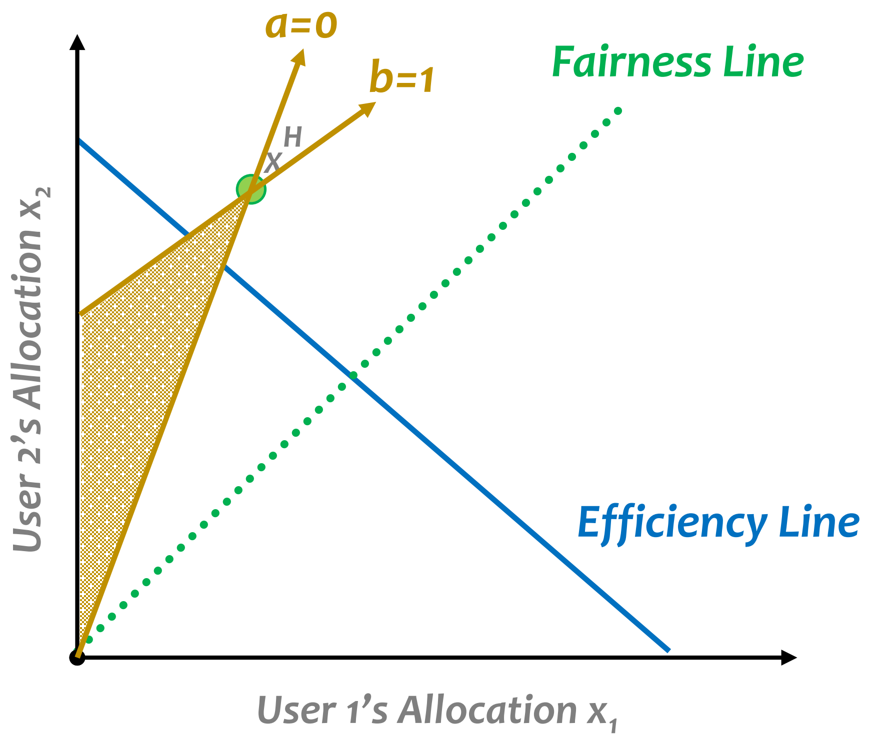


Convergence to Efficiency

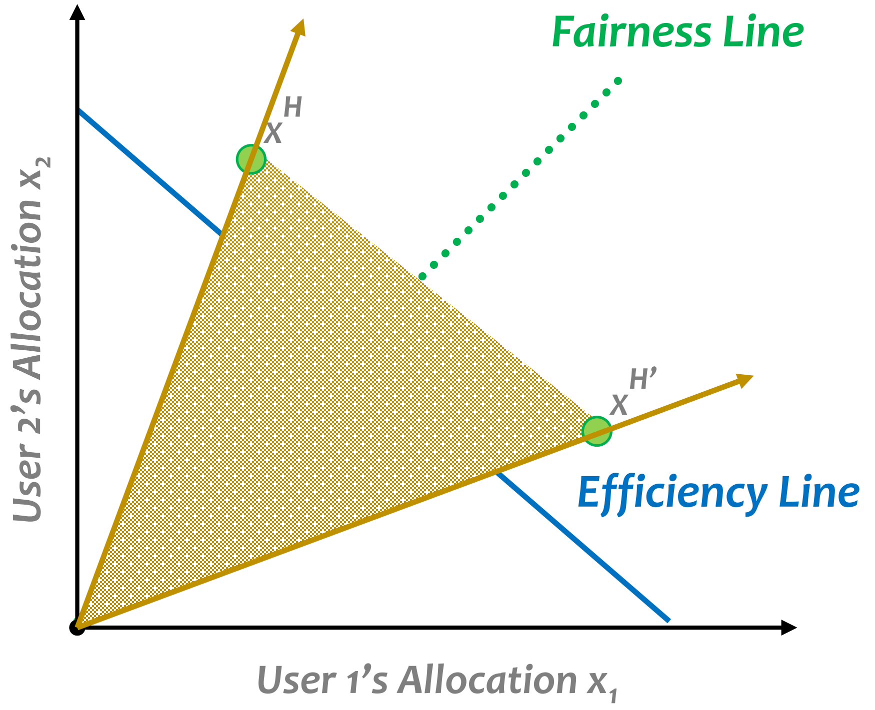
Want to converge quickly to intersection of fairness and efficiency lines



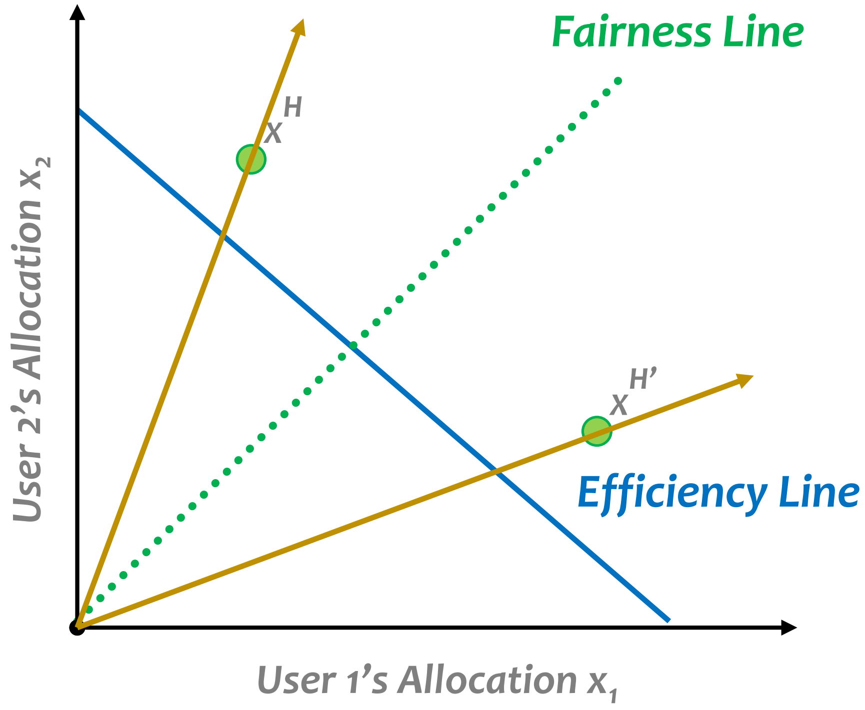
Distributed Convergence to Efficiency



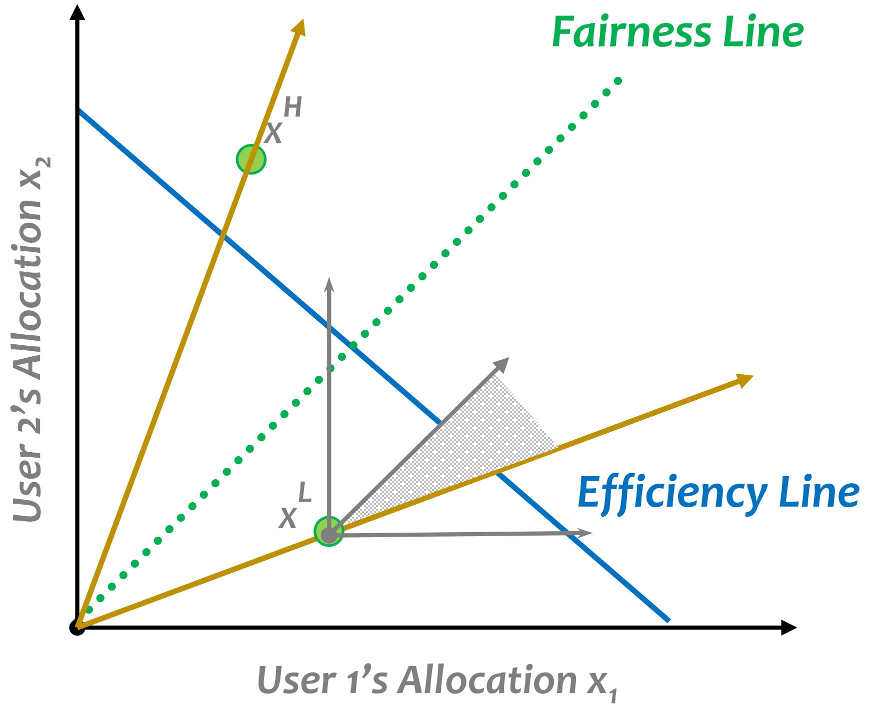
Convergence to Fairness



Convergence to Efficiency and Fairness



Increase

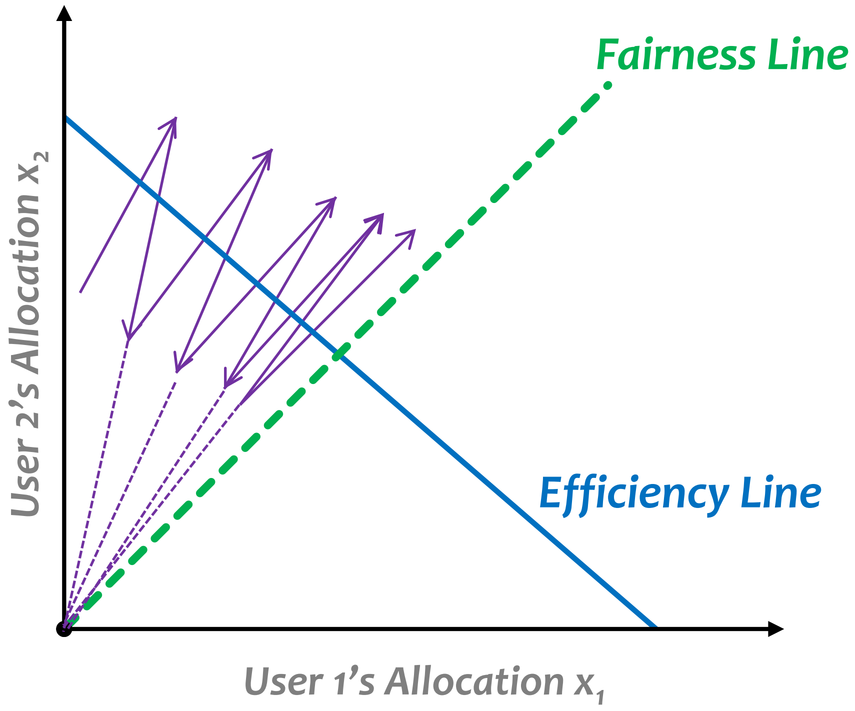


Fairness Line
Efficiency Line
User 2's Allocation x_2
User 1's Allocation x_1
X^H
X^L

What is a good Choice?

Constraints limit us to AIMD

- Can have multiplicative term in increase
- AIMD moves towards optimal point



Outline

- *Connection-oriented* transport: TCP
  - Reliable data transfer
  - Flow control
  - Connection management
- Congestion control
  - Principles
  - Up next: Mechanism