Congestion Control In The Internet

Part 1:

Contents

1. What is the problem; congestion collapse
2. Efficiency vs Fairness
3. Definitions of fairness
4. Additive Increase Multiplicative Decrease (AIMD)
5. Slow start


Rate adaptation, Congestion Control and Fairness: A Tutorial

JEAN-YVES LE BOUDEC
Ecole Polytechnique Fédérale de Lausanne (EPFL)

No book; check the pdf uploaded on Moodle: "Rate adaptation, Congestion Control and Fairness: A tutorial"

1. Congestion Collapse

In October of '86, the Internet had the first of what became a series of 'congestion collapses'. During this period, the data throughput from LBL to UC Berkeley (sites separated by 400 yards and three IMP hops) dropped from 32 Kbps to 40 bps. Mike Karels[1] and I were fascinated by this sudden factor-of-thousand drop

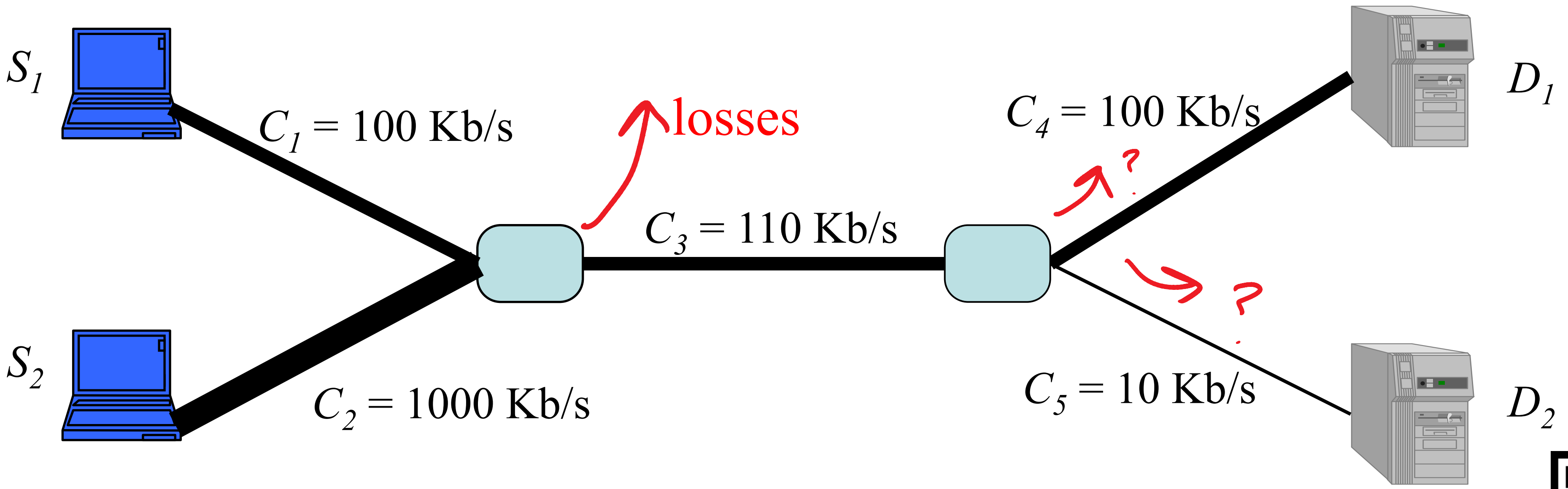
Jacobson, Van. "Congestion avoidance and control." *ACM SIGCOMM computer communication review*. Vol. 18. No. 4. ACM, 1988.

Example 1: Congestion due to greedy sources

Assume: Two flows $S_1 \rightarrow D_1$ and $S_2 \rightarrow D_2$.

Sources are greedy (i.e. send as much as they want); loss may happen

Loss is proportional to submitted traffic and links can be fully utilized



What is the max throughput attained by flow $S_1 \rightarrow D_1$?

A. 10 kb/s B. 50 kb/s. C. 100 kb/s. D. I don't know

Go to web.speakup.info or download speakup app

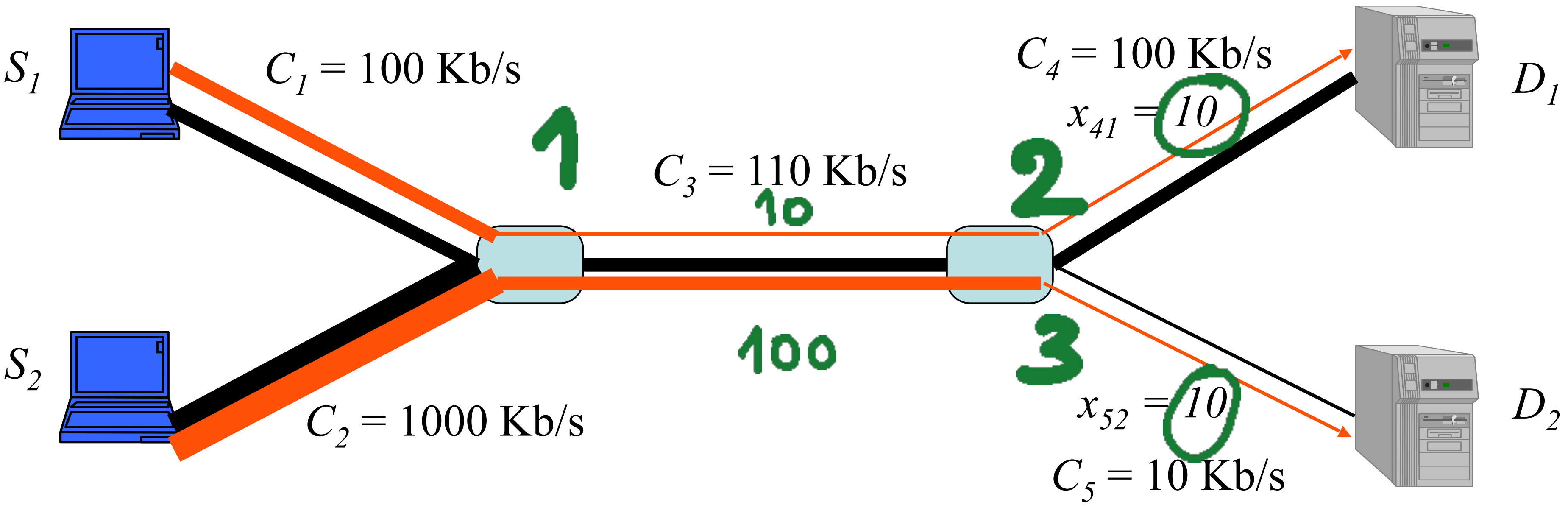
Solution

Answer A: Ratio of traffic that survives at 1: $110/(100+1000) = 10\,\%$

Ratio of traffic that survives at 2: 100%

Ratio of traffic that survives at 3: $10/100 = 10\,\%$

Both flows attain 10 kb/s even if the sources have different access links and send at different rates!



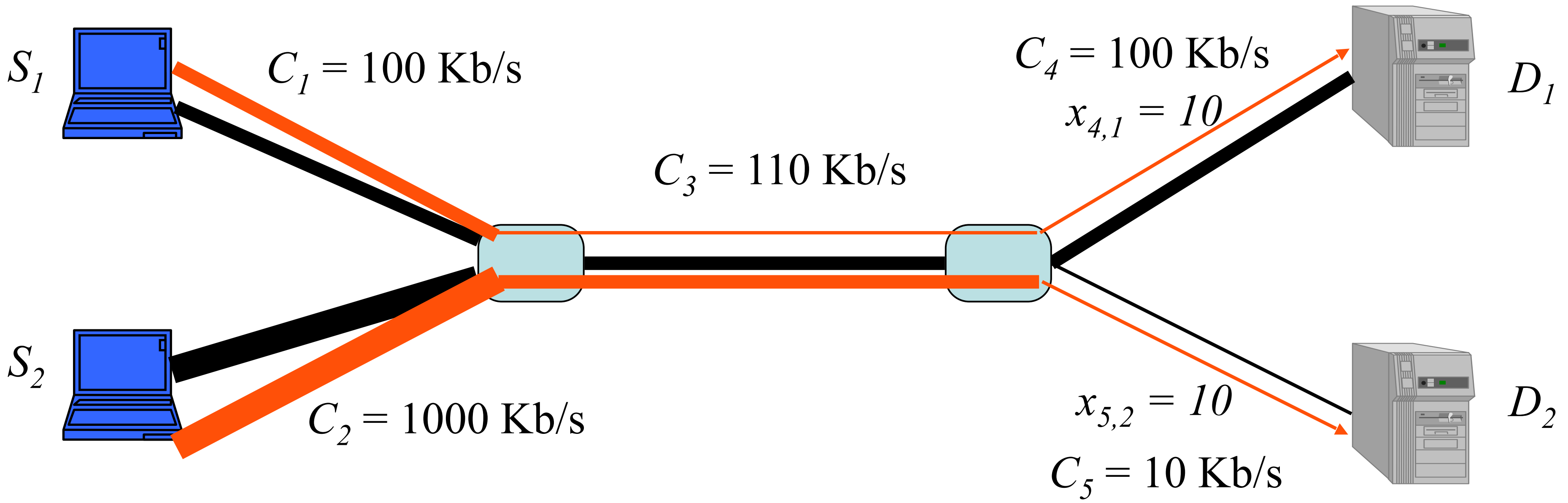
Take-home message 1: greedy sources may be *inefficient*

A better allocation is:

S_1: 100 kb/s

S_2: 10 kb/s

The problem was that S_2 sent too much (but it did not know)



Example 2: Congestion collapse

Assume: Each source sends traffic 2 hops away at rate λ

e.g., source i, at node i, sends traffic to a destination at node $i+2$

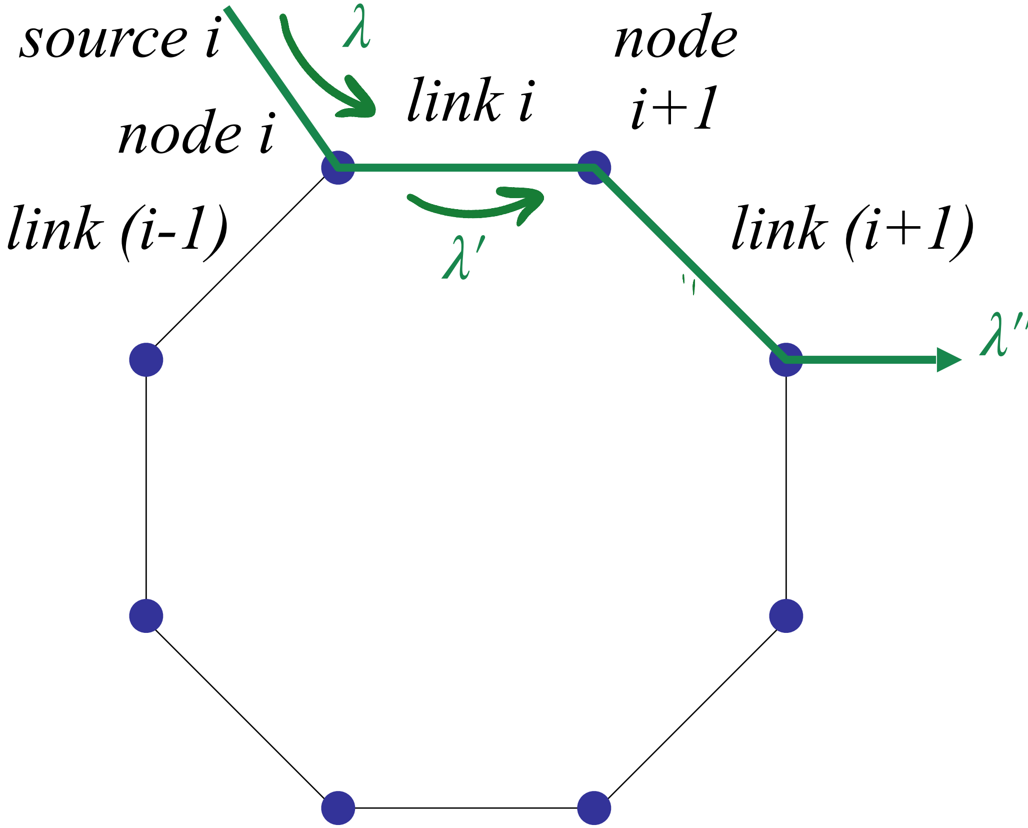
All links have the same capacity c

Loss is proportional to submitted traffic and links can be fully utilized

Let:

- λ' = the survived rate of λ at the *first* link
- λ'' = the survived rate of λ' at the *second* link = throughput

➡ *How much is λ'' (i.e. the rate at which destination $i+2$ receive traffic)?*
How much throughput each source can attain?



Solution: Attained throughput λ''

At each link i, the submitted traffic is $\lambda + \lambda'$

- If $\lambda < \dfrac{c}{2}$ there is no loss; so $\lambda'' = \lambda$
- If $\lambda > \dfrac{c}{2}$ there is loss:

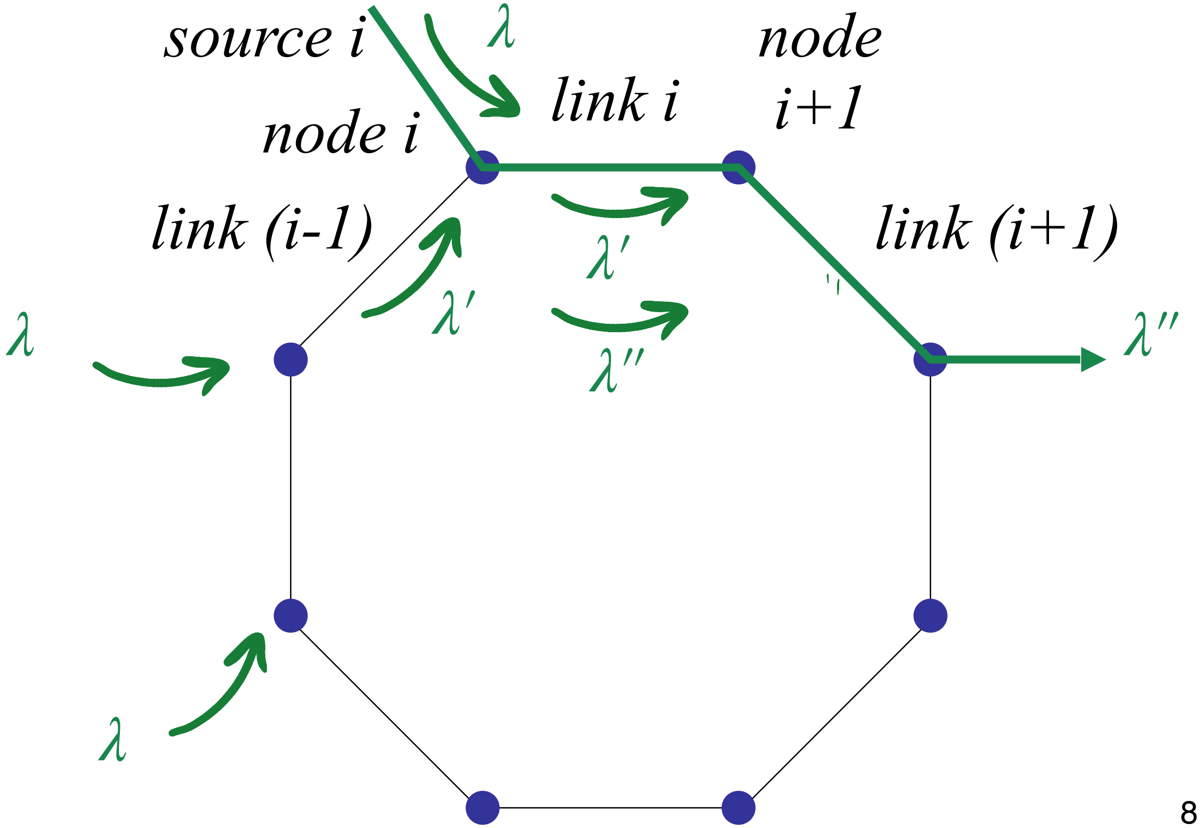
Traffic survival ratio at each node = $\dfrac{c}{\lambda + \lambda'}$

Due to loss proportionality:

$$\lambda' = \frac{c}{\lambda + \lambda'}\lambda \quad (1)$$

$$\lambda'' = \frac{c}{\lambda + \lambda'}\lambda' \quad (2)$$

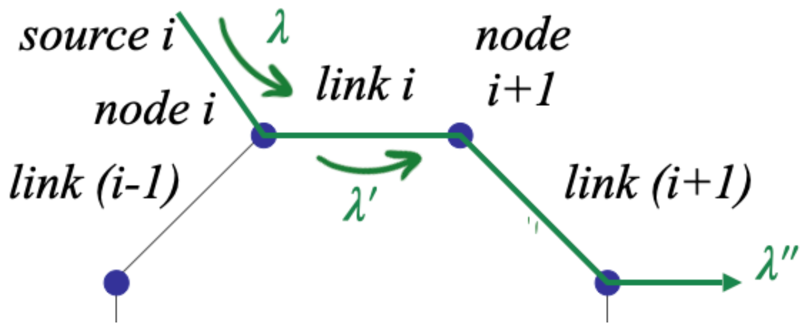
We solve (1) for λ',
plug into (2)
and obtain a solution for λ''



For large offered traffic load λ, the limit of throughput is 0

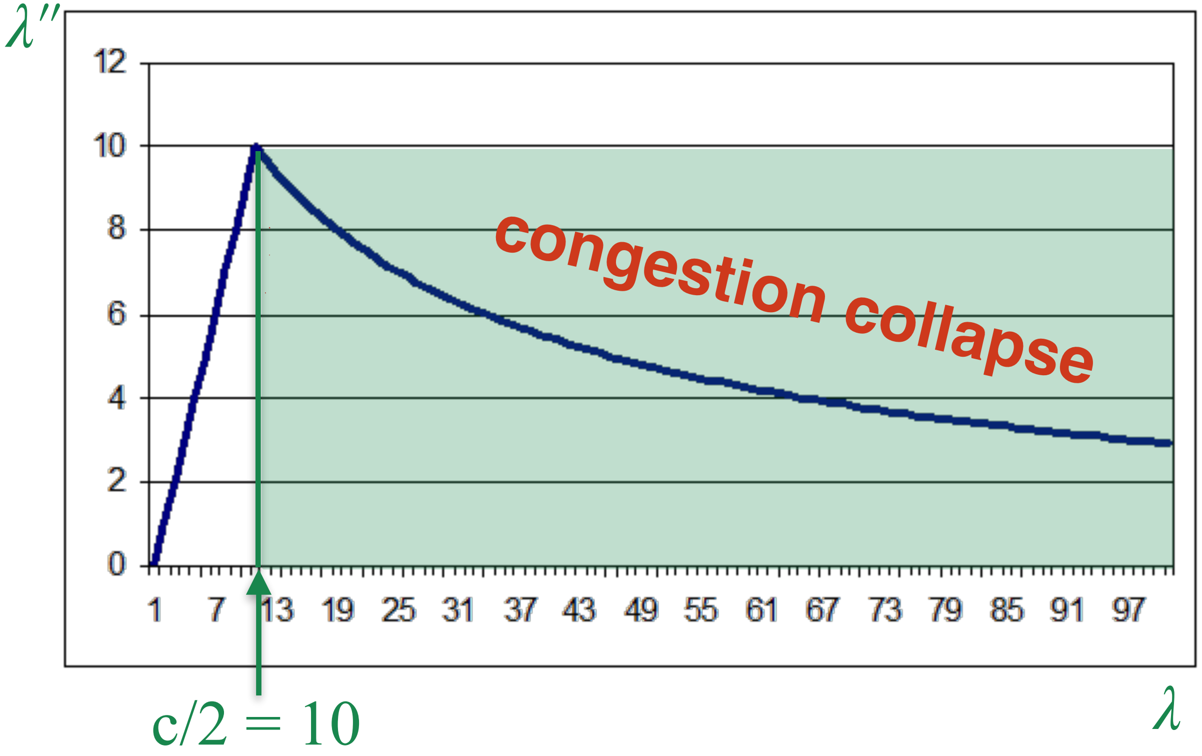
If $\lambda > \frac{c}{2}$, solving the system returns:

$$\lambda'' = c - \frac{\lambda}{2}\left(-1 + \sqrt{1 + \frac{4c}{\lambda}}\right)$$



$$\sqrt{1+u} = 1 + \frac{1}{2}u - \frac{1}{8}u^2 + o(u^2)$$

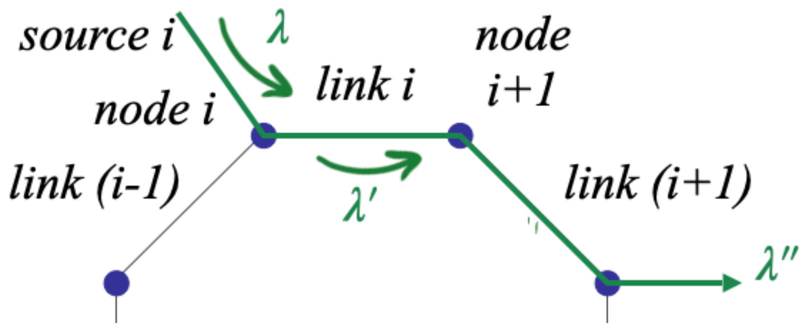
$$\lambda'' = \frac{c^2}{\lambda} + o\left(\frac{1}{\lambda}\right)$$

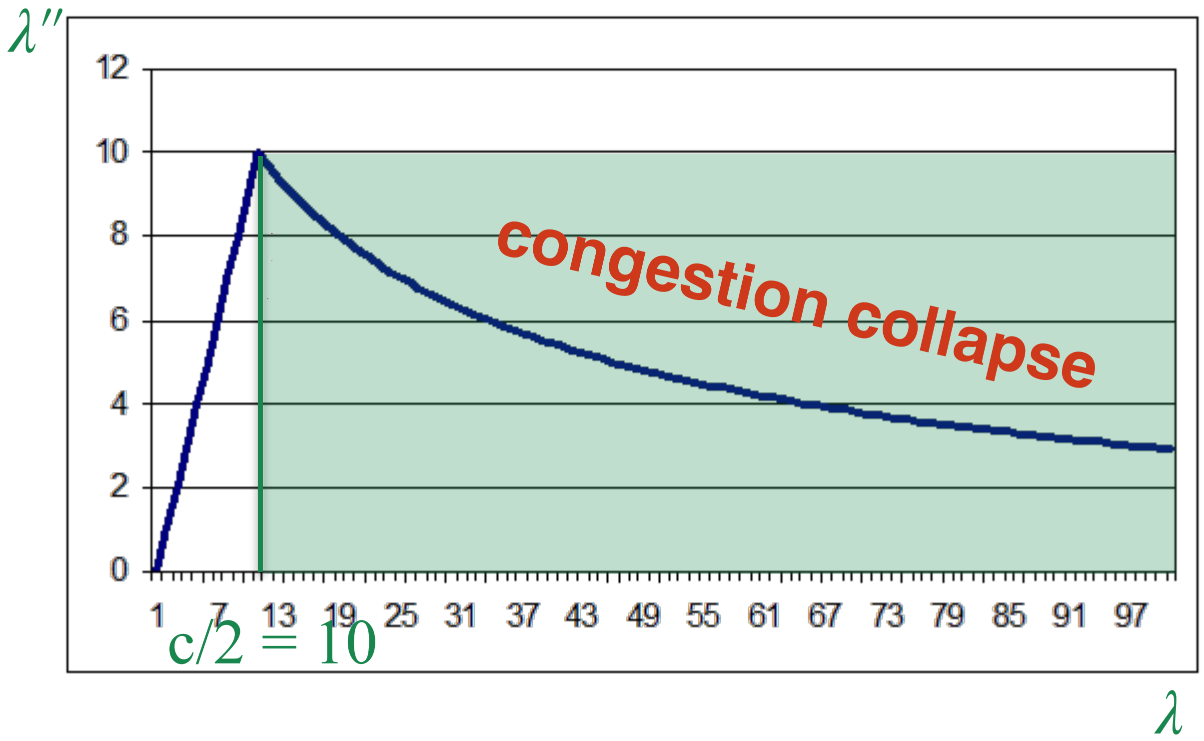


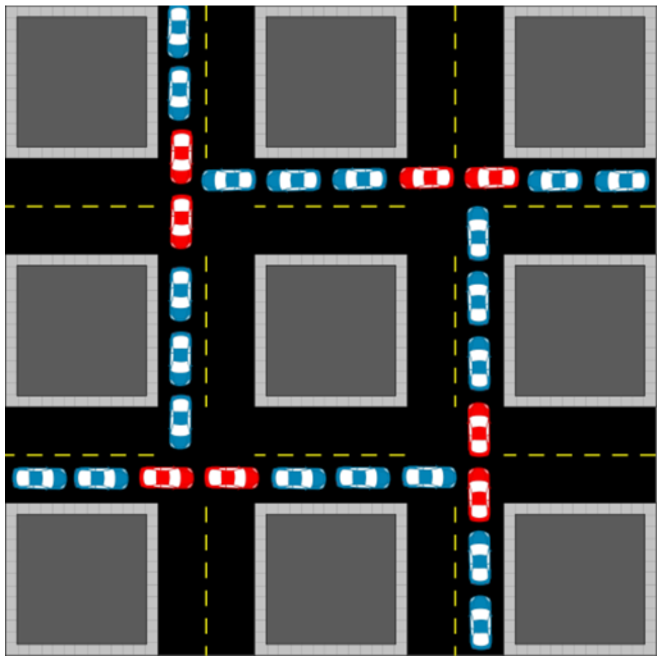
So, as $\lambda \rightarrow +\infty$, throughput $\rightarrow 0$

Take-Home Message 2: *Congestion collapse*

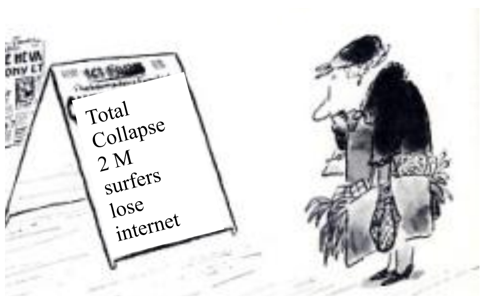
- As the offered *load increases*, *throughput decreases*, may even go to 0
- Sources must limit their sending rates and *adapt* to network conditions; otherwise inefficiency or congestion collapse may occur







2. Efficiency vs Fairness

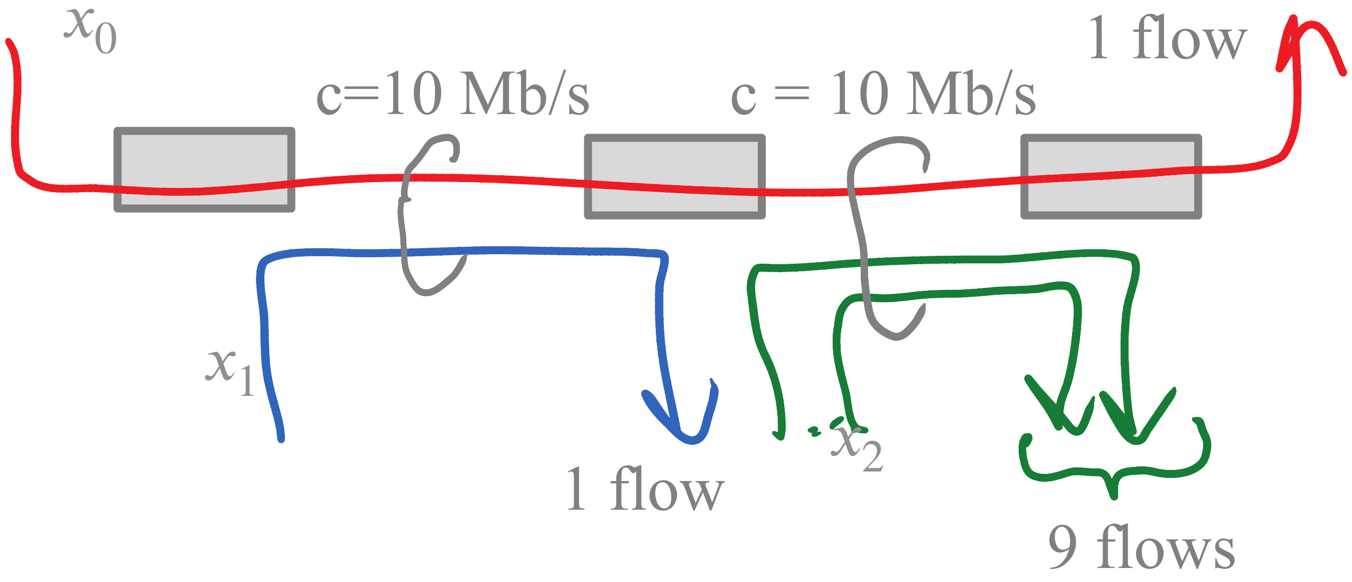


- Being *maximally efficient* may lead to *unfairness*
- A network should be organized to avoid both:

  inefficiency and unfairness

Example: what is the maximum aggregate throughput in this network ?

A. 5 Mb/s
B. 10 Mb/s
C. 20 Mb/s
D. None of the above
E. I don't know

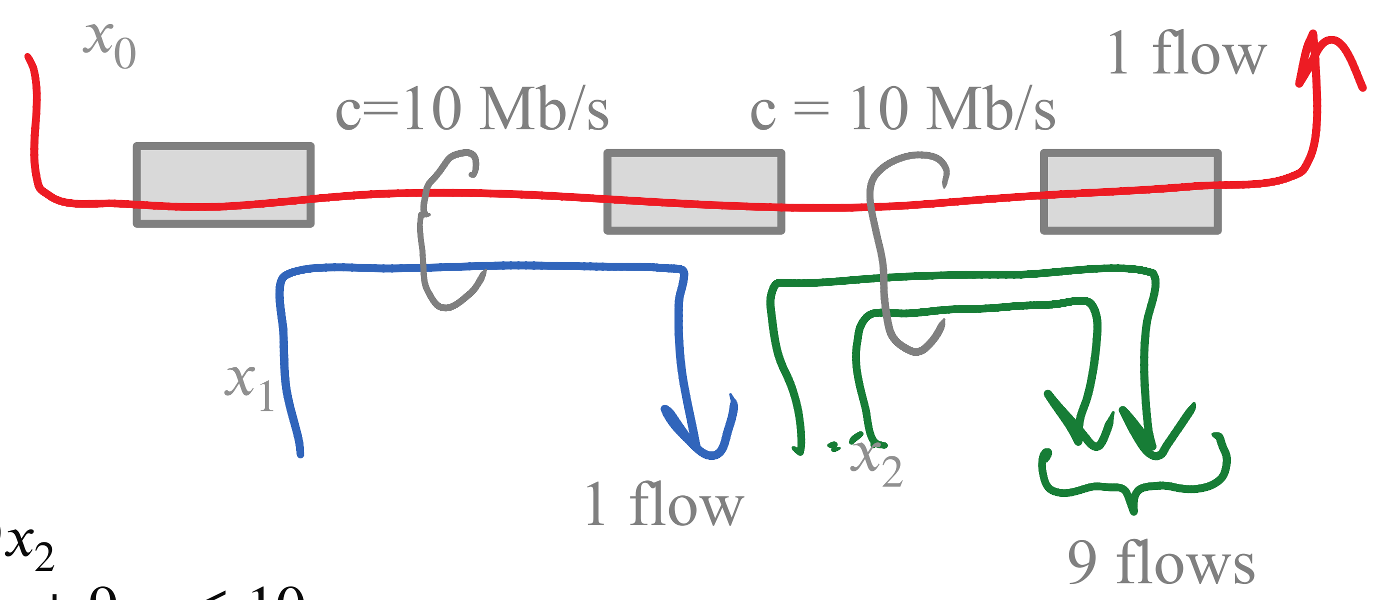


Go to web.speakup.info or download speakup app

Join room 87072

Solution

Answer C



Total throughput $\theta = x_0 + x_1 + 9x_2$

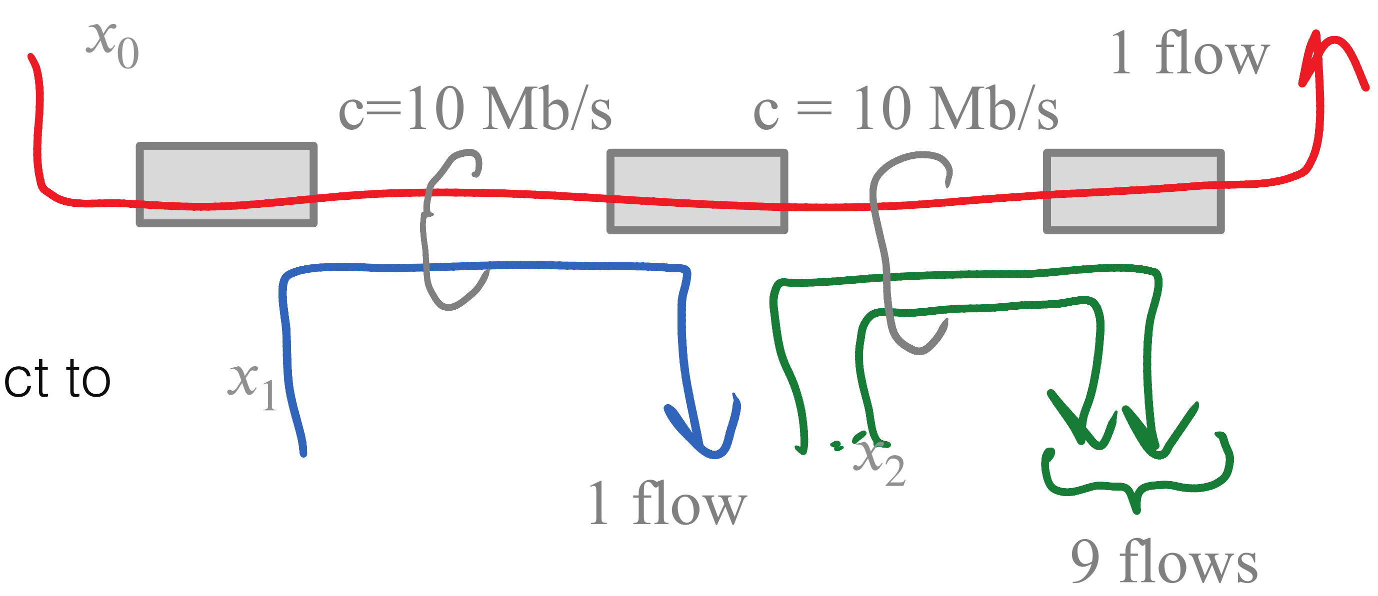
Maximize $\theta = x_0 + x_1 + 9x_2$
subject to $x_0 + x_1 \leq 10,\ x_0 + 9x_2 \leq 10$
over $x_0 \geq 0,\ x_1 \geq 0\ x_2 \geq 0$

The max can be obtained *by linear programming*, or directly here *by inspection*:

- $\theta \leq\ 20$ because $x_0 + x_1 \leq 10,\ 9x_2 \leq 10 - x_0$ and $x_0 \geq 0$
- $\theta = 20$ is achieved with $x_1 = 10$ and $x_2 = 10/9$

therefore the max is 20 Mb/s

Solution



And we can also prove that this is the only maximizing allocation:

Find all $x_0 \geq 0$ $x_1 \geq 0$ $x_2 \geq 0$ subject to

$x_0 + x_1 \leq 10$ (1)

$x_0 + 9x_2 \leq 10$ (2)

$x_0 + x_1 + 9x_2 = 20$ (3)

By (1) and (3): $9x_2 \geq 10$

Compare to (2): $9x_2 = 10$

Thus $x_0 = 0$ (and $x_1 = 10$, $x_2 = 10/9$)

Max is achieved only if $x_0 = 0$ —> rather *unfair*

➡ We need to qualify the *tradeoff between efficiency and fairness*

Pareto Efficiency

- A feasible rate allocation $\vec{x}$ is called *Pareto-efficient* (or *Pareto-optimal*), iff increasing the rate of a flow *implies* decreasing the rate of another

$\vec{x}$ is Pareto-efficient iff : for any other feasible $\vec{x}'$, $\exists i: x'_i > x_i \Rightarrow \exists j: x'_j < x_j$

$\Leftrightarrow$ every flow i has a *bottleneck* = *saturated* link,

i.e. its constraint is satisfied with *equality*

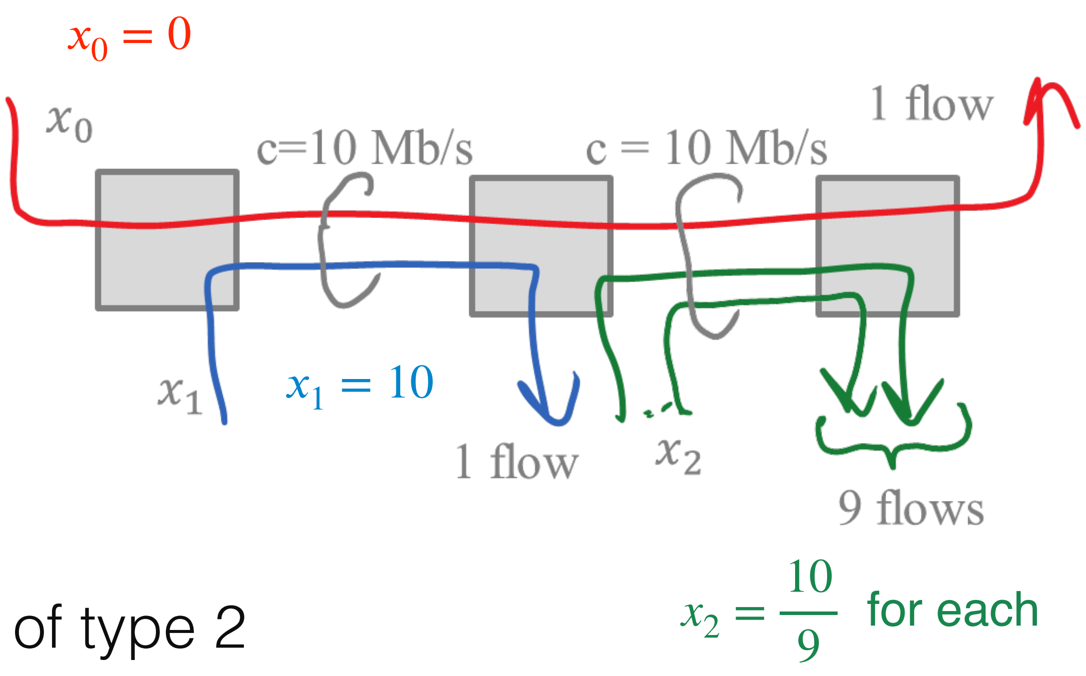
- *Conversely:* An allocation $\vec{x}$ is not Pareto-efficient iff a rate can be improved unilaterally,
  i.e. there exists a feasible allocation $\vec{x}'$, such that $x'_i > x_i$ for some i and $x'_j \geq x_j$ for all j

Example

Is the allocation

$$\{x_0 = 0, x_1 = 10, x_2 = \frac{10}{9}\}$$

Pareto-efficient?



- Link 1 is bottleneck for flows 0 and 1
- Link 2 is bottleneck for flow 0 and all flows of type 2

➡ Every flow has a bottleneck and cannot be increased unilaterally: **The allocation is Pareto-efficient.**

- *Note*: the throughput-maximizing allocation is *always* Pareto-efficient.

Which allocations are Pareto-Efficient ?

A. $x_0 = 1, x_1 = 0.5, \ x_2 = 8, \ x_3 = 1$

B. $x_0 = 1, \ x_1 = 1, \ x_2 = 8, \ x_3 = 1$

C. $x_0 = 1, \ x_1 = 1, \ x_2 = 2, \ x_3 = 7$

D. A and B

E. A and C

F. B and C

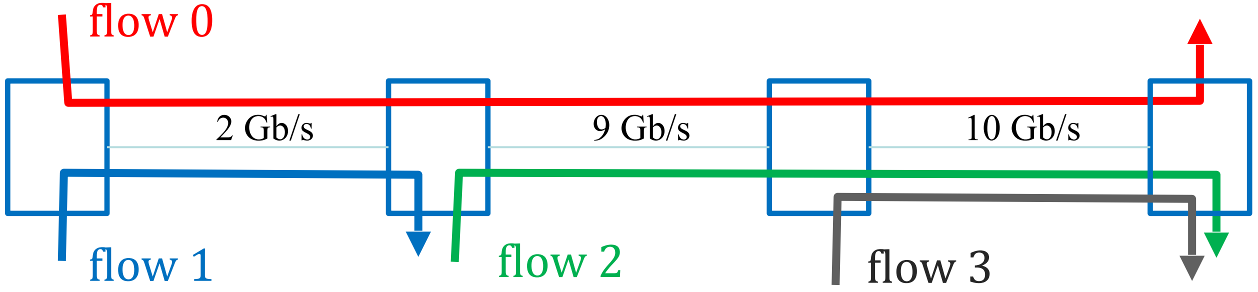
G. All

H. None

I. I don't know

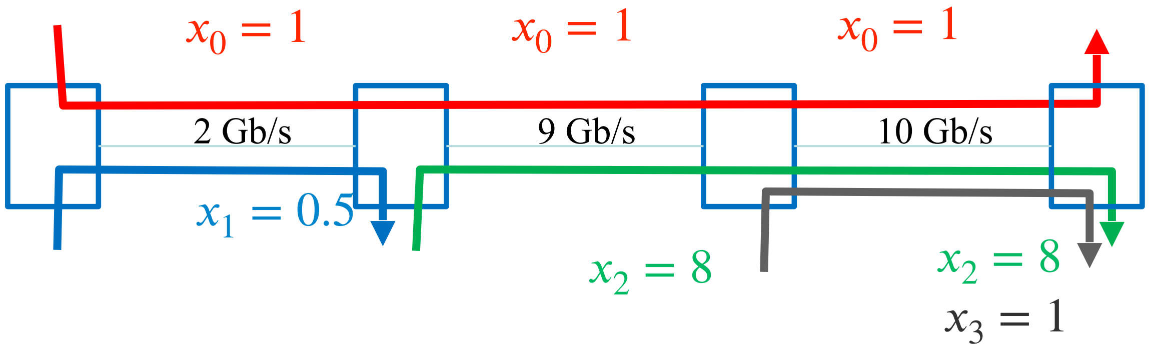
Go to web.speakup.info or download speakup app

Join room 87072



Solution

Answer F (B and C)



Allocation A: $x_0 = 1, x_1 = 0.5,\ x_2 = 8,\ x_3 = 1$

Flow 1 does not have a bottleneck.

Its rate can be increased unilaterally.

For example, we can increase x_1 to $x'_1 = 0.6$ while leaving the other rates unchanged and still obtain a feasible allocation.

Allocation A is not Pareto-efficient

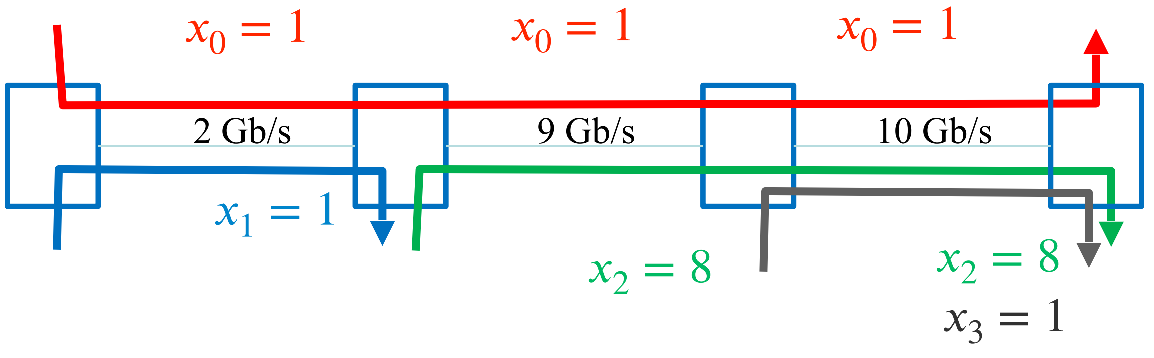
Solution

Allocation B: $x_0 = 1,\ x_1 = 1,\ x_2 = 8,\ x_3 = 1$

Link 1 is bottleneck for flows 0 and 1

Link 2 is bottleneck for flows 0 and 2

Link 3 is bottleneck for flows 0, 2 and 3



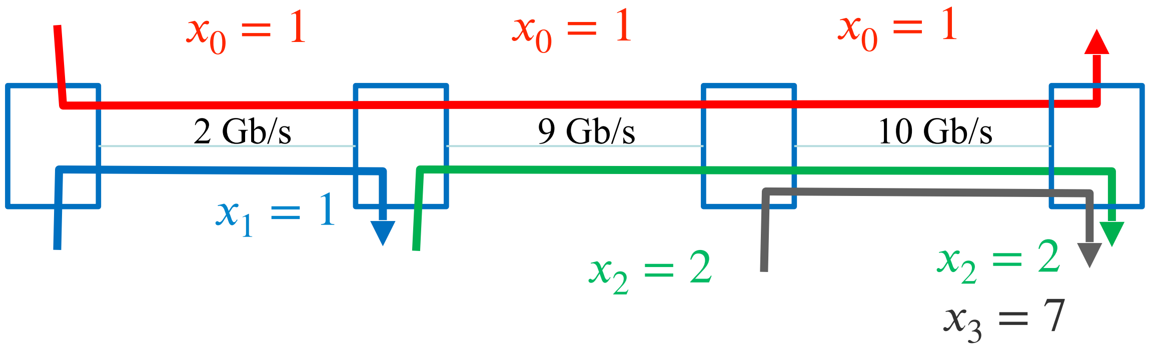
Every flow has a bottleneck. None can be increased unilaterally.
Allocation B is Pareto-efficient.

Solution

Allocation C: $x_0 = 1,\ x_1 = 1,\ x_2 = 2,\ x_3 = 7$

Link 1 is bottleneck for sources 0 and 1

Link 3 is bottleneck for sources 0, 2 and 3

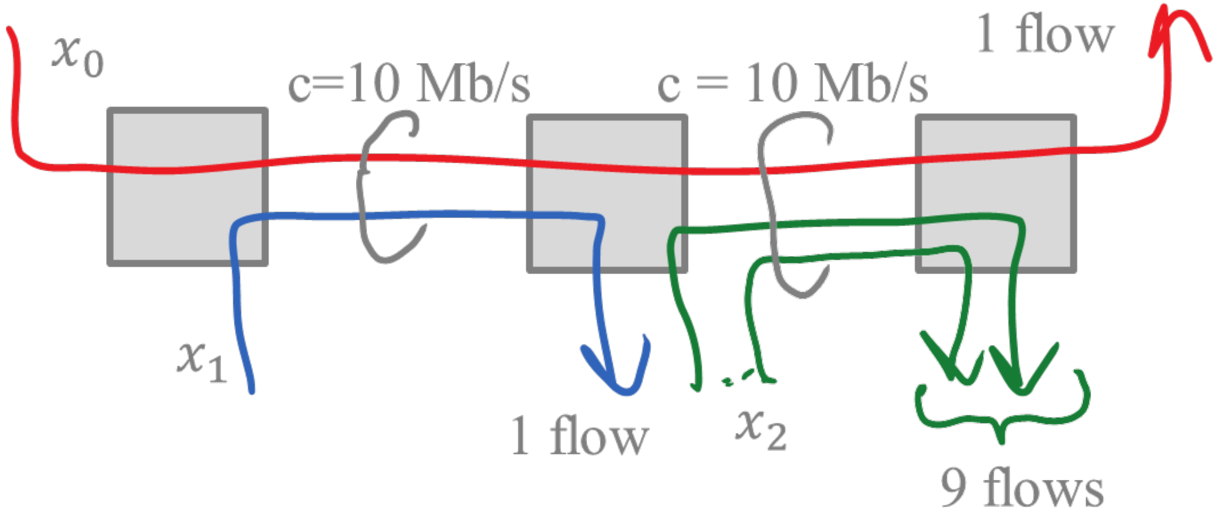


Every flow has a bottleneck. None can be increased unilaterally. Allocation C is Pareto-efficient.

Observation: link 2 is ***not*** saturated in this Pareto-efficient allocation.

Recap

- Maximizing total throughput is Pareto efficient, but may be *unfair*
  - e.g. in figure: max efficiency means Pareto efficiency,
    but also means “shutting down flow 0”



So:

➡ Are there Pareto efficient allocations that are *fair* ?

➡ What is a good *definition* of fairness?

3. Definition 1: Egalitarianism (or Neutrality):
"Allocate as much as possible but same to all."

Go to web.speakup.info
or
download speakup app
Join room
46045

In this example, what is a fair/egalitarian allocation?

A. $x_0 = x_1 = x_2 = 0.5 \; Mb/s$

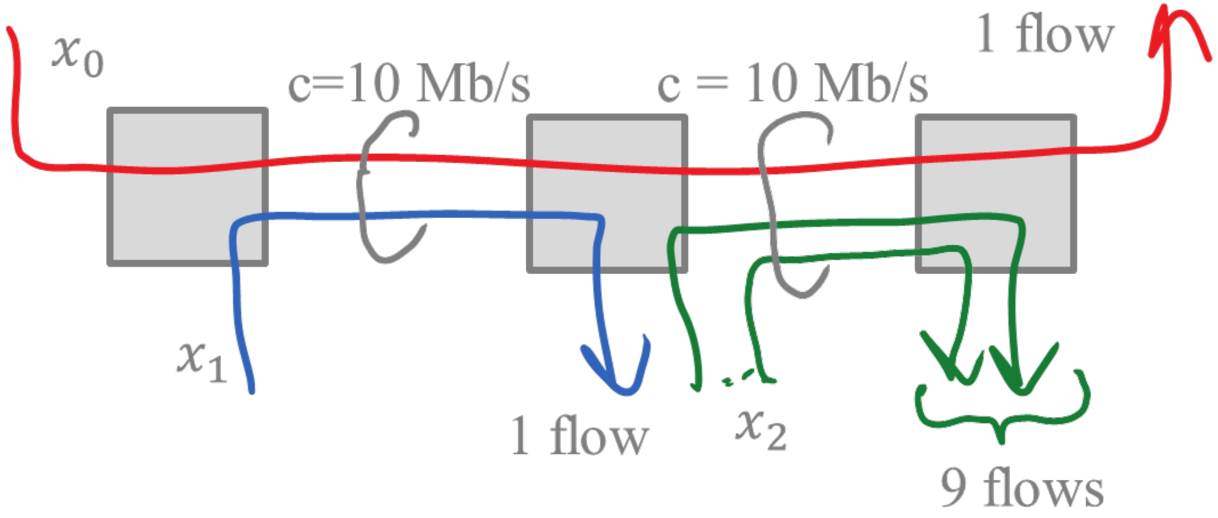
B. $x_0 = x_1 = x_2 = 1 \; Mb/s$

C. $x_0 = x_1 = x_2 = \frac{10}{9} \; Mb/s$

D. All of them

E. None of the above

F. I don't know



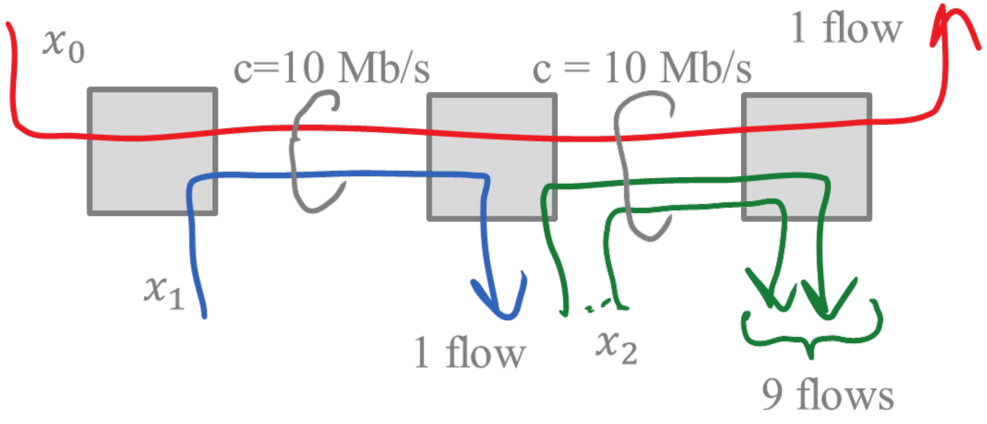
Solution

Maximize $x = x_0 = x_1 = x_2$ subject to

$2\,x \leq 10$

$10\,x \leq 10$

with $x \geq 0$

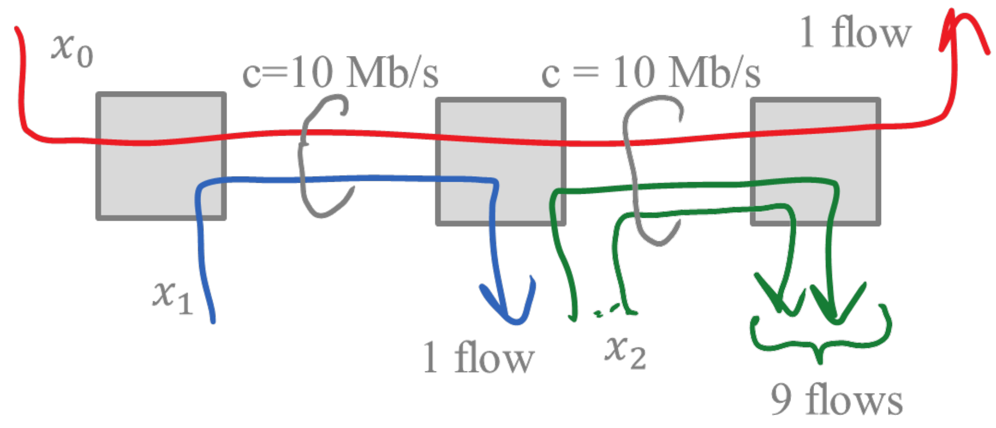


The solution is $x = 1$ Mb/s

Answer B

Egalitarianism is *not always* Pareto-efficient

- Egalitarianism gives $x = 1$ Mb/s to all but, we could give more to x_1 without hurting anyone
  - ➡ *So, allocation is not Pareto-efficient*



- A *better* allocation would be:

  $\{x_0 = 1,\ x_1 = 9,\ \ x_2 = 1\}$, which is
  - Pareto-efficient (= every resource has a bottleneck)
  - but also "fair" (= it gives to every one at least as much as egalitarianism)
  - in fact, this is a *max-min fair* allocation [see next slide]

Max-Min fairness

A feasible allocation $\vec{x}$ is max-min fair iff for any other feasible allocation

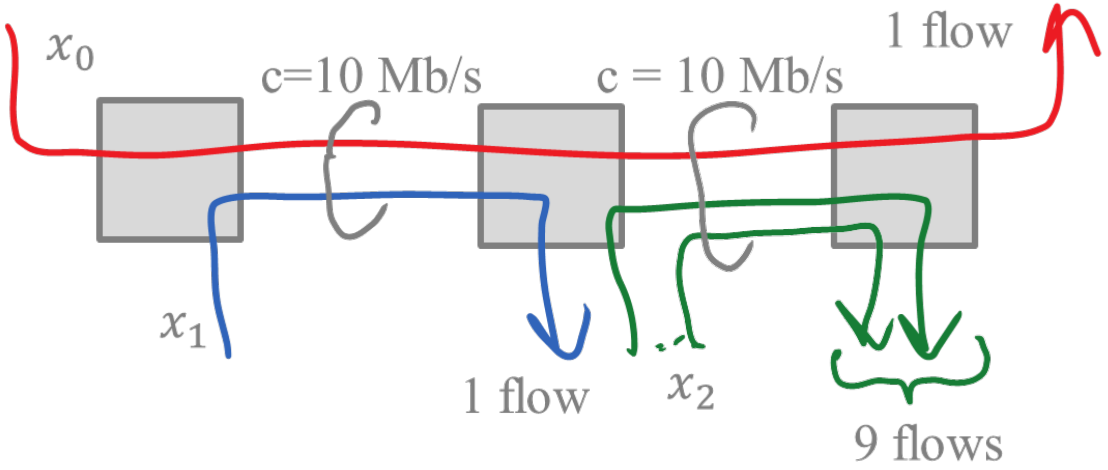
$$\vec{x}', \; (\exists i:\; x'_i > x_i \Rightarrow \exists j:\; x'_j < x_j \text{ and } x_j \leq x_i)$$

I.e. for every flow i, increasing its rate must force the rate of some other, *not richer* flow j to decrease

Note: the max-min fairness implies *Pareto-efficiency* (converse is not true)

Which Pareto-efficient allocations are also max-min fair?

A	B
$x_0 = 0$ Mb/s	$x_0 = 1$ Mb/s
$x_1 = 10$ Mb/s	$x_1 = 9$ Mb/s
$x_2 = \frac{10}{9}$ Mb/s	$x_2 = 1$ Mb/s



A. A
B. B
C. A and B
D. None
E. I don't know

Go to web.speakup.info or download speakup app

Join room 87072

Solution

A

$x_0 = 0$ Mb/s

$x_1 = 10$ Mb/s

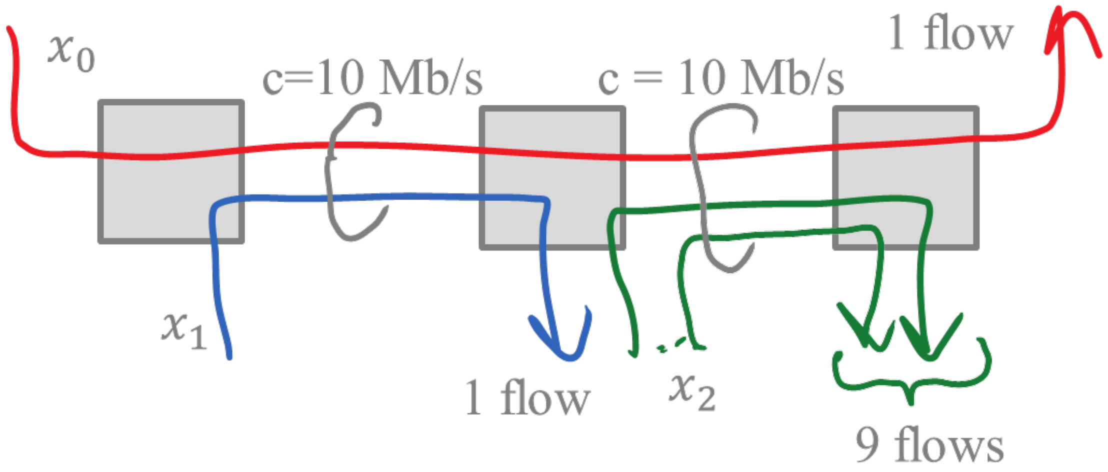
$x_2 = \frac{10}{9}$ Mb/s

B

$x_0 = 1$ Mb/s

$x_1 = 9$ Mb/s

$x_2 = 1$ Mb/s



Allocation A

- Increase x_0 (e.g. $x_0 \leftarrow 1$) and decrease x_1 ($x_1 \leftarrow 9$) and x_2 ($x_2 \leftarrow 1$); this does not contradict fairness because x_1 and x_2 are larger than x_0
- x_1 and x_2 cannot be increased because the capacity constraints will be violated
- So, only one flow rate can increase but this does not contradict fairness.

A is not max-min fair

Allocation B

- If I increase x_0 I must decrease $x_2 \Rightarrow$ contradicts fairness
- If I increase x_1 I must decrease $x_0 \Rightarrow$ contradicts fairness
- If I increase x_2 I must decrease $x_0 \Rightarrow$ contradicts fairness

Any increase contradicts fairness
B is max-min fair

Max-Min fairness: properties and computation

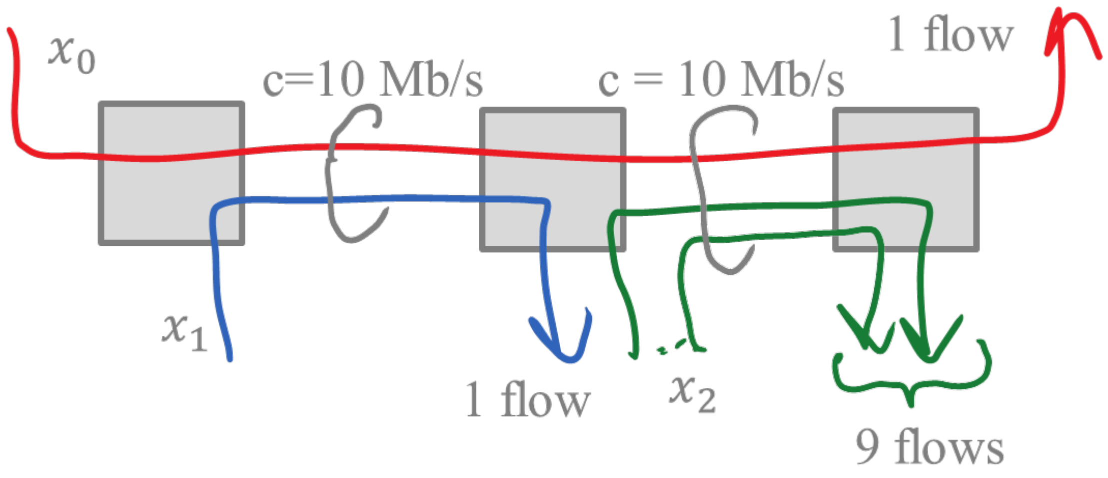
Given a set of constraints, i.e. a set of feasible allocations:

a. if it exists, the max-min fair allocation is *unique*
b. there *does exist* a max-min fair allocation, if the set of feasible allocations is *convex* (which is the case in networks, as we have linear constraints)
c. the max-min fair allocation is *Pareto-efficient* (converse is not true)

For a convex set of feasible rates (as in our case), the *unique* max min fair allocation is obtained by the *water-filling algorithm*:

```
1. mark all flows as unfrozen
2. do
3.   set rates of unfrozen flows to a common max attainable value (egalitarian)
4.   mark flows using a saturated link as frozen
5. until
      all flows are frozen
```

Water-Filling: Example



Step 1:

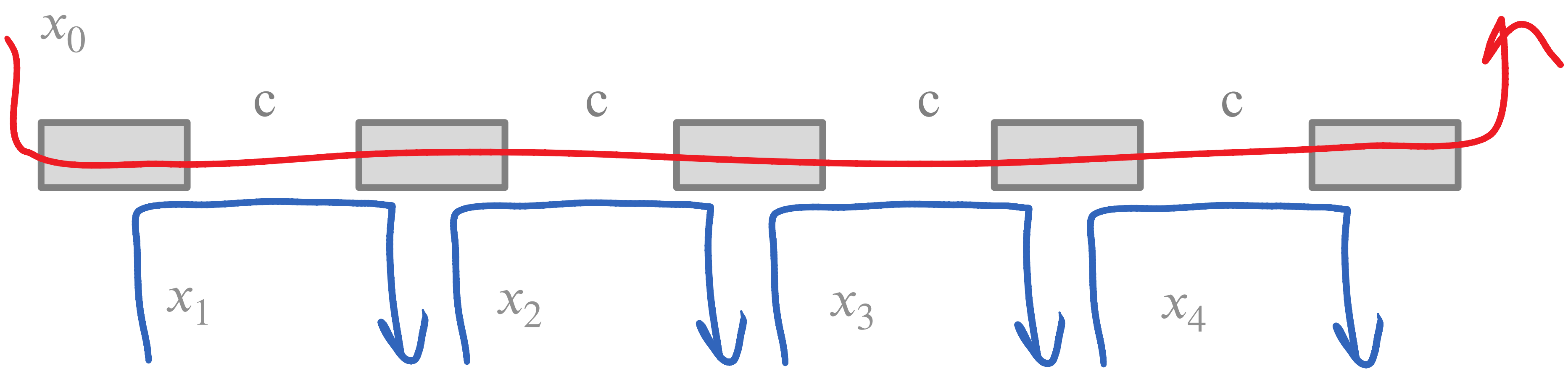
- maximize t such that $x_0 = x_1 = x_2 = t$ and all constraints are satisfied; we find $t = 1$, hence $x_0 = x_1 = x_2 = 1$;
- link 2 is saturated, is used by flows 0 and 2 ⇒ mark flows 0 and 2 as *frozen*

Step 2 :

- maximize t such that $x_1 = t$, with $x_0 = 1, x_2 = 1$ and all constraints are satisfied; we find $t = 9$, hence $x_0 = x_2 = 1$ and $x_1 = 9$
- link 1 is saturated, is used by sources 0 and 1 ⇒ mark flow 1 as frozen; all flows are frozen, STOP.

The max-min fair allocation is $x_0 = x_2 = 1$ and $x_1 = 9$

What is the max-min fair allocation ?



A. $x_i = \frac{c}{2} \quad \forall i$

B. $x_0 = \frac{c}{3}, \quad x_i = \frac{2c}{3} \quad \forall i \neq 0$

C. $x_0 = \frac{c}{4}, \quad x_i = \frac{3c}{4} \quad \forall i \neq 0$

D. $x_0 = \frac{c}{5}, \quad x_i = \frac{4c}{5} \quad \forall i \neq 0$

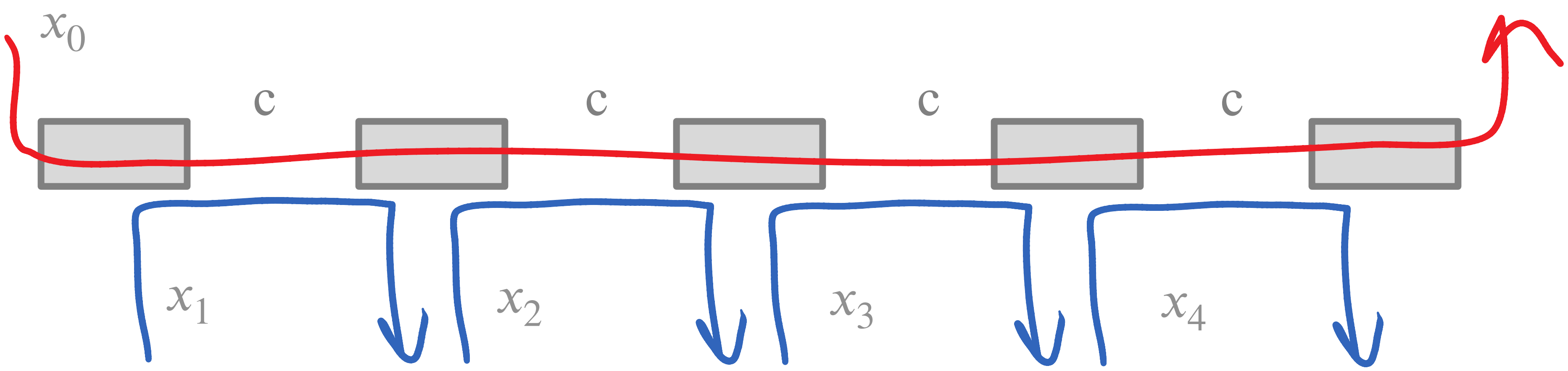
E. None of the above

F. I don't know

Go to web.speakup.info or download speakup app

Join room 87072

Solution



Answer A

The max-min fair allocation via water-filling gives the same rate $\frac{c}{2}$ to all flows

But this seems *"not fair enough"* in terms of resource usage!

- One could claim that x_0 should get 4x less, because it uses 4x more links (answer D).

This is what led to the definition of *proportional fairness*…

Definition of Proportional Fairness

A feasible allocation $\vec{x}$ is proportionally fair iff $\vec{x} > 0$ and for any other feasible $\vec{x}'$, it holds:

$$\sum_i \frac{x_i' - x_i}{x_i} \leq 0$$

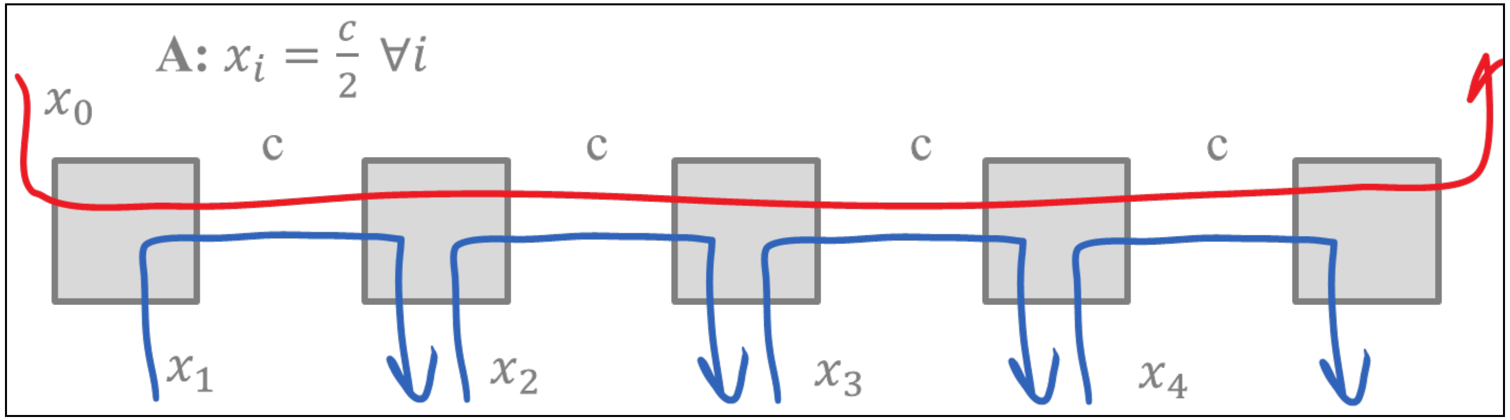
I.e.: An allocation is proportionally fair, if for any other allocation, the *aggregate rate of change* or *relative change* $\sum_i \frac{dx_i}{x_i}$ is *non-positive*, i.e. *"the other allocations become relatively worse overall"*

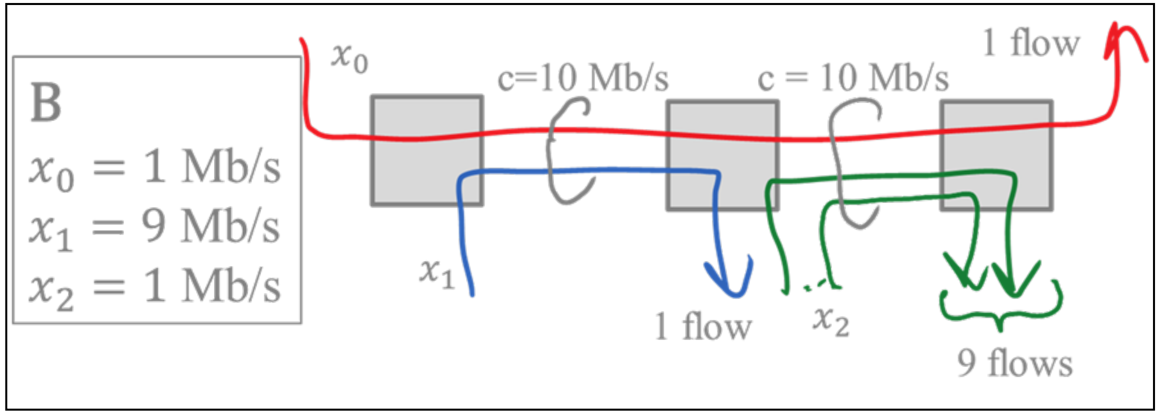
Conversely: An allocation $\vec{x} > 0$ is *not* proportionally fair iff there exists $\vec{x}'$ s.t.: $\sum_i \frac{x_i' - x_i}{x_i} > 0.$

Two important points:

- Sum of *all* rates of changes matters, not only one
- *Relative* changes matter, not absolute

Which allocations seem proportionally fair ?



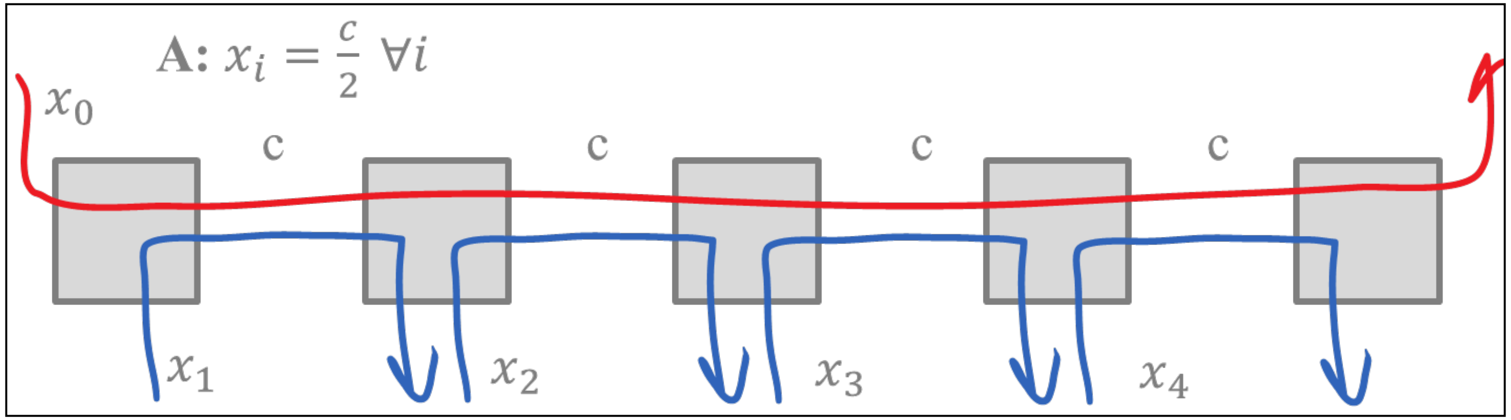


A. A
B. B
C. A and B
D. None
E. I don't know

Go to web.speakup.info or download speakup app

Join room 87072

Solution



Answer D

Let $x_i \leftarrow x_i + \delta$ for $i = 1\ldots4$ and $x_0 \leftarrow x_0 - \delta$.

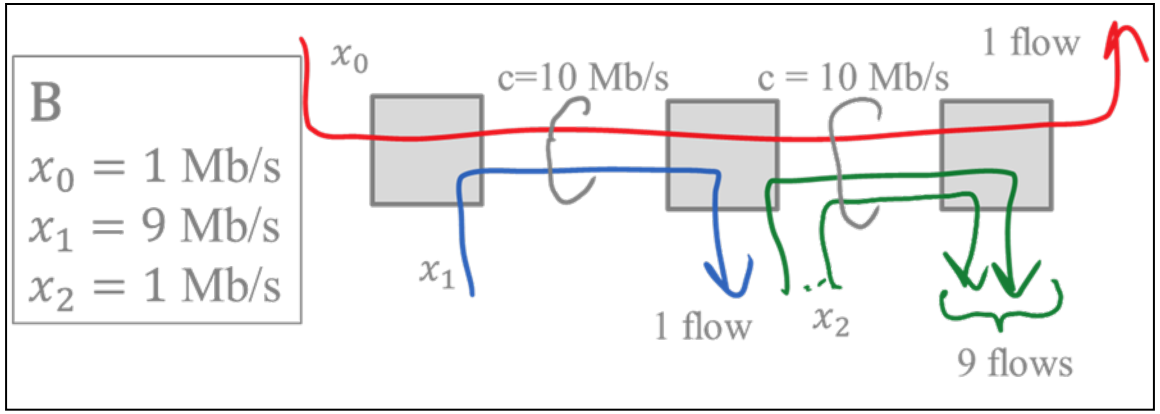
If δ is small enough (i.e. $0 < \delta \leq \frac{c}{2}$), the new allocation is feasible (= within constraints).

The total rate of change is $-\frac{\delta}{\frac{c}{2}} + 4\,\frac{\delta}{\frac{c}{2}} > 0$.

So, we could change the allocation and obtain a positive total rate of change.

A is not proportionally fair

Solution



Let $x_2 \leftarrow x_2 + \delta$ and $x_0 \leftarrow x_0 - 9\delta$; $x_1 \leftarrow x_1 + 9\delta$.

If δ is small enough (i.e. $0 < \delta \leq \frac{1}{9}$), the new allocation is feasible.

The total rate of change is $-\frac{9\delta}{1} + \frac{9\delta}{9} + 9\,\frac{\delta}{1} = \delta > 0$.

So, we could change the allocation and obtain a positive total rate of change.

B is not proportionally fair

So, min-max fairness does not imply proportional fairness

Proportional Fairness: properties and computation

a. A proportionally fair allocation is *Pareto-efficient*

b. Given a convex set of constraints for the rates (as in our case), the proportionally fair allocation *exists* and is *unique*

c. It is obtained by maximizing

$$J(\vec{x}) := \sum_i \log x_i$$

over all feasible allocations

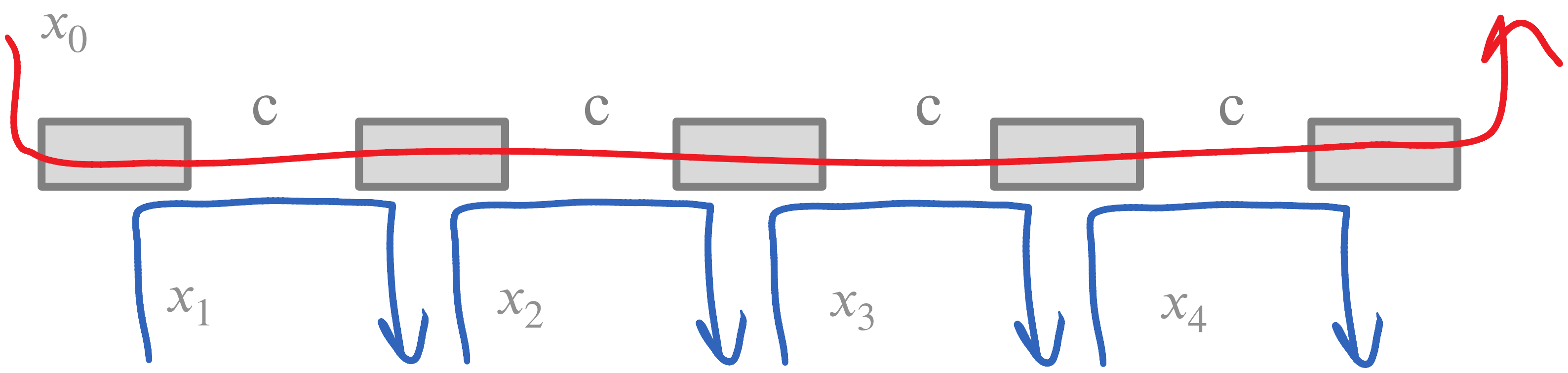
Intuitive explanation via gradient: $dJ(\vec{x}) = \sum_i \frac{dx_i}{x_i}$.

Deviating from the maximum implies going towards a non-positive gradient

=> the total rate of change is non-positive

=> the maximizing allocation is proportionally fair!

Let us compute the proportionally fair allocation



We have to solve the optimization problem:

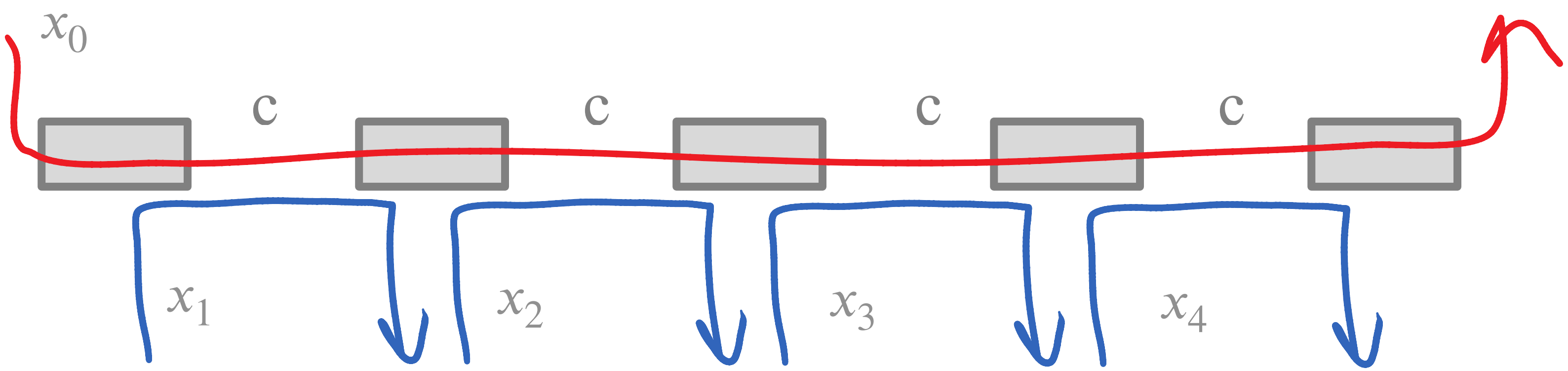
$$\max U = \log x_0 + \log x_1 + \log x_2 + \log x_3 + \log x_4$$

subject to

$$x_0 + x_1 \leq c$$
$$x_0 + x_2 \leq c$$
$$x_0 + x_3 \leq c$$
$$x_0 + x_4 \leq c$$

We can use convex optimization techniques to solve this, but here we can also do a direct solution…

Let us compute the proportionally fair allocation



We have to solve the optimization problem:

$$\max U = \log x_0 + \log x_1 + \log x_2 + \log x_3 + \log x_4$$

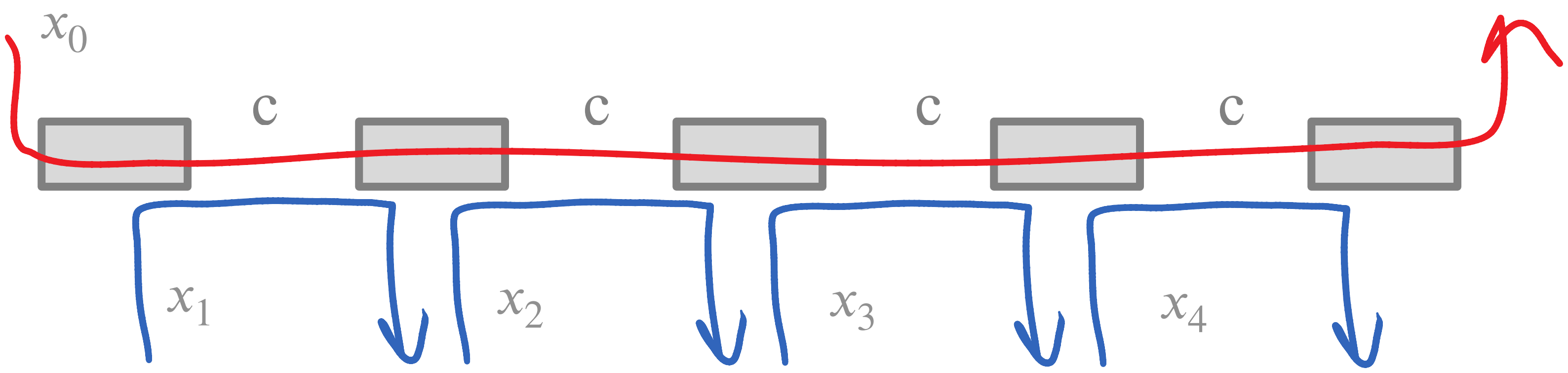
subject to

$$x_0 + x_1 \leq c$$
$$x_0 + x_2 \leq c$$
$$x_0 + x_3 \leq c$$
$$x_0 + x_4 \leq c$$

Observe: at the maximum point, we must have equality in all constraints otherwise we can increase x_i $(i \neq 0)$ and increase *U* (i.e. find a better maximum).

Therefore, for any choice of $x_0 = x^*$, we must have $x_1 = x_2 = x_3 = x_4 = c - x^*$.

Let us compute the proportionally fair allocation



So, we rewrite the optimization problem as:

$$\max U = \log x^* + 4\log\left(c - x^*\right)$$

$$\text{subject to } \; 0 < x^* < c$$

This is a 1d problem, can be solved by computing the derivative

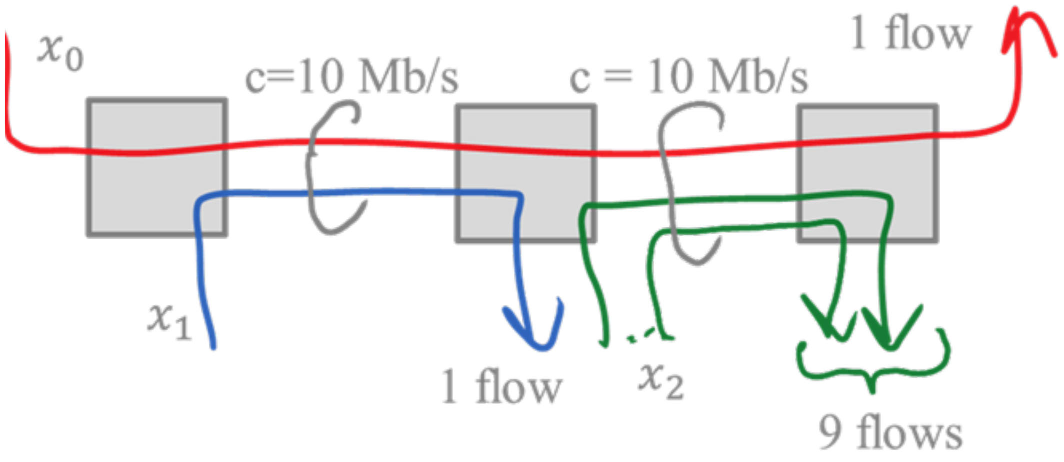
We find $\dfrac{dU}{dx^*} = \dfrac{1}{x^*} - \dfrac{4}{c - x^*}$

There is a maximum for $x^* = \dfrac{c - x^*}{4}$ i.e. $x^* = \dfrac{c}{5}$

The proportionally fair allocation is

$$x_0 = \frac{c}{5}, \qquad x_1 = x_2 = x_3 = x_4 = \frac{4c}{5}$$

Which one is *the* proportionally fair allocation? (in Mb/s) (only one answer)



A. $x_0 = 1, x_1 = 9, x_2 = 1$

B. $x_0 = 0.909,\ x_1 = 9, x_2 = 1.010$

C. $x_0 = 1.009,\ x_1 = 8.991,\ x_2 = 0.999$

D. $x_0 = 0.909,\ x_1 = 9.091,\ x_2 = 1.010$

E. I don’t know

Go to web.speakup.info or download speakup app

Join room 87072

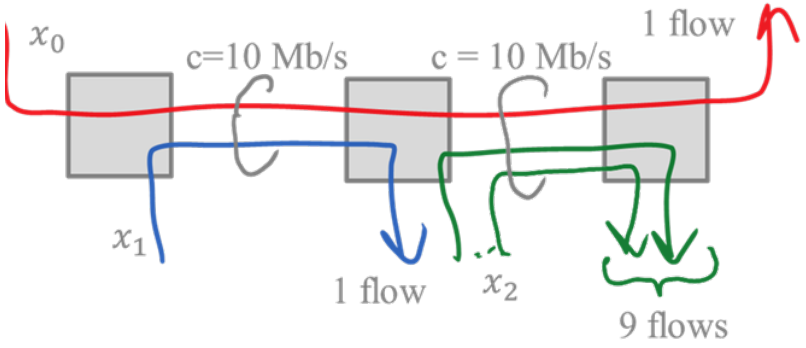
Solution

A. $x_0 = 1, x_1 = 9, x_2 = 1$

B. $x_0 = 0.909, x_1 = 9, x_2 = 1.010$

C. $x_0 = 1.009, x_1 = 8.991, x_2 = 0.999$

D. $x_0 = 0.909, x_1 = 9.091, x_2 = 1.010$

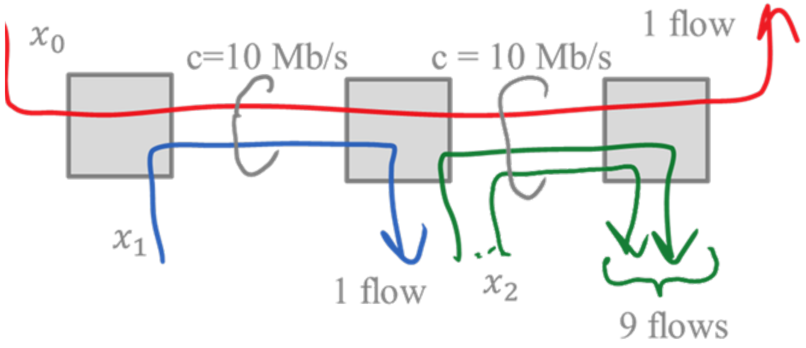


Answer D. Why?

- We saw earlier that A is not proportionally fair
- B is not Pareto-efficient (you can increase x_1 only)—therefore is also not proportionally fair
- C goes in the wrong direction (gives more to 0 than to 2) and is probably not proportionally fair
- D is probably the correct answer

Solution

A. $x_0 = 1, x_1 = 9, x_2 = 1$
B. $x_0 = 0.909, x_1 = 9, x_2 = 1.010$
C. $x_0 = 1.009, x_1 = 8.991, x_2 = 0.999$
D. $x_0 = 0.909, x_1 = 9.091, x_2 = 1.010$



We can compute the proportionally fair allocation with the *same trick* as before; and obtain

$$x_0 = x^*$$
$$x_1 = 10 - x^*$$
$$x_2 = \frac{10 - x^*}{9}$$

with x^* that maximizes $\log x + \log(10 - x) + 9\log\frac{10 - x}{9}$

$x^* = \frac{10}{11}$ Mb/s

This is allocation D

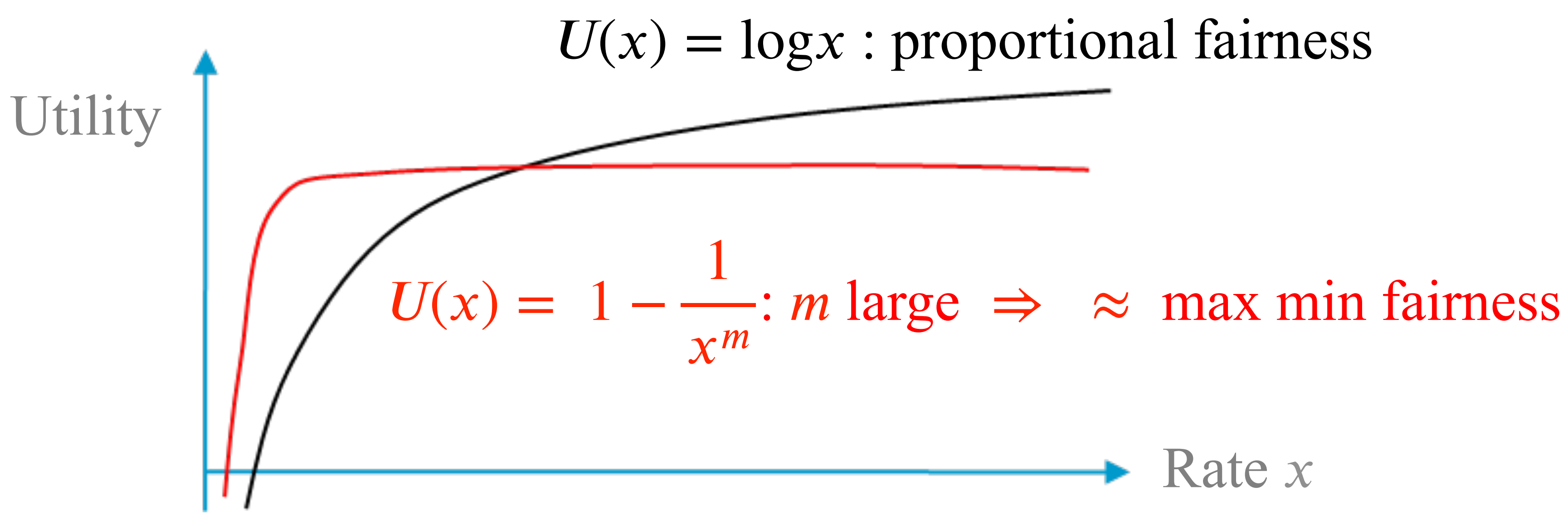
Utility Fairness

One can interpret proportional fairness as the allocation that *maximizes a global utility* $\sum_i U_i(x_i)$ with $U_i(x_i) = \log x_i$.

If we take some other utility function, we have what is called a *utility fairness.*

It can be shown that max-min fairness is the limit of utility fairness when the utility function converges to a step function.

But max-min fairness cannot be expressed exactly as a utility fairness (only at the limit).



Recap

Sources should adapt their rate to the state of the network in order to avoid inefficiencies and congestion collapse.

This is called "*congestion control*".

Such control mechanism should target a form of fairness that is Pareto-efficient

e.g. max-min fairness or proportional fairness.

4. Towards a practical implementation of congestion control

How can congestion control be implemented ?

Explicit/ Rate-based: tell every host how fast it can send

MPLS networks (smart grid)

Cellular networks

Hop by hop = backpressure: STOP/GO signals sent upstream

Gigabit LAN switches

Fair Queuing per Flow: One queue per flow / per user, served round robin

Cellular networks, industrial networks, in-vehicle networks

End-to-end: hosts "taste the water" and increase or decrease their sending rate using a host congestion control algorithm

The solution in the Internet

Additive Increase Multiplicative Decrease (AIMD)

- First congestion control algorithm deployed in the Internet and before that, in Decnet (the "**Decbit**")
- Still widely deployed today

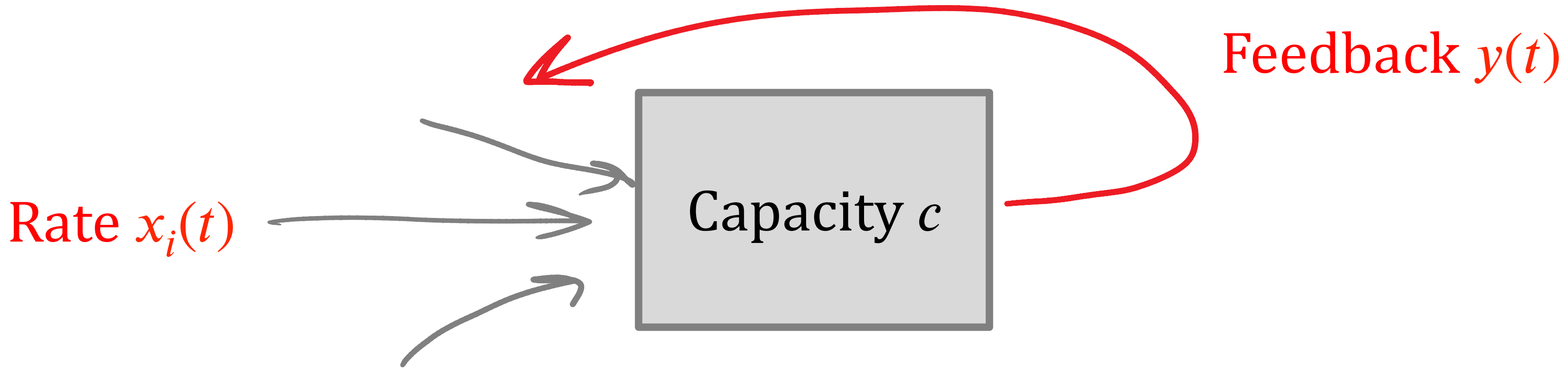
We have designed a scheme that allows a network to operate at its knee. As shown in Figure 3, the scheme uses one bit called the **congestion avoidance bit** in the network layer header of the packet for feedback from the subnet to the users. A source clears the congestion avoidance bit as the packet enters the subnet. All routers in the subnet monitor their load and if they detect that they are operating above the knee, they set the congestion avoidance bit in the packets belonging to users causing overload. Routers operating below the knee pass the bit as received. When the packet is received at the destination the network layer passes the bit to the destination transport, which takes action based on the bits.

August 1987

Raj Jain

Raj Jain, K.K. Ramakrishnan, and Dah-Ming Chiu. Congestion avoidance in computer networks with a connectionless network layer. Technical Report DEC-TR-506, Digital Equipment Corporation, August 1987.

A Simple Network Model



Network sends a one-bit feedback :

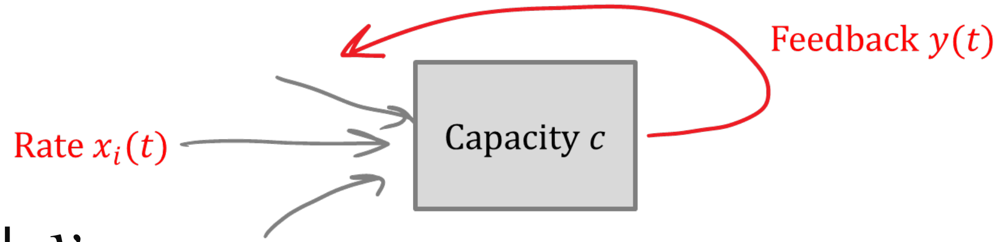
$$y(t) = 0 \text{ if } \sum_i x_i(t) \leq c$$ —> positive feedback

$$y(t) = 1 \text{ if } \sum_i x_i(t) > c$$ —> negative feedback

Sources reduce rate $x_i(t+1)$ if $y(t) = 1$, increase otherwise

➡ *Question*: what form of increase/decrease laws should one pick?

Linear Laws



We consider *linear laws*

$$\text{if } y(t) = 1 \text{ then } \quad x_i(t+1) = u_1 \cdot x_i(t) + v_1$$
$$\text{if } y(t) = 0 \text{ then } \quad x_i(t+1) = u_0 \cdot x_i(t) + v_0$$

We want to decrease when $y(t) = 1$, so

$u_1 \leq 1$ and $v_1 \leq 0$ and at least one inequality must be strict

Multiplicative decrease factor

Additive decrease term

We want to increase when $y(t) = 0$, so

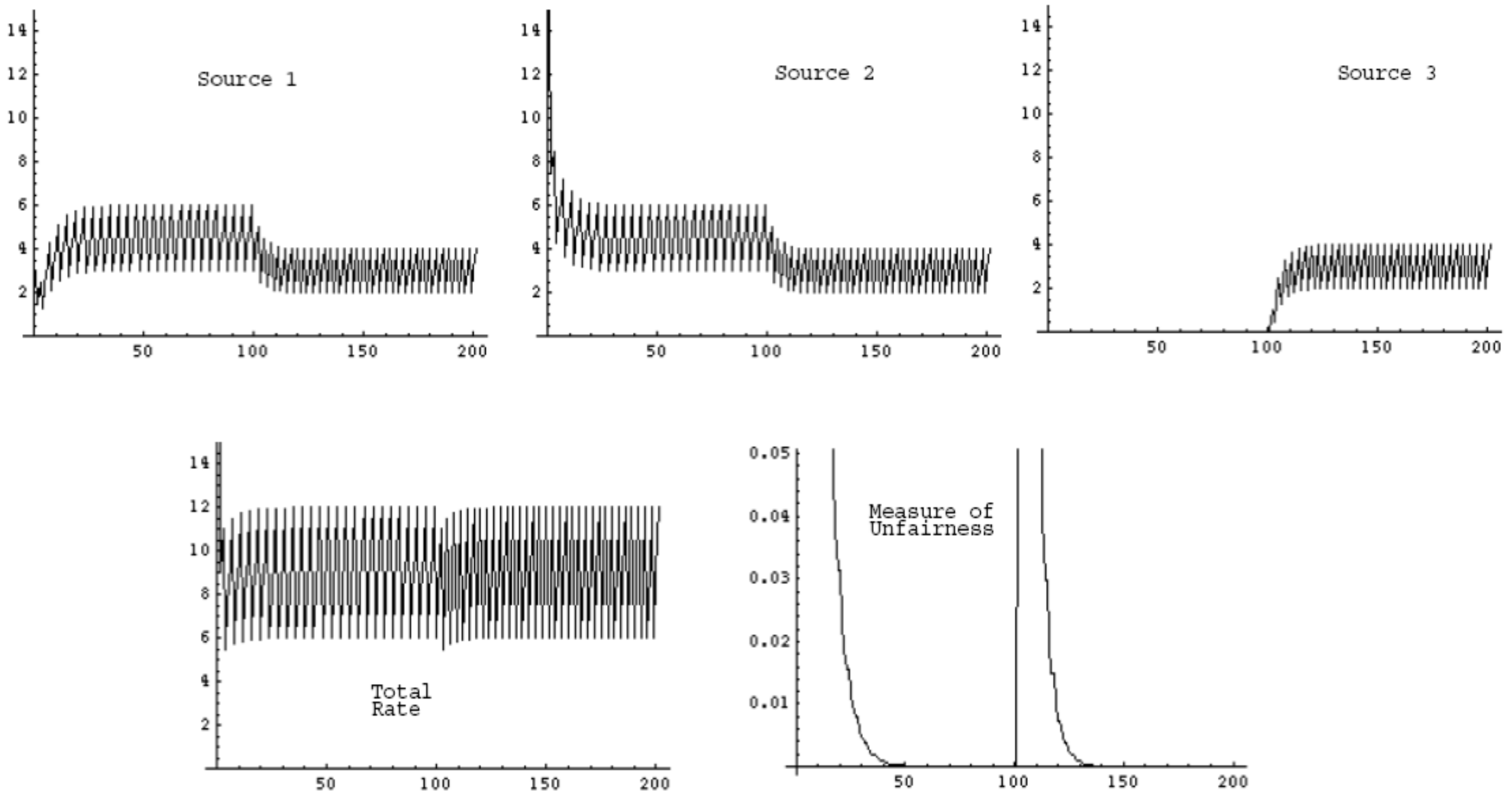
$u_0 \geq 1$ and $v_0 \geq 0$ and at least one inequality must be strict

Multiplicative increase factor

Additive increase term

Example

$u_1 = 0.5, \; v_1 = 0$ (multiplicative decrease)
$u_0 = 1, v_0 = 1$ (Mb/s) (additive increase)



So fairness seems to be achieved using this idea, but after some time and oscillations!

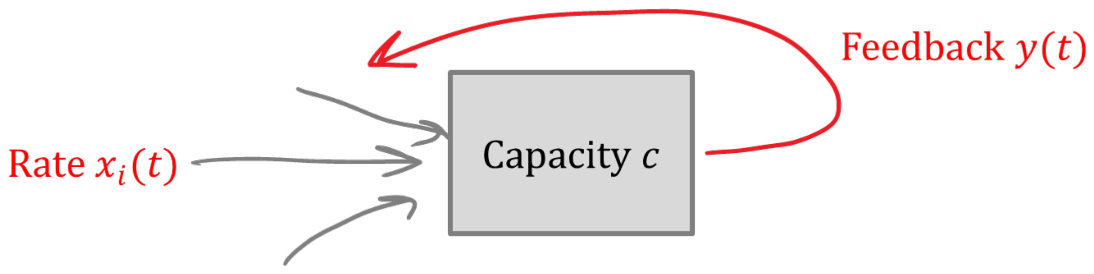
Analysis of Linear Control schemes

We want to achieve efficiency and fairness

We could target either max-min fair or proportionally fair allocations

Here (in the example with 1 link) they are the same

We will now analyze the impact of each of the four coefficients $u_0, u_1,\ v_0$ and v_1.



- We consider linear laws

  if $y(t) = 1$ then $x_i(t+1) = u_1\, x_i(t) + v_1$

  if $y(t) = 0$ then $x_i(t+1) = u_0\, xi(t) + v_0$

- We want to decrease when $y(t) = 1$, so

  $u_1 \le 1$ and $v_1 \le 0$ and at least one inequality must be strict

  Multiplicative decrease factor

  Additive decrease term

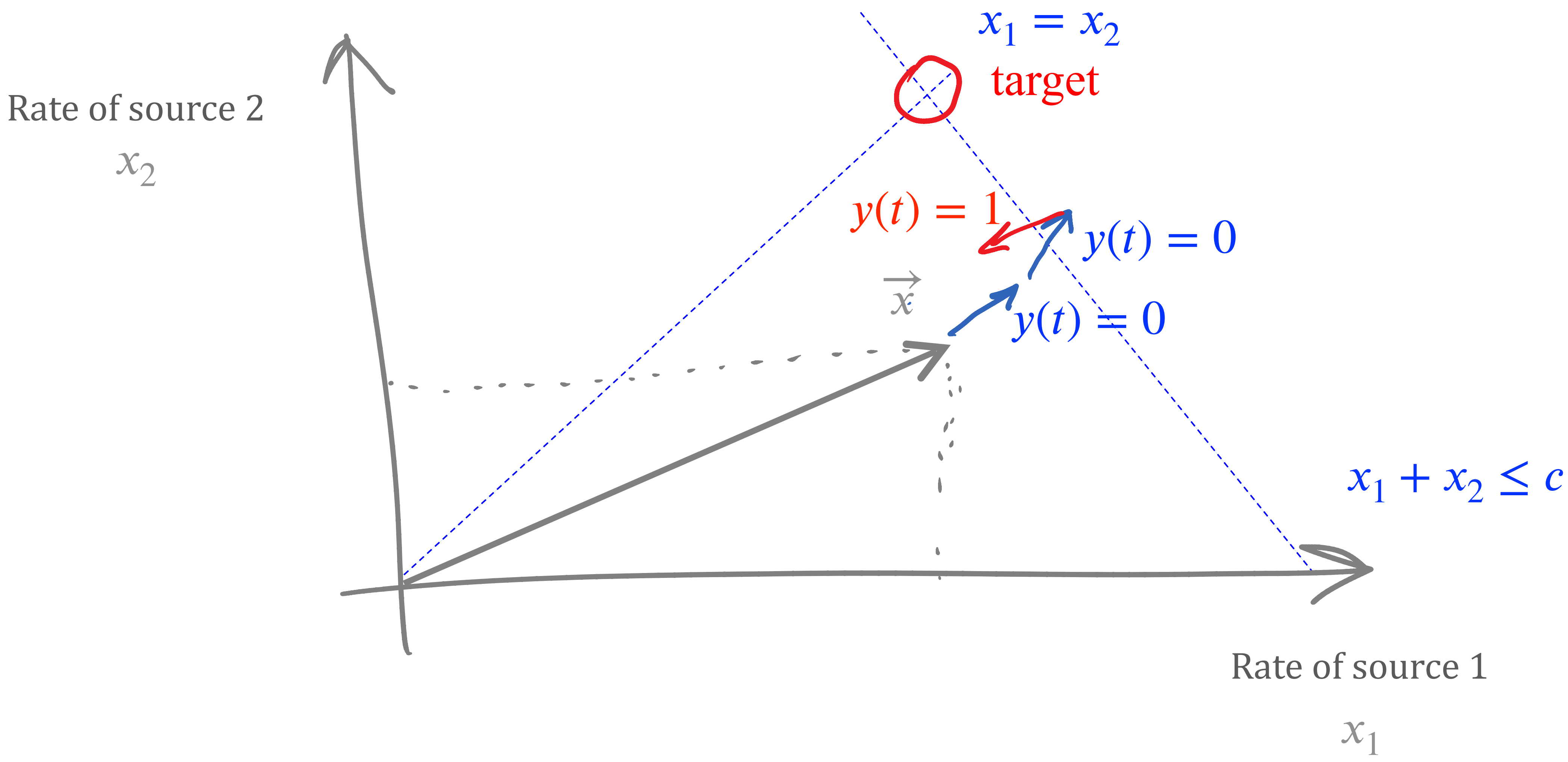
- We want to increase when $y(t) = 0$, so

  $u_0 \ge 1$ and $v_0 \ge 0$ and at least one inequality must be strict

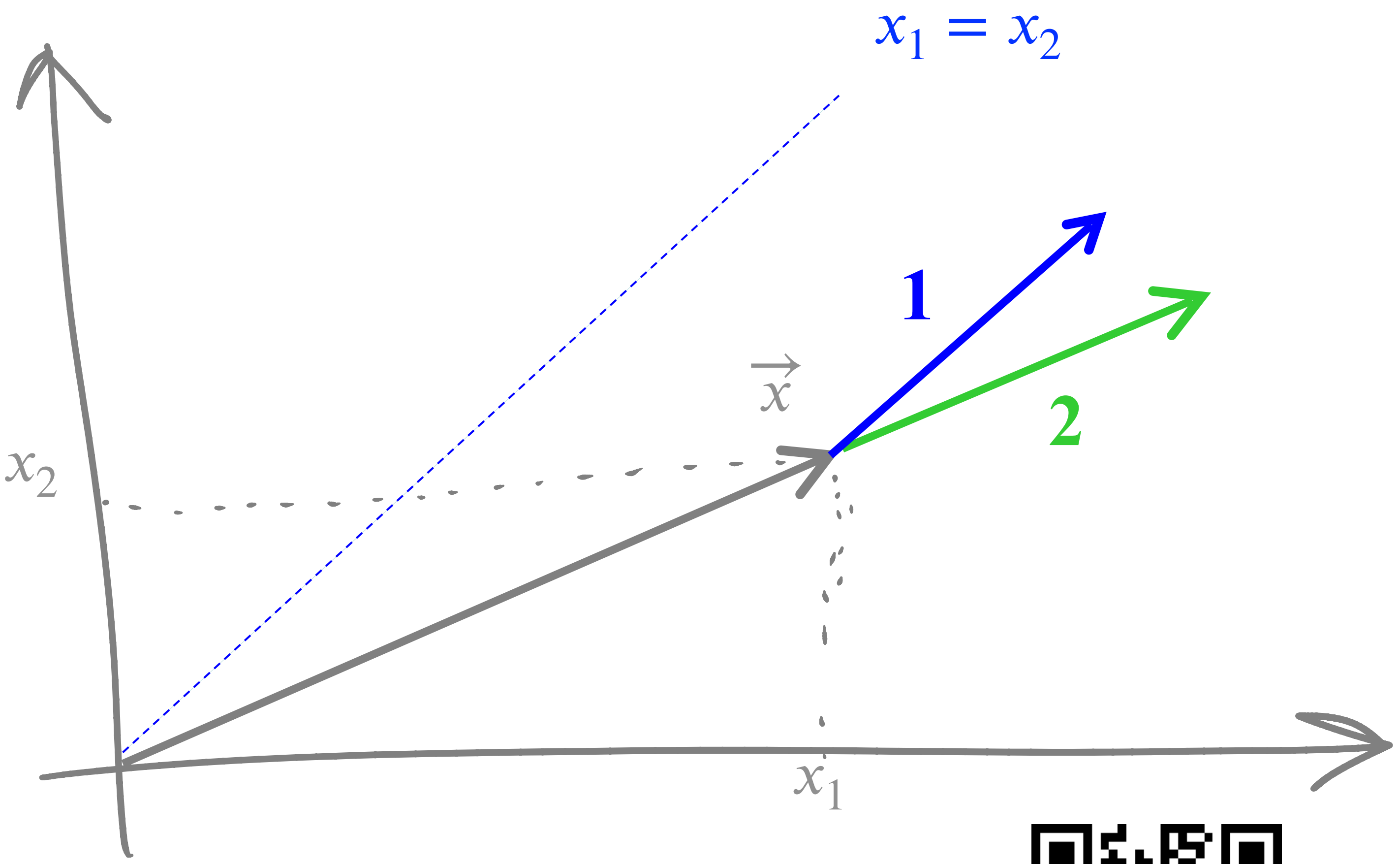
  Multiplicative increase factor

  Additive increase term

Zoom on 2 sources using a single link



Zoom on 2 sources ; say what is true

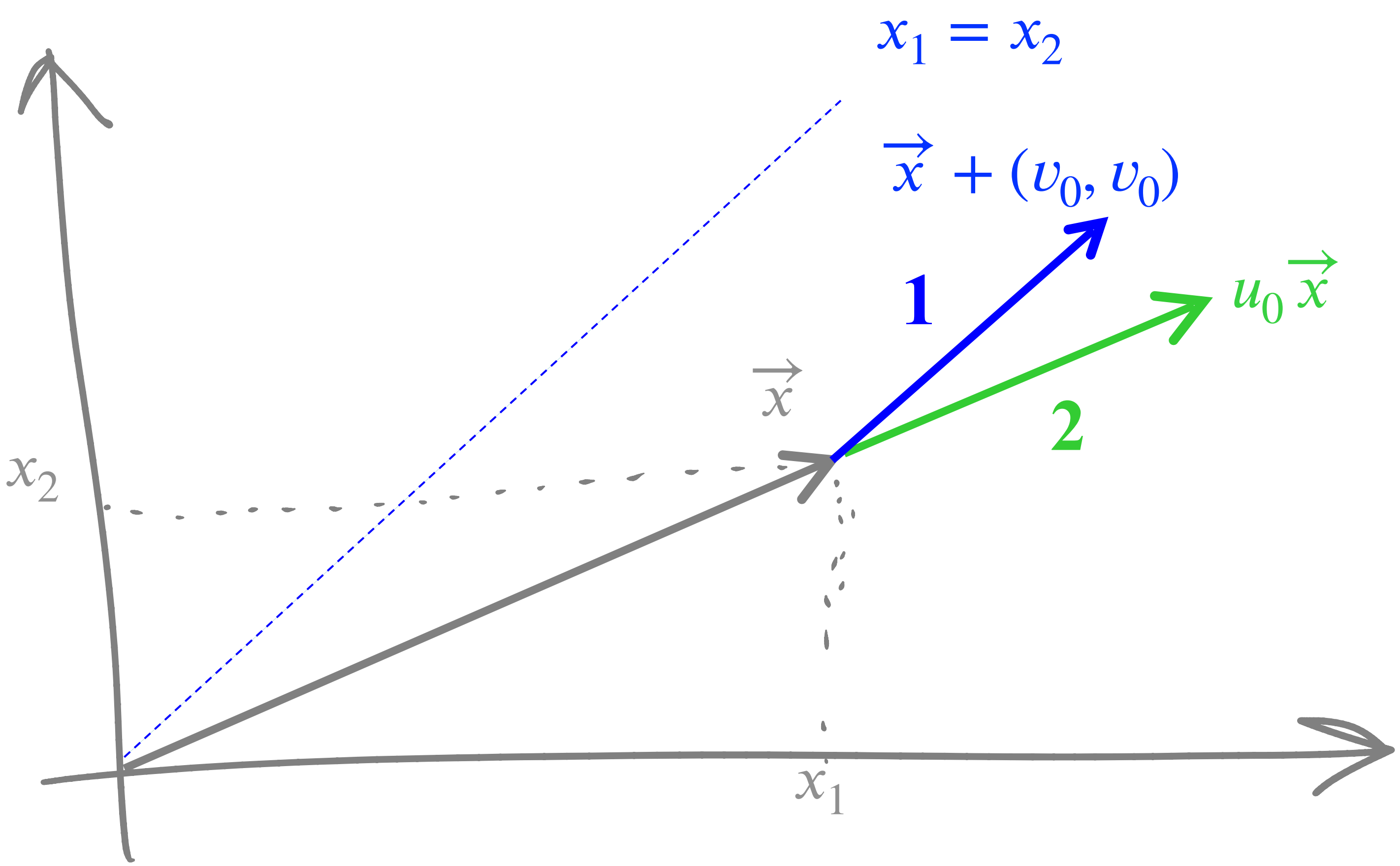


A. 1 = additive increase,
   2 = multiplicative increase
B. 1 = multiplicative increase
   2 = additive increase,
C. None of the above
D. I don’t know

Go to web.speakup.info or download speakup app

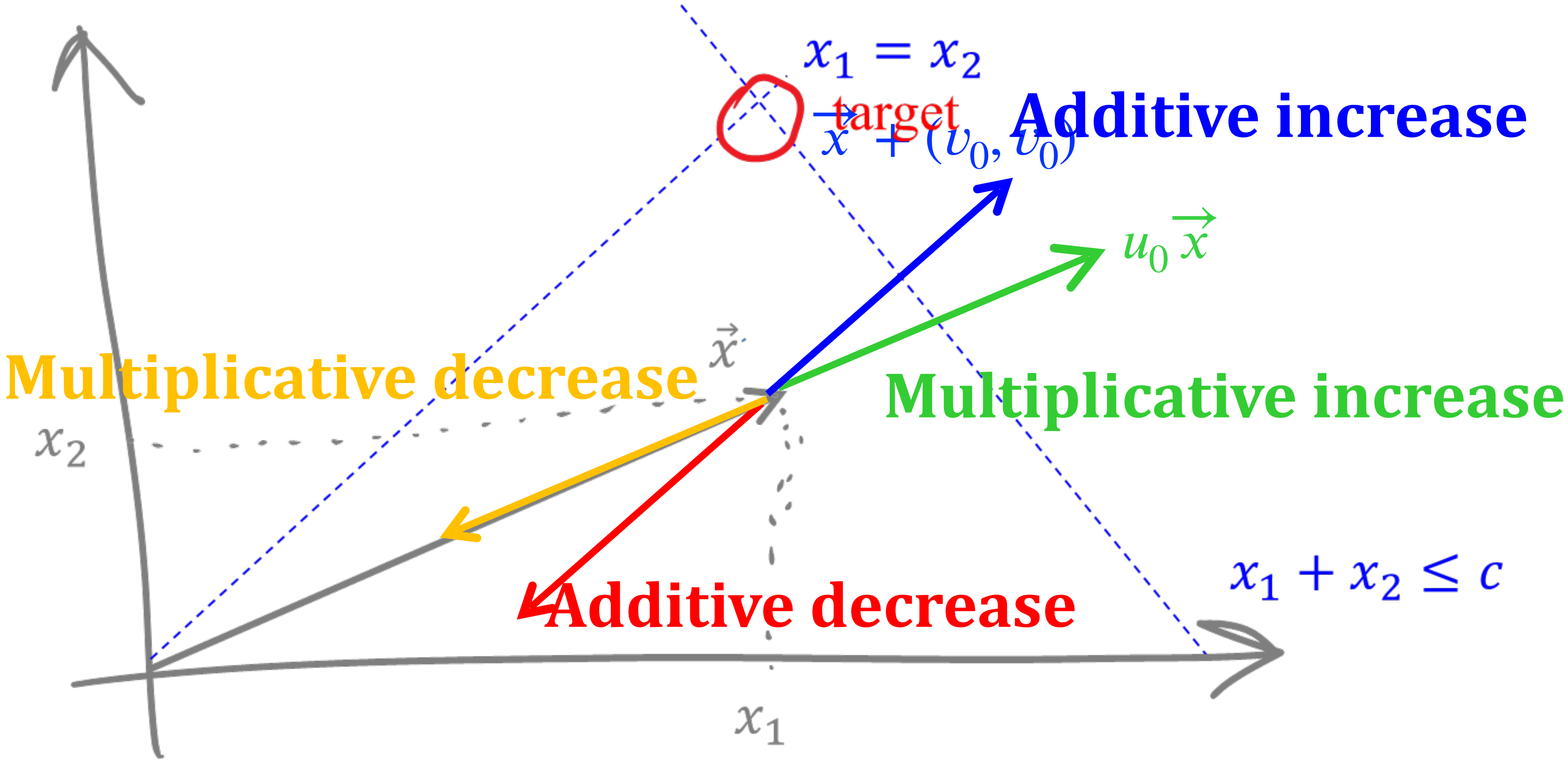
Join room 87072

Solution



1 = additive increase,
2 = multiplicative increase

Answer A



1. *Additive decrease* worsens fairness (goes away from $x_1 = x_2$) and should be avoided ⇒ decrease should be mutiplicative
2. *Additive increase* is the only move that increases fairness and should be therefore be included ⇒ increase should be additive

More generally…

Among the linear controls, *only* additive increase – multiplicative decrease (AIMD) tends to bring the allocation towards fairness and efficiency.

This is what was implemented in the Internet after the first congestion collapses.

In a more complex network setting, what type of fairness does AIMD achieve?

A. Max-min
B. Proportional
C. None of the above
D. I don't know

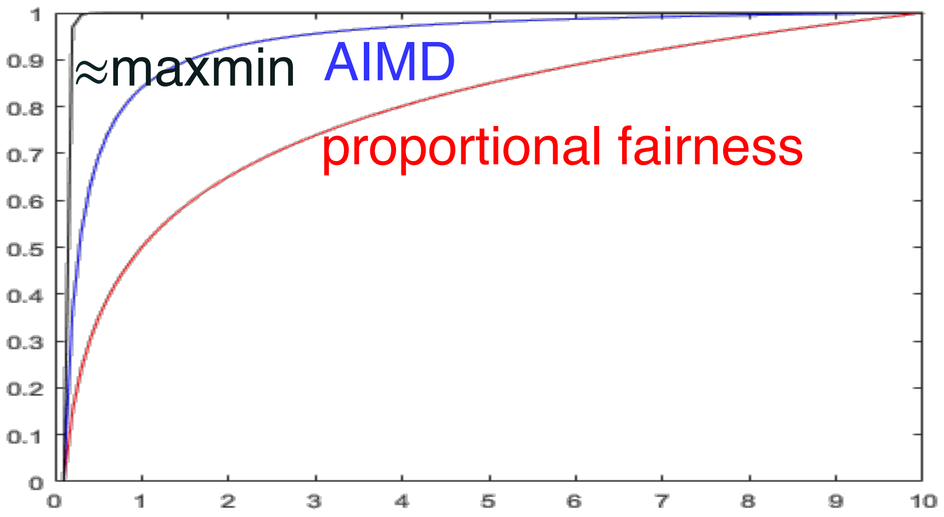
Fairness of AIMD

Answer C

AIMD with: additive increase $+r$, multiplicative decrease $\times(1-\eta)$, and one update per time unit implements utility fairness, with utility of flow i given by

$$U(x_i) = \log\frac{x_i}{r+\eta x_i}, \qquad \text{where } x_i = \text{rate.}$$

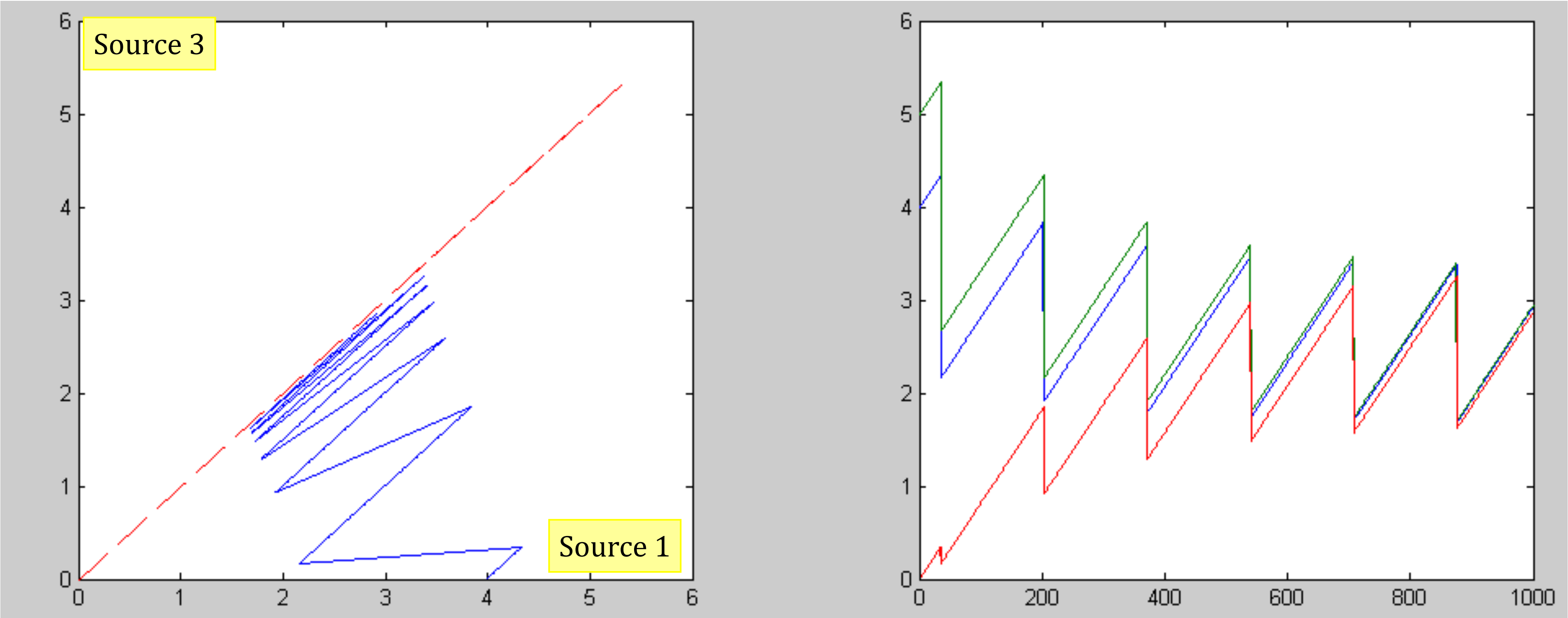
The fairness of AIMD is between max-min and proportional fairness, closer to proportional fairness. [see “Rate adaptation, Congestion Control and Fairness: A Tutorial”]



rescaled utility functions;
AIMD is for $\eta = 0.5$ $r =$ 1MSS per RTT = 100 ms
maxmin approx. is $U(x) = 1 - x^{-5}$

5. Slow Start

- AIMD convergence can be accelerated when initial conditions are very different
- Slow start is an additional method, added to AIMD
- Used at beginning of connection and when losses are detected by timeout

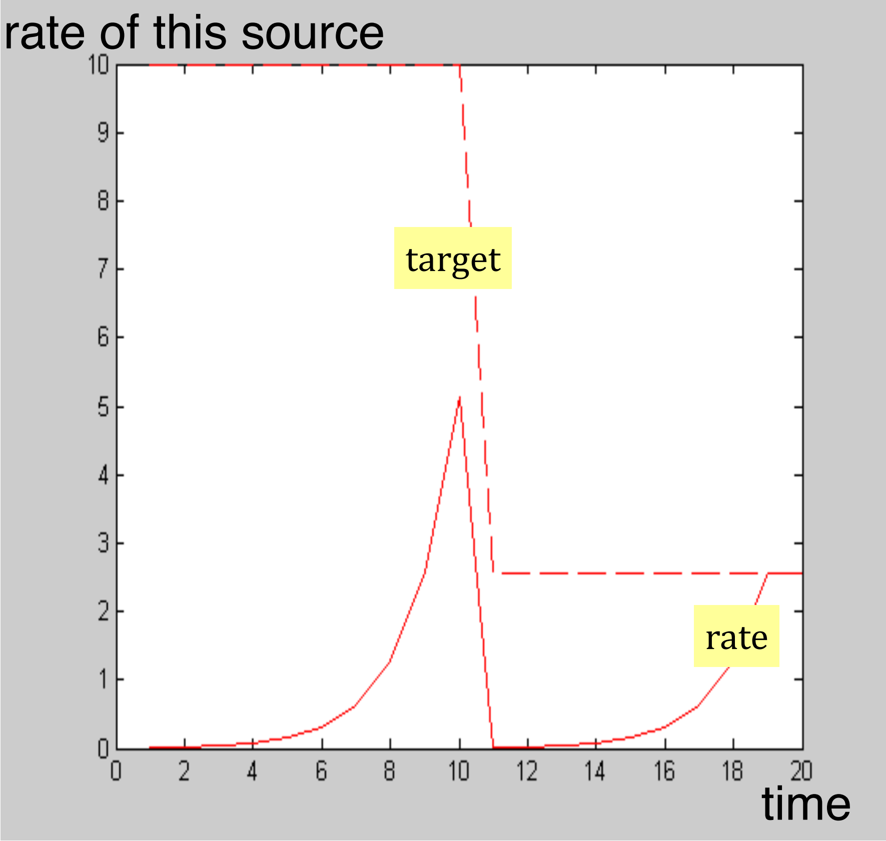


Slow Start

- For a *short* period of time *increase the rate multiplicatively* (by w_0, e.g. $w_0 = 2$) until a target rate is reached or negative feedback is received

- If negative feedback is received, reduce the target rate (by u_1`*actual_rate`, e.g. $u_1 = 0.5$) and restart

- Exit slow start when target rate is reached



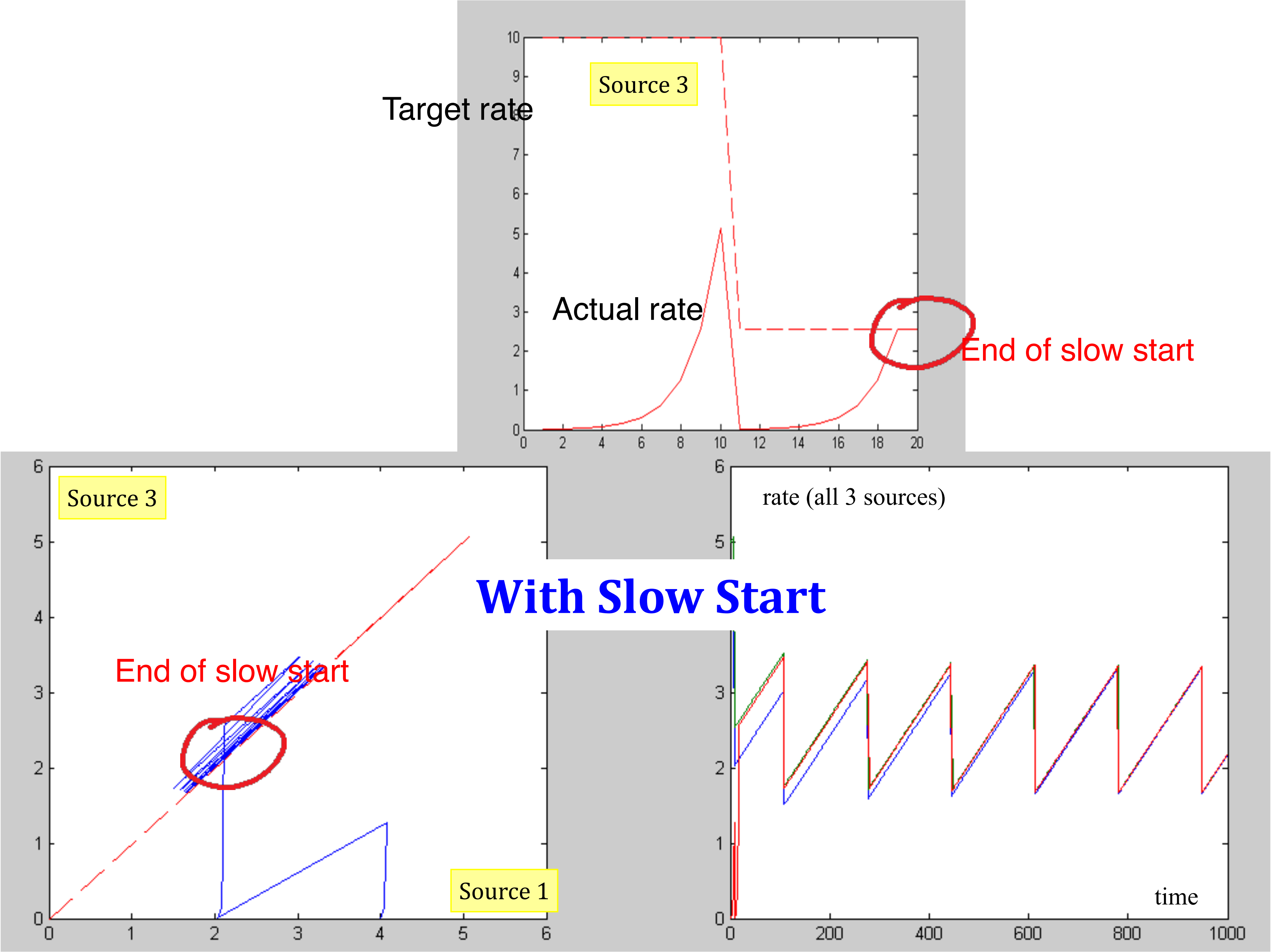
Algorithm 2 Slow Start with the following parameters: AIMD constants multiplicative increase factor $w_0 > 1$; maximum rate $r_{max} > 0$.

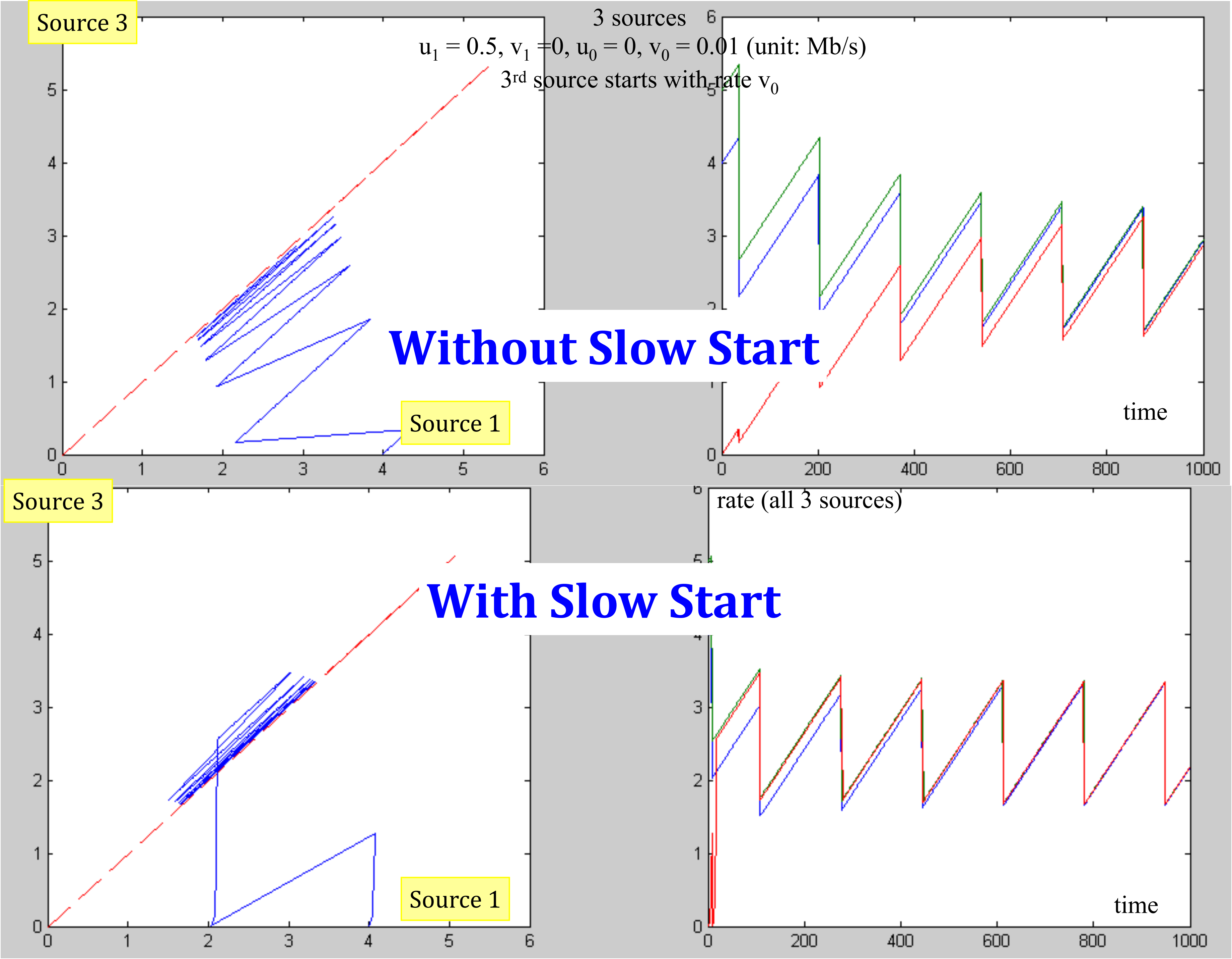
rate $\leftarrow v_0$
targetRate $\leftarrow r_{max}$
do forever
receive feedback
if feedback is positive **then**
    rate $\leftarrow w_0 \cdot$ rate
    if rate $\geq$ targetRate **then**
        rate $\leftarrow$ targetRate
        exit do loop
    end if
else
    targetRate $\leftarrow \max(u_1 \cdot \text{rate}, v_0)$
    rate $\leftarrow v_0$
end if
end do

Multiplicative increase

decrease drastically

With Slow Start




Source 3
3 sources
u1 = 0.5, v1 = 0, u0 = 0, v0 = 0.01 (unit: Mb/s)
3rd source starts with rate v0
Without Slow Start
Source 1
time
Source 3
rate (all 3 sources)
With Slow Start
Source 1
time

Conclusion

- Congestion control is necessary to avoid inefficiencies and congestion collapses
- A congestion control scheme aims at allocating rates according to some form of fairness
- In the internet, we use *end-to-end congestion control* with:

  $\sim$ AIMD

  Slow Start

  and other refinements like fast recovery
  [see part 2: Congestion Control — Implementation]