Congestion Control

Congestion control aims to solve three problems

#1	bandwidth estimation	How to adjust the bandwidth of a single flow to the bottleneck bandwidth? could be 1 Mbps or 1 Gbps...
#2	bandwidth adaptation	How to adjust the bandwidth of a single flow to variation of the bottleneck bandwidth?
#3	fairness	How to share bandwidth “fairly" among flows, without overloading the network

Congestion control differs from flow control

Flow control	prevents **one fast sender** from overloading **a slow receiver**
Congestion control	prevents **a set of senders** from overloading **the network**

TCP solves both using two distinct windows

Flow control

prevents one fast sender from overloading a slow receiver

solved using a **receiving window**

Congestion control

prevents a set of senders from overloading the network

solved using a **“congestion” window**

The sender adapts its sending rate based on these two windows

Receiving Window RWND	How many bytes can be sent without overflowing the receiver buffer? based on the receiver input
Congestion Window CWND	How many bytes can be sent without overflowing the routers? based on network conditions
Sender Window	minimum(CWND, RWND)

The 2 key mechanisms of Congestion Control

detecting
congestion

reacting to
congestion

The 2 key mechanisms of Congestion Control

detecting congestion

reacting to congestion

There are essentially three ways to detect congestion

Approach #1

Network could tell the source

but signal itself could be lost

Approach #2

Measure packet delay

but signal is noisy

delay often varies considerably

Approach #3

Measure packet loss

fail-safe signal that TCP already has to detect

There are essentially three ways to detect congestion

Approach #1

Network could tell the source

Best solution - delay and signaling-based methods are hard & risky

but signal is noisy

delay often varies considerably

Approach #3

Measure packet loss

fail-safe signal that TCP already has to detect

Detecting losses can be done using ACKs or timeouts, the two signal differ in their degree of severity

duplicated ACKs	mild congestion signal packets are still making it
timeout	severe congestion signal multiple consequent losses

The 2 key mechanisms of Congestion Control

detecting congestion

reacting to congestion

Remember: congestion control aims to solve three problems

#1	bandwidth estimation	How to adjust the bandwidth of a single flow to the bottleneck bandwidth? could be 1 Mbps or 1 Gbps...
#2	bandwidth adaptation	How to adjust the bandwidth of a single flow to variation of the bottleneck bandwidth?
#3	fairness	How to share bandwidth "fairly" among flows, without overloading the network

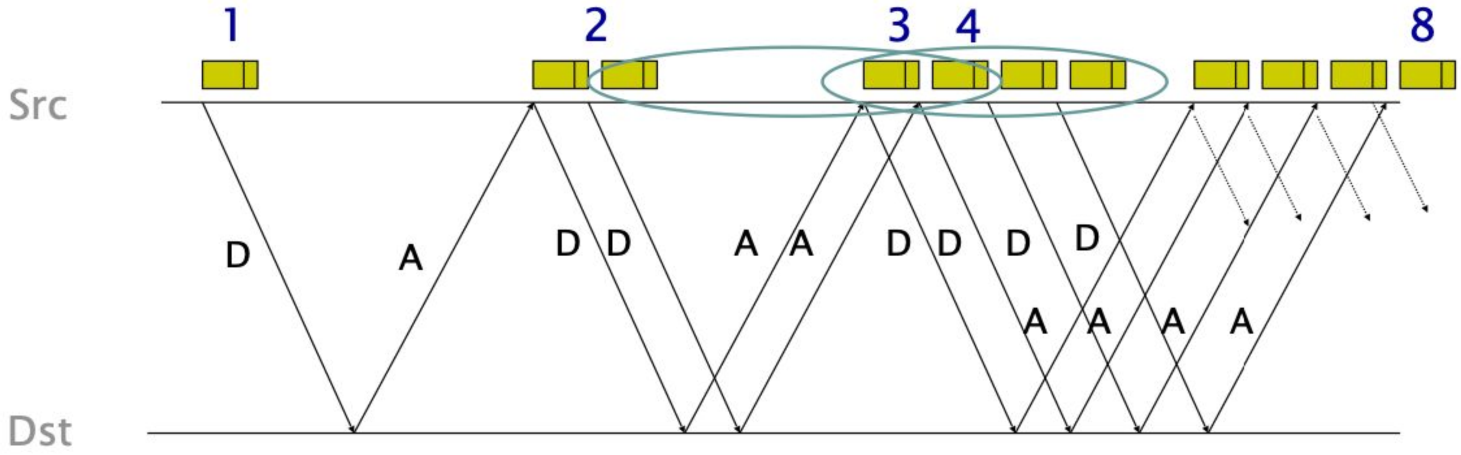
Remember: congestion control aims to solve three problems

#1	bandwidth estimation	How to adjust the bandwidth of a single flow to the bottleneck bandwidth? could be 1 Mbps or 1 Gbps…
#2	bandwidth adaptation	How to adjust the bandwidth of a single flow to variation of the bottleneck bandwidth?
#3	fairness	How to share bandwidth "fairly" among flows, without overloading the network

The goal here is to quickly get a first-order estimate of the available bandwidth

Intuition	Start slow but rapidly increase until a packet drop occurs	
Increase policy	cwnd = 1	initially
	cwnd += 1	upon receipt of an ACK

This increase phase, known as slow start, corresponds to an… exponential increase of CWND



slow start is called like this only because of starting point

The problem with slow start is that it can result in a full window of packet losses

Example	Assume that CWND is just enough to “fill the pipe” After one RTT, CWND has doubled All the excess packets are now dropped
Solution	We need a more gentle adjustment algorithm once we have a rough estimate of the bandwidth

#1	bandwidth **estimation**	How to adjust the bandwidth of a single flow to the bottleneck bandwidth? could be 1 Mbps or 1 Gbps…
#2	bandwidth **adaptation**	How to adjust the bandwidth of a single flow to variation of the bottleneck bandwidth?
#3	**fairness**	How to share bandwidth “fairly" among flows, without overloading the network

The goal here is to track the available bandwidth, and oscillate around its current value

Two possible variations

- **M**ultiplicative **I**ncrease or **D**ecrease

  cwnd = a * cwnd

- **A**dditive **I**ncrease or **D**ecrease

  cwnd = b + cwnd

… leading to four alternative design

The goal here is to track the available bandwidth, and oscillate around its current value

	increase behavior	decrease behavior
AIAD	gentle	gentle
AIMD	gentle	aggressive
MIAD	aggressive	gentle
MIMD	aggressive	aggressive

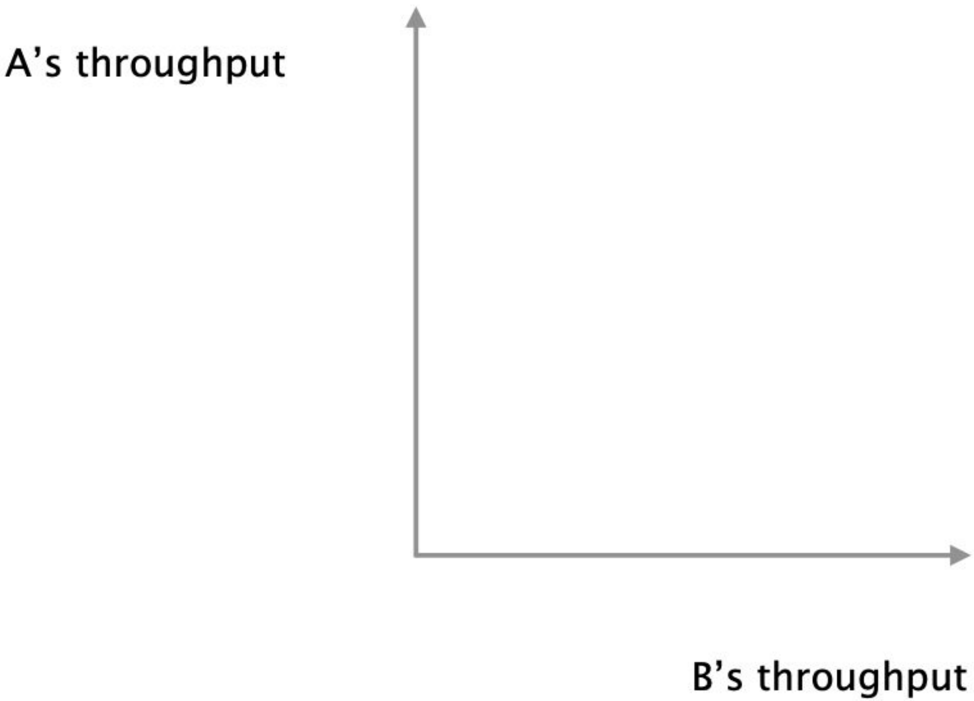
The goal here is to track the available bandwidth, and oscillate around its current value

How do we choose a scheme? Based on fairness

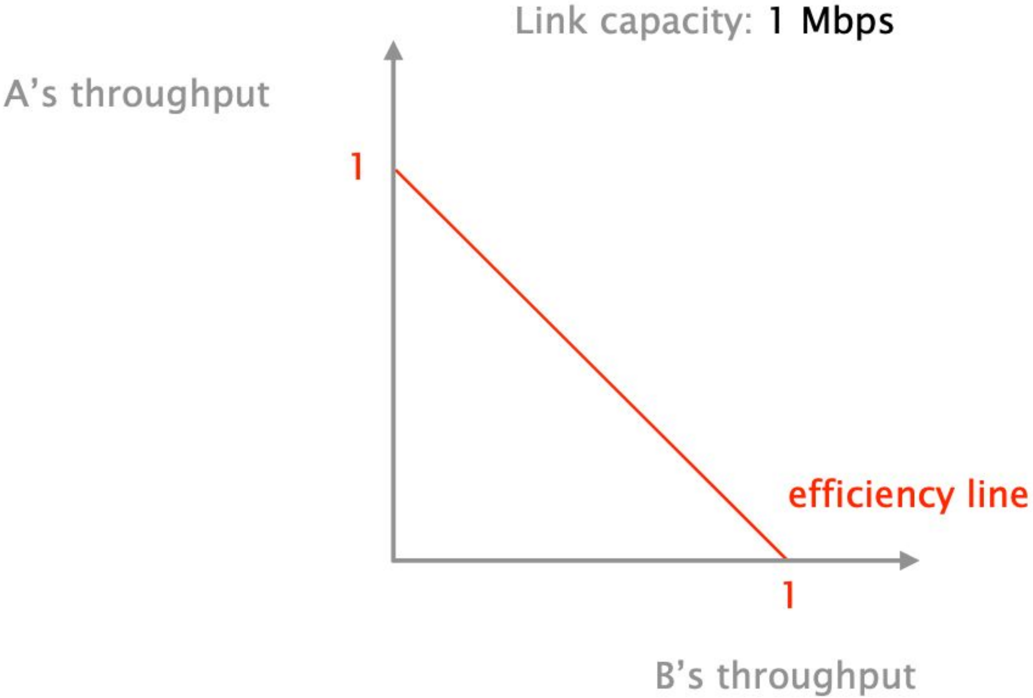
AIAD	gentle	gentle
AIMD	gentle	aggressive
MIAD	aggressive	gentle
MIMD	aggressive	aggressive

TCP notion of fairness: 2 identical flows should end up with the same bandwidth

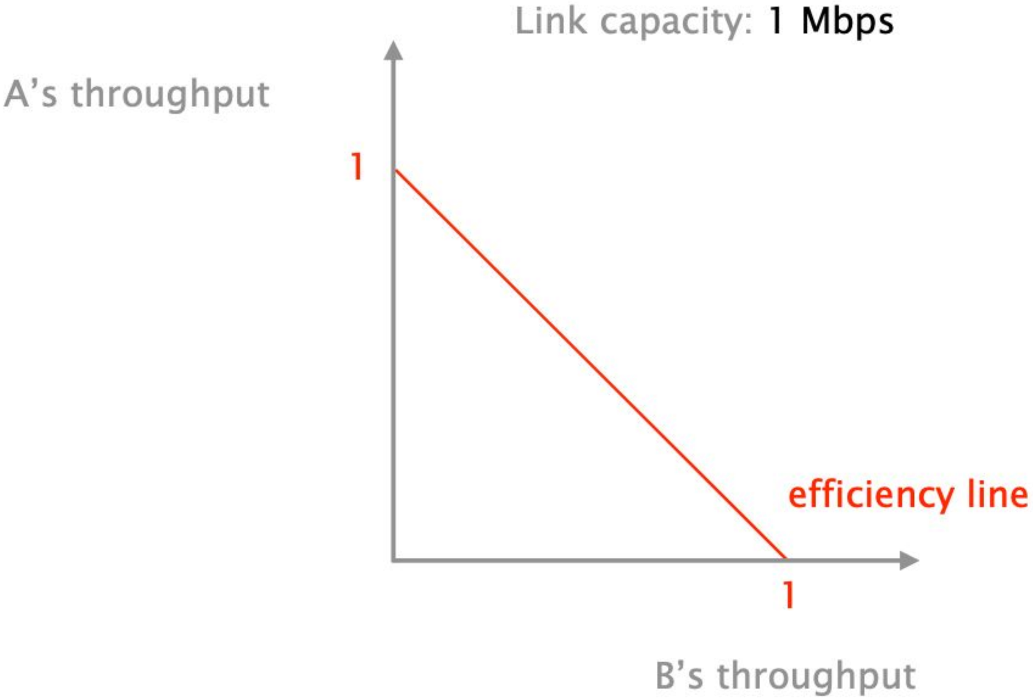
We can analyze the system behavior using a system trajectory plot



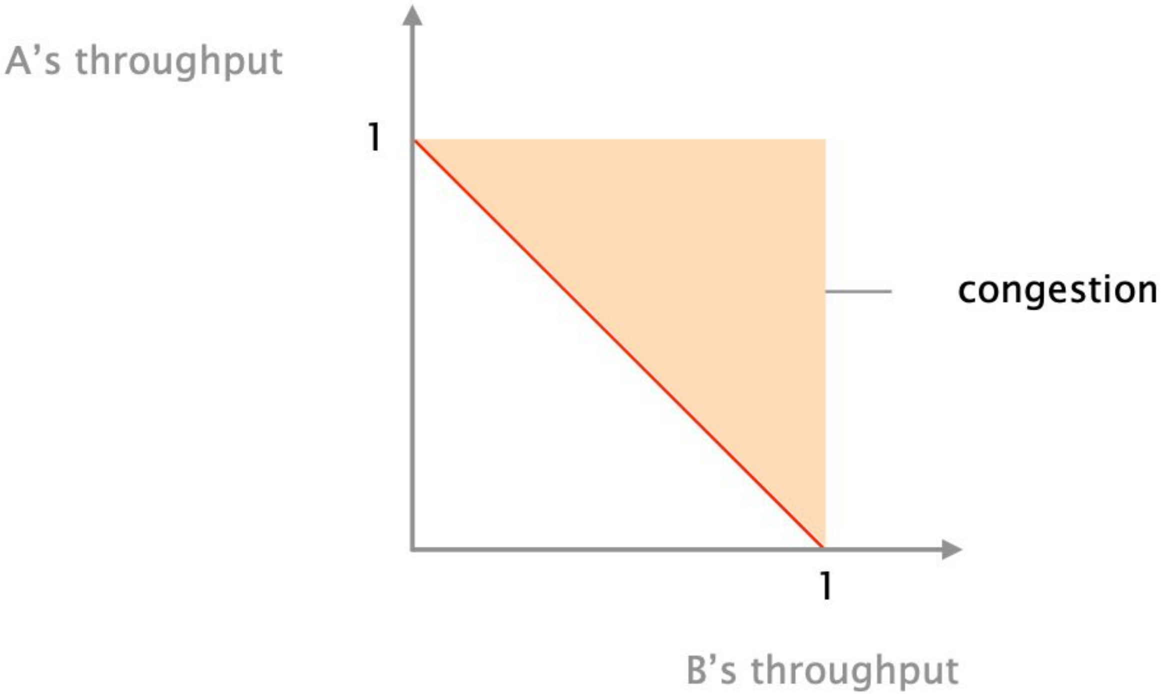
The system is efficient if the capacity is fully used, defining an efficiency line where a + b = 1



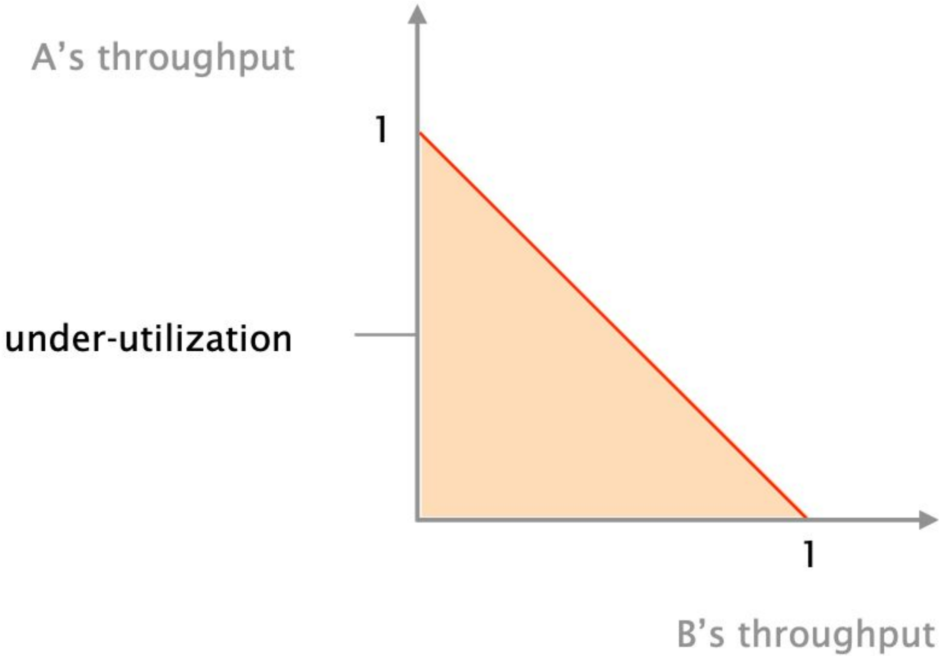
The goal of congestion control is to bring the system as close as possible to this line, and stay there



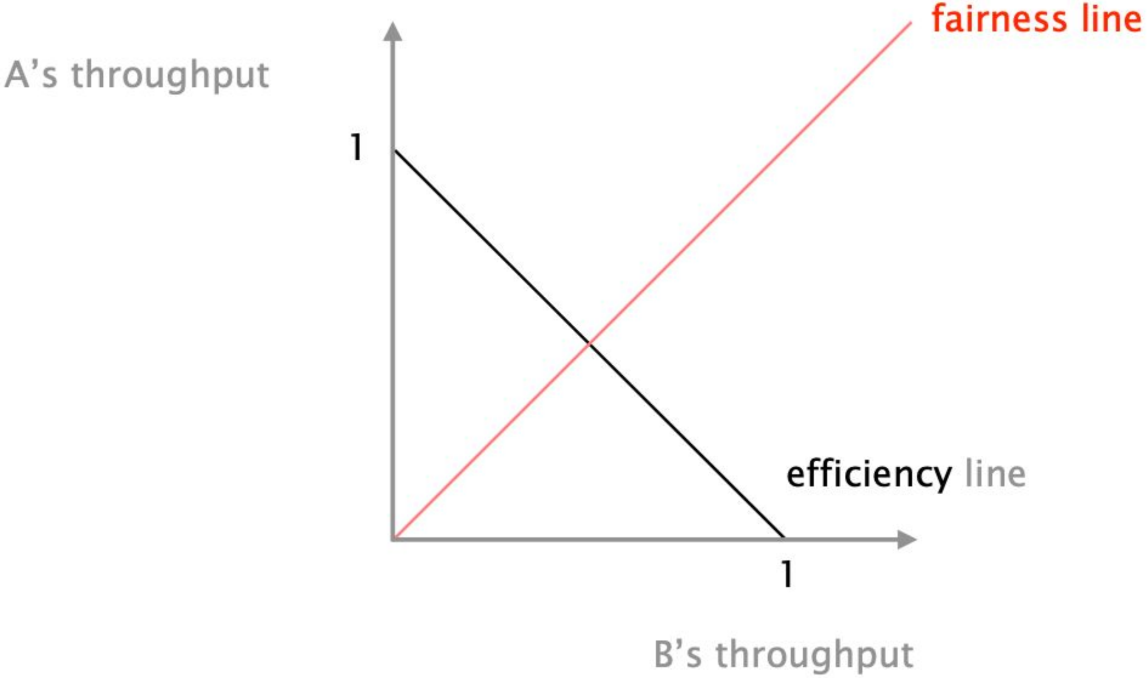
The goal of congestion control is to bring the system as close as possible to this line, and stay there



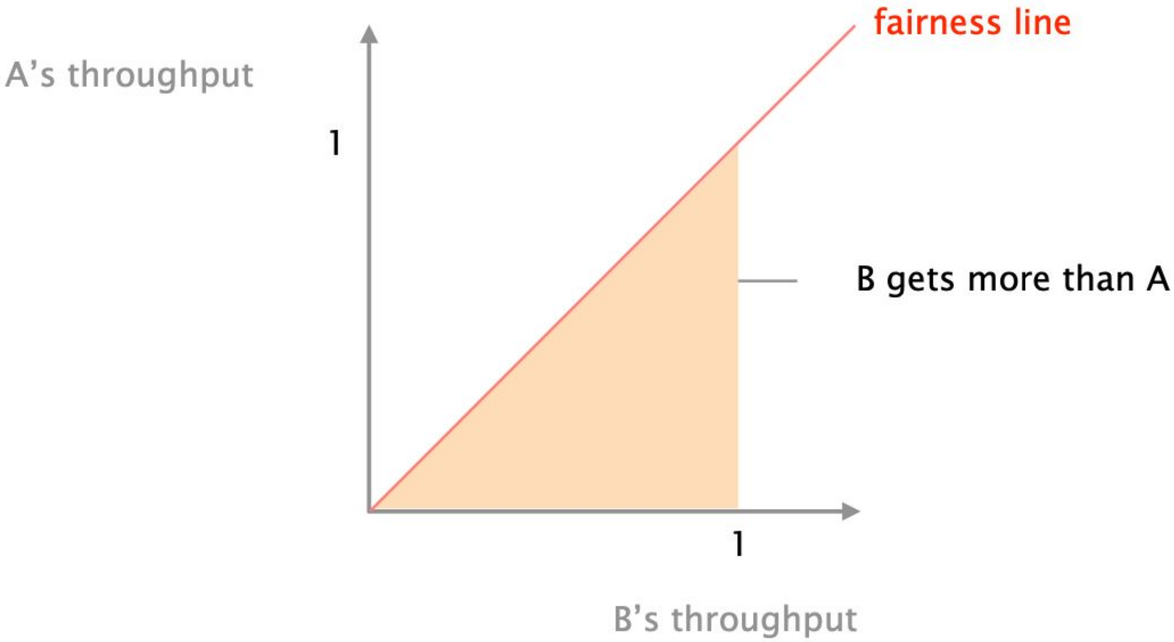
The goal of congestion control is to bring the system as close as possible to this line, and stay there



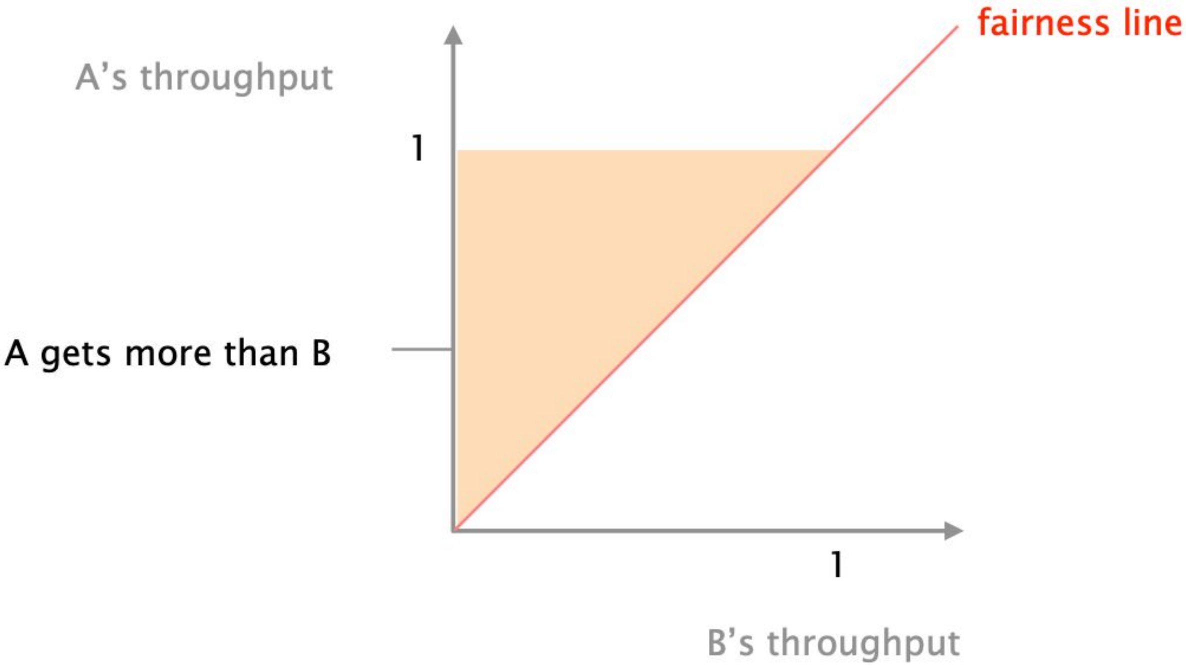
The system is fair whenever A and B have equal throughput, defining a fairness line where a = b

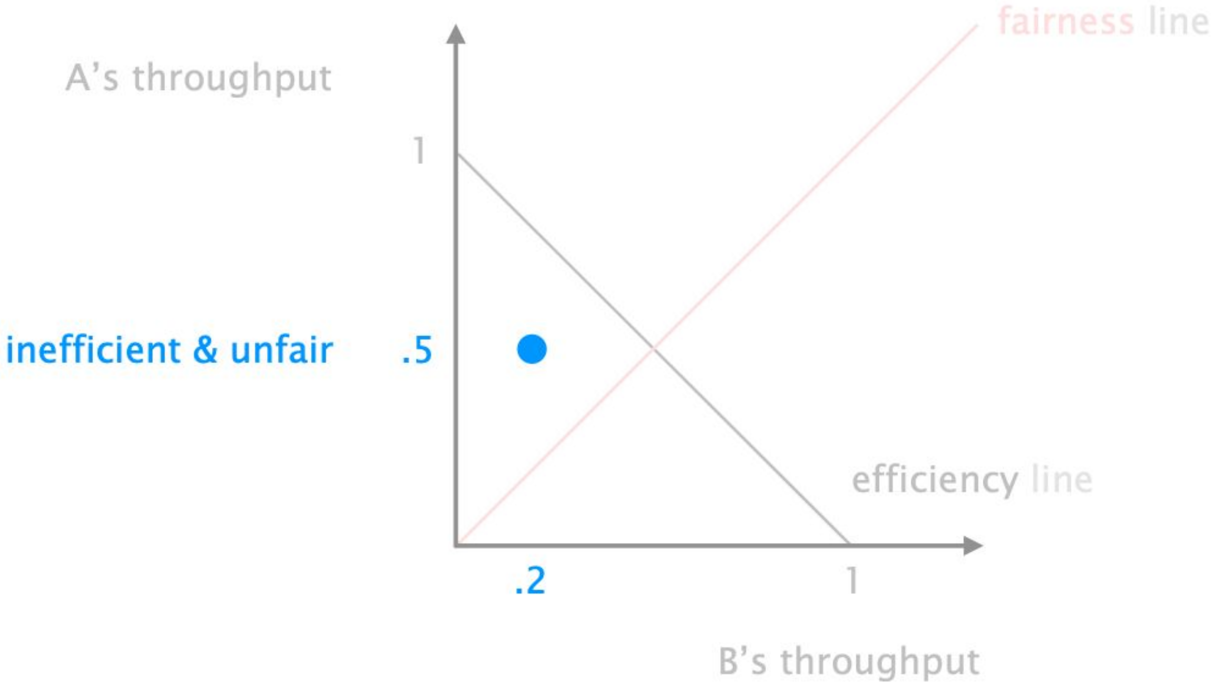


The system is fair whenever A and B have equal throughput, defining a fairness line where a = b

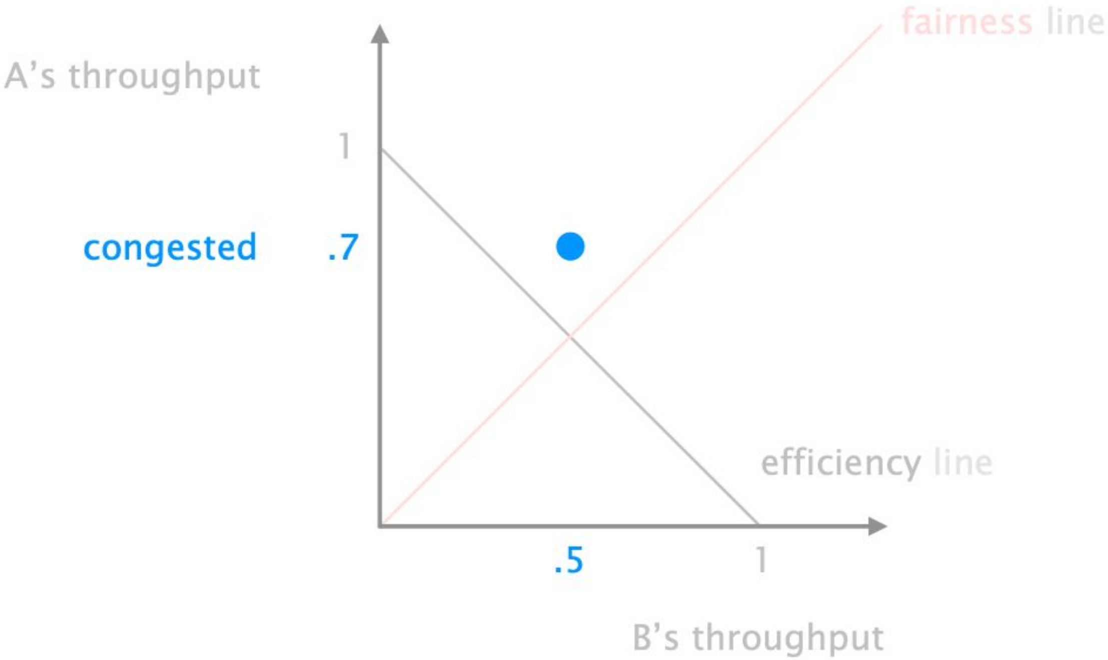


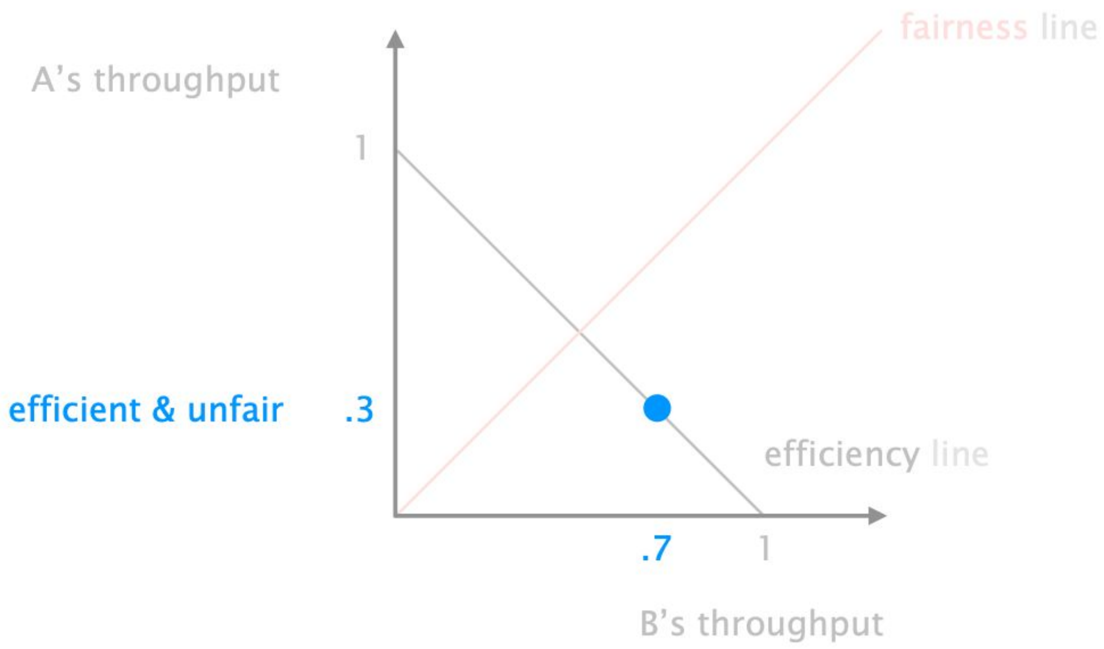
The system is fair whenever A and B have equal throughput, defining a fairness line where a = b

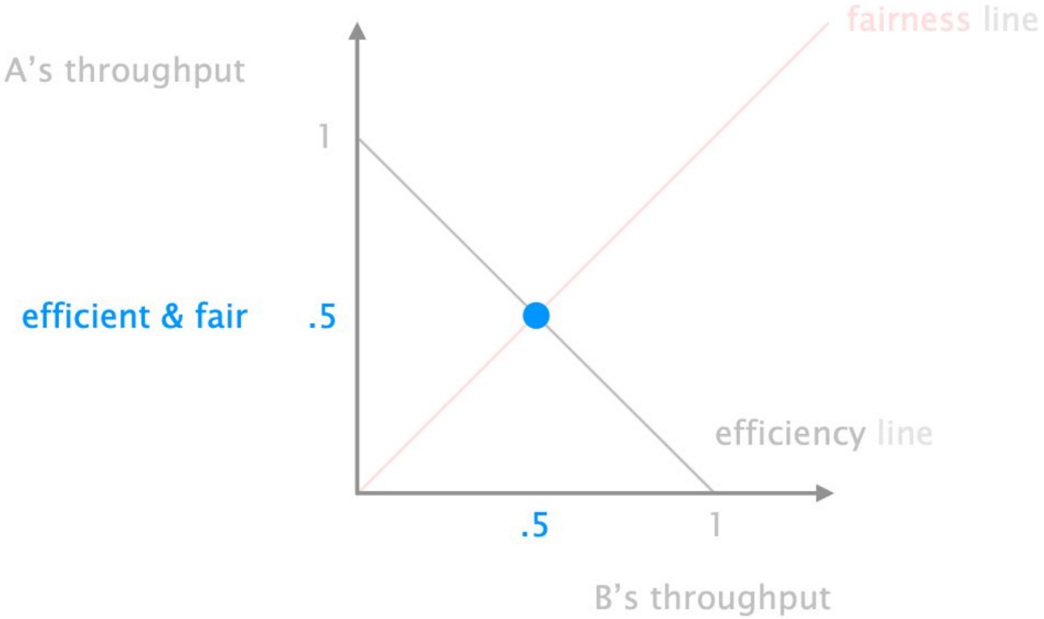




fairness line
A's throughput
1
inefficient & unfair
.5
efficiency line
.2
1
B's throughput

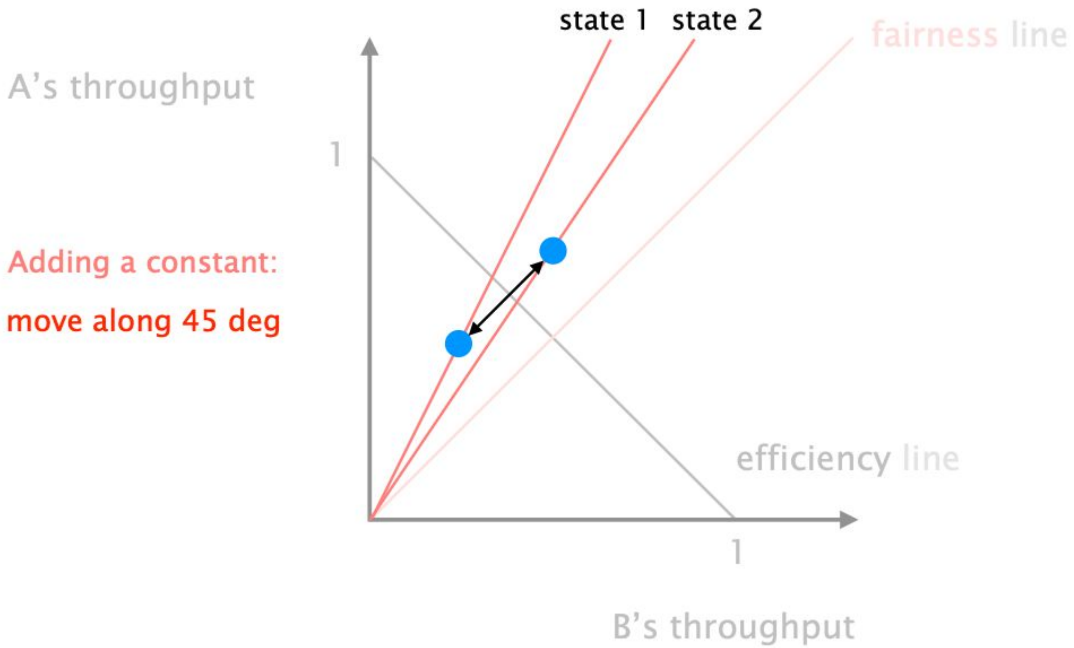

fairness line
A's throughput
1
congested
.7
efficiency line
.5
1
B's throughput


fairness line
A's throughput
1
efficient & unfair
.3
efficiency line
.7
1
B's throughput


fairness line
A's throughput
1
efficient & fair
.5
efficiency line
.5
1
B's throughput

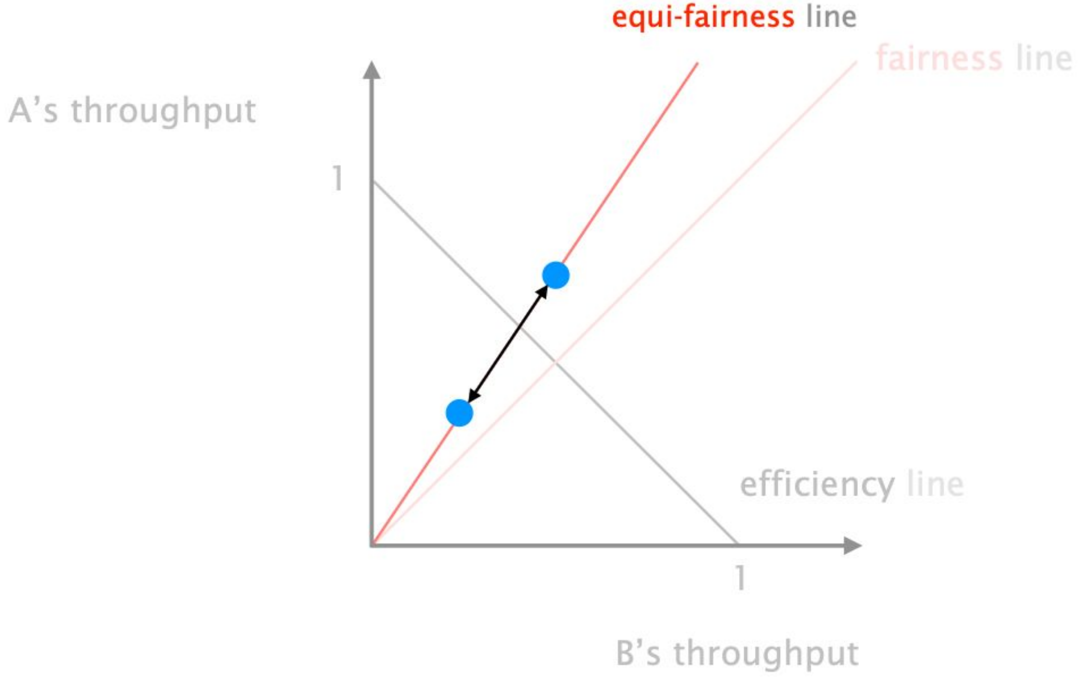
	increase behavior	decrease behavior
AIAD	gentle	gentle
AIMD	gentle	aggressive
MIAD	aggressive	gentle
MIMD	aggressive	aggressive

AIAD does not converge to fairness, nor efficiency: the system fluctuates between two fairness states



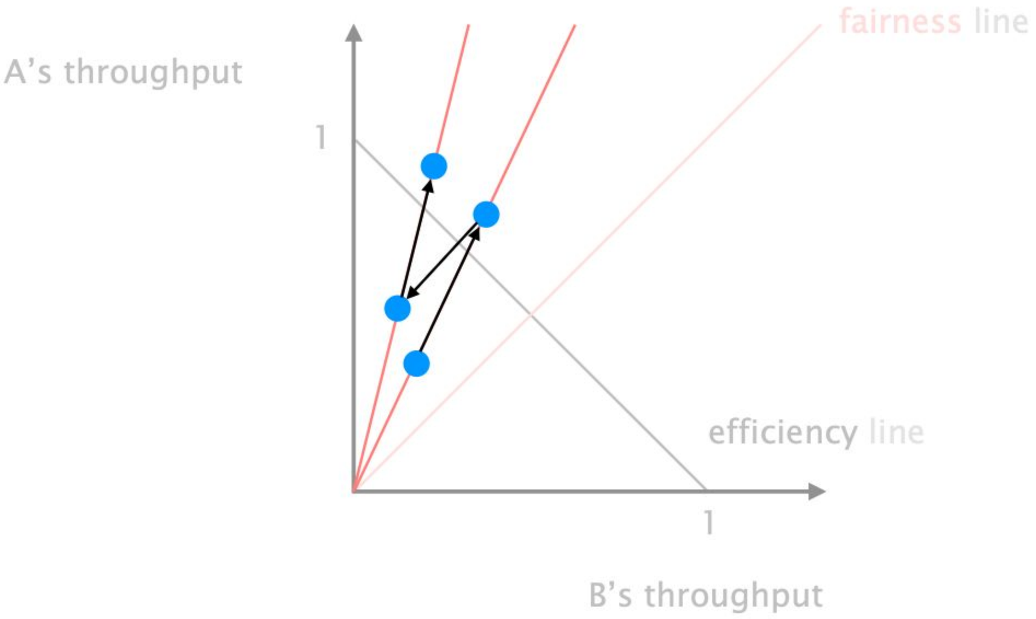
	increase behavior	decrease behavior
AIAD	gentle	gentle
AIMD	gentle	aggressive
MIAD	aggressive	gentle
MIMD	aggressive	aggressive

MIMD does not converge to fairness, nor efficiency: the system fluctuates along a equi-fairness line



	increase behavior	decrease behavior
AIAD	gentle	gentle
AIMD	gentle	aggressive
MIAD	aggressive	gentle
MIMD	aggressive	aggressive

MIAD converges to a totally unfair allocation, favoring the flow with a greater rate at the beginning

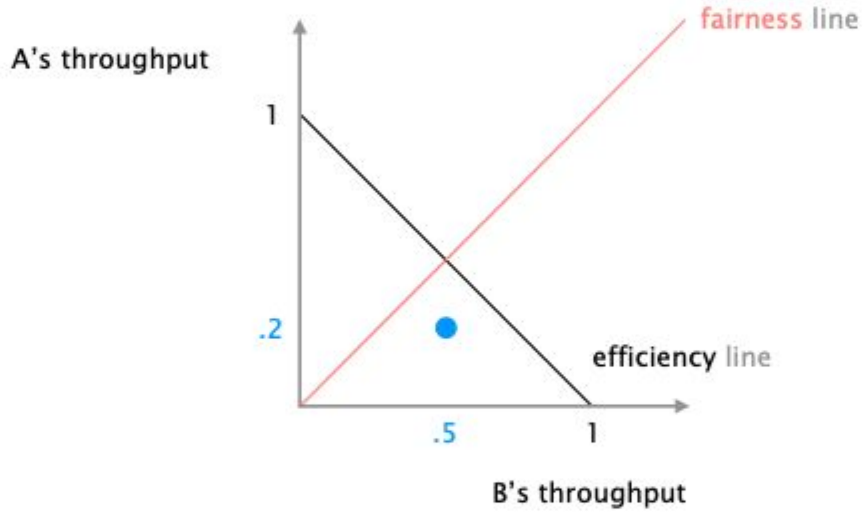


Congestion control exercise

Consider the situation in which two hosts, A and B, are concurrently using a 1 Mbps link with a Maximum Segment Size (MSS) of 100 kb.

Assuming that B starts with 500 kbps and A with 200 kbps (see left picture).

What would happen if both are using MIAD (assume both senders double their CWND MSS when there is no congestion and subtract it by 1 upon congestion).

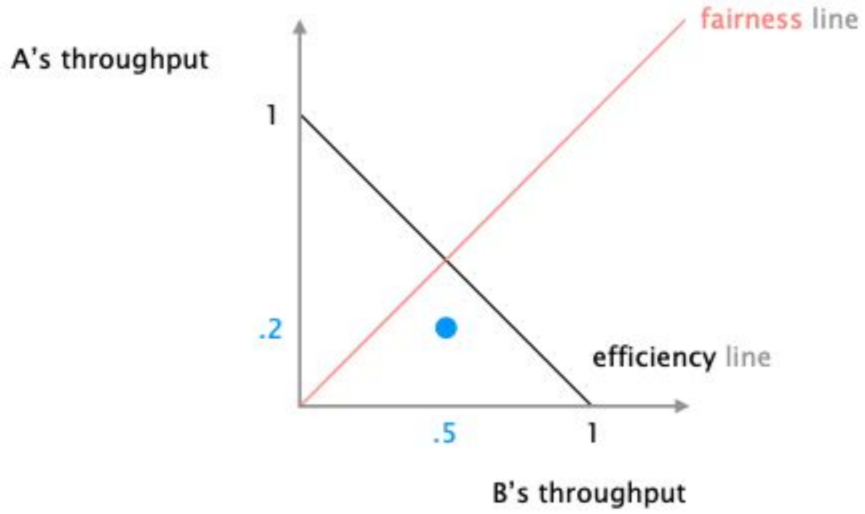


Congestion control exercise

Solution

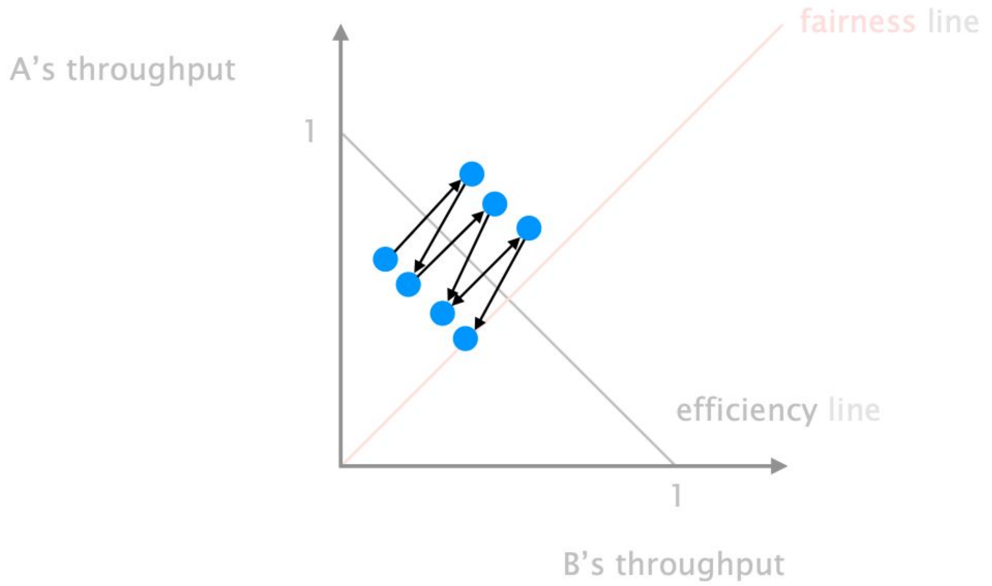
1. (.2, .5)
2. (.4, 1) > congestion!
3. (.3, .9) > congestion!
4. (.2, .8)
5. (.4, 1.6) > congestion!
6. (.3, 1.5) > congestion!
7. (.2, 1.4) > congestion!
8. (.1, 1.3) > congestion!
9. (0, 1.2) > congestion!
10. (0, 1.1) > congestion!
11. (0, 1)

The sender which benefits from a bigger initial share will end up using the entire link.



	increase behavior	decrease behavior
AIAD	gentle	gentle
AIMD	gentle	aggressive
MIAD	aggressive	gentle
MIMD	aggressive	aggressive

AIMD converge to fairness and efficiency, it then fluctuates around the optimum (in a stable way)



AIMD converge to fairness and efficiency, it then fluctuates around the optimum (in a stable way)

Intuition

During increase,
both flows gain bandwidth at the same rate

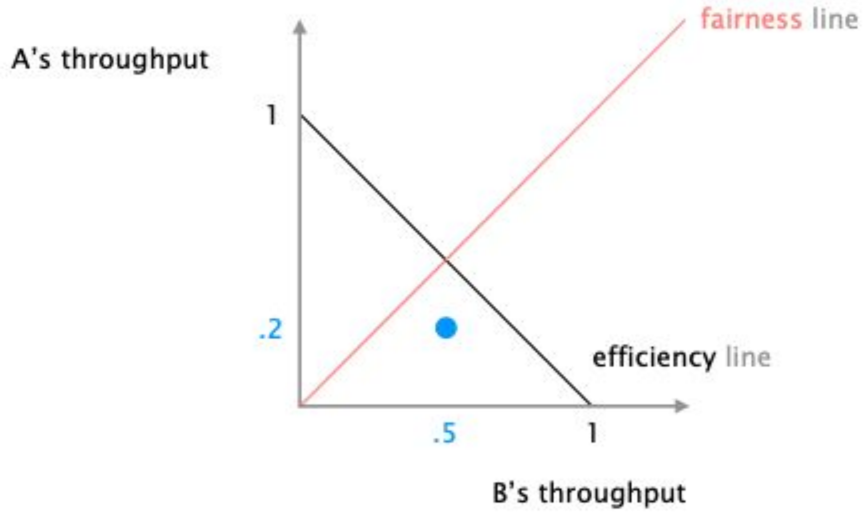
During decrease,
the faster flow releases more

Congestion control exercise

Consider the situation in which two hosts, A and B, are concurrently using a 1 Mbps link with a Maximum Segment Size (MSS) of 100 kb.

Assuming that B starts with 500 kbps and A with 200 kbps (see left picture).

What would happen if both are using AIMD (assume both senders increase their CWND by 1 MSS when there is no congestion and divide it by 2 upon congestion).



Congestion control exercise

Solution

1. (.3, .6)
2. (.4, .7) > congestion!
3. (.2, .35)
4. (.3, .45)
5. (.4, .55)
6. (.5, .65) > congestion!
7. (.25, .325)

Because of its bigger share, B loses more than A because of the halving, eventually the system converges along the fairness line.