Advanced Computer Networks

Congestion control

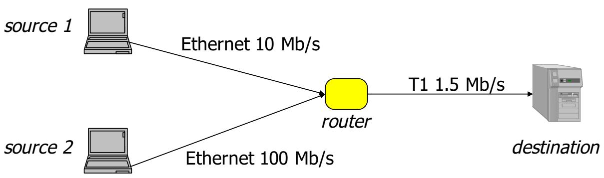
Prof. Andrzej Duda
duda@imag.fr

http://duda.imag.fr

Contents

- Objectives of Congestion Control
  - effciency
  - fairness
- Max-min fairness
- Proportional fairness
- Additive increase, multiplicative decrease
- Different forms of congestion control

Congestion control

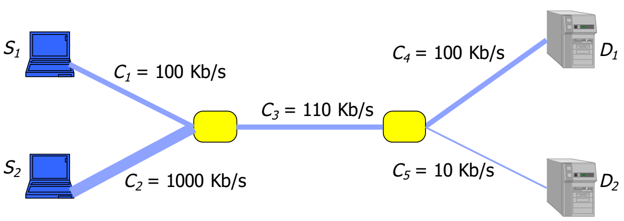


- How to allocate network resources?
  - link capacity
  - buffers at routers or switches
- What to do when the traffic exceeds link capacity?
  - congestion control

Performance criteria

- Efficiency
  - best use of allocated resources
  - max throughput - 100 % utilization
  - min delay - 0 % utilization
- Fairness (équité)
  - fair share to each user
  - different definitions of fairness
    - equal share
    - max-min fairness
    - proportional fairness

Congestion Control - example



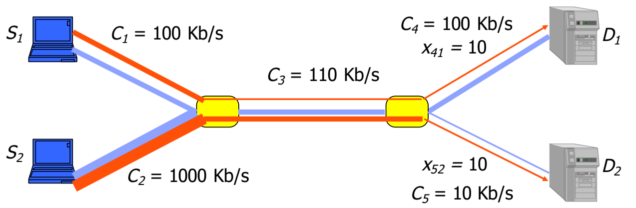
- Sources send as much as possible
- Allocation of throughput
  - if the offered traffic exceeds the capacity of a link, all sources see their traffic reduced in proportion of their offered traffic
  - approximately true if FIFO in routers

Throughput allocation

- Throughput x_{ls} : source s on link l
- Traffic λ_s : generated by source s
- Allocation

Allocation	Our example:
x_{11} = min (λ_1, C_1)	x_{11} = 100
x_{22} = min (λ_2, C_2)	x_{22} = 1000
x_{3i} = min (x_{ii}, $C_3 x_{ii}/(x_{11} + x_{22})$)	x_{31} = 110×100/1100=10
	x_{32} = 110×1000/1100=100
x_{41} = min (x_{31}, C_4)	x_{41} = 10
x_{52} = min (x_{32}, C_5)	x_{52} = 10
throughput $\theta = x_{41} + x_{52}$	throughput θ = 20 Kb/s

Congestion Control - example

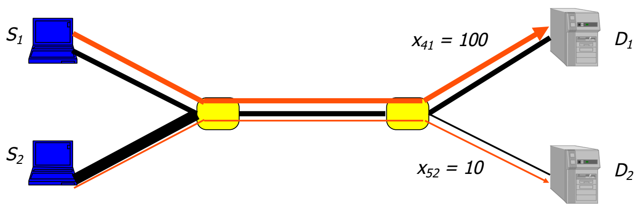


- S1 sends 10 Kb/s because it is competing with S2 on link 3
- S2 is limited on link 5 anyway

Congestion Control - exemple

- How to increase throughput?
  - if S_2 is aware of the global situation and if it would cooperate
  - S_2 reduces x_{22} to 10 Kb/s, because anyway, it cannot send more then 10 Kb/s on link 5
  - x_{31} = 100 Kb/s and x_{41} = 100 Kb/s without any penalty for S_2
  - throughput is now θ = 110 Kb/s

Congestion Control - exemple

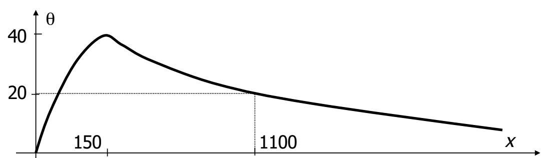


- Optimal use of resources

Efficiency criterion

- In a packet network, sources should limit their sending rate by taking into consideration the state of the network. Ignoring this may put the network into congestion collapse
  - network resources are not used efficiently
  - performance indices perceived by sources are not satisfactory
- One objective of congestion control is to avoid such inefficiencies

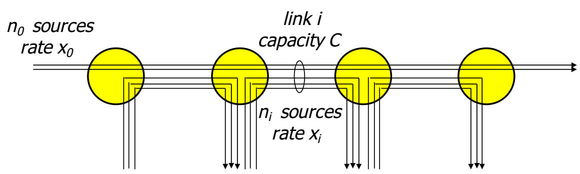
Throughput vs. offered load



- Same example - sources increase their throughput in parallel but at different rate
  - $\lambda_1 = \lambda$, $\lambda_2 = \lambda^2/10$, λ - a parameter
  - λ_1 (1) = 1 , λ_2 (1) = 1/10
  - λ_1 (10) = 10 , λ_2 (10) = 10
  - λ_1 (100) = 100 , λ_2 (100) = 1000
  - offered load $x = \lambda_1 + \lambda_2$
  - x = 1100, θ = 20 Kb/s

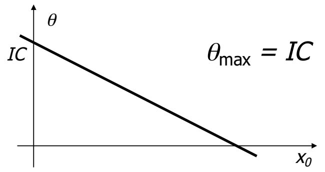
Efficiency versus Fairness

- Parking lot scenario
  - link capacity : C
  - n_i sources, rate x_i , i = 1, ..., I
  - traffic on link i : $n_0\, x_0 + n_i\, x_i$



Maximal throughput

- For given n_0 and x_0, maximizing the throughput requires that
  - $n_i x_i = C - n_0 x_0$
- Total throughput, measured at the network output
  - $\theta = n_0 x_0 + \Sigma n_i x_i = n_0 x_0 + \Sigma(C - n_0 x_0) =$
    $= n_0 x_0 + I(C - n_0 x_0) = IC - (I - 1) n_0 x_0$



$\theta_{max} = IC$ for $x_0 = 0$!

Fairness

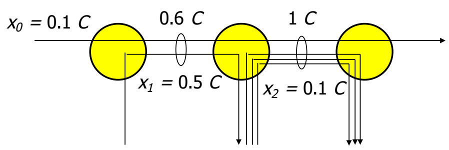
- Maximizing network throughput as a primary objective may lead to large unfairness
  - some sources may get a zero throughput
- Fairness criterion
  - let allocate the same share to all sources, e.g. for $n_i = 1$
    - $x_i = C/2$
    - $\theta_{fair} = (I+1)C/2$
  - roughly half of the maximal throughput

Equal share fairness

- Consider the parking lot scenario for general values of n_i
  - equal share on link i
    - $x_i = C / (n_0 + n_i)$, $i = 1, ..., I$
  - let decrease x_0 to increase θ (we have seen that this maximizes throughput)
    - $x_0 = \min C / (n_0 + n_i)$,
  - example
    - $I = 2$, $n_0 = n_1 = 1$, $n_2 = 9$
    - link 2: $x_2 = C / (1 + 9) = 0.1\ C$
    - link 1: $x_1 = C / (1 + 1) = 0.5\ C$
    - $x_0 = \min (0.5\ C, 0.1\ C) = 0.1\ C$
- Allocating equal shares is not a good solution
  - some flows can get more

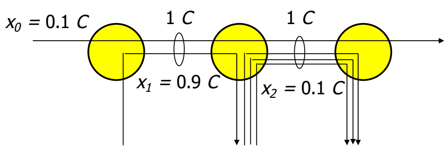
Example

- Problem
  - link 1 : 0.6 C
    - underutilized
  - link 2 : 1 C



Max-Min Fairness

- We can increase x_1 without penalty for other flows
  - $x_0 = 0.1\ C$, $x_1 = 0.9\ C$, $x_2 = 0.1\ C$



Max-Min Fairness

- Allocating resources in an equal proportion is not a good solution since some sources can get more that others without decreasing others' shares
- Max-Min fair allocation
  - Min: because of the fairness on bottleneck links
  - Max: because we can increase throughput whenever possible

Progressive filling

- Bottleneck link l for source s
  - link l is saturated : $\Sigma\ x_i = C$
  - source s on link l has the maximum rate among all sources using that link
- Progressive filling allocation
  - $x_i = 0$
  - increase x_i equally until $\Sigma\ x_i = C$
  - rates for the sources that use this link are not increased any more
    - all the sources that do not increase have a bottleneck link (Min)
  - continue increasing the rates for other sources (Max)

Example

- Parking lot scenario
  - $x_i = 0$
  - $x_i = d$ until $n_0 x_0 + n_i x_i \leq C$
  - bottleneck link for $d_1 = \min(C / (n_0 + n_i))$, source 0 or i
    - $x_0 = \min(C / (n_0 + n_i))$
  - increase other sources
    - $x_i = (C - n_0 x_0) / n_i$
- In our example
  - $x_0 = 0.1\ C$, $x_2 = 0.1\ C$
  - $x_1 = 0.9\ C$

Proportional Fairness

- Equal share fairness and Max-min fairness
  - per link only
  - do not take into account the number of links used by a flow
  - flows x_0 benefit from more network resources than flows x_i
- Another fairness
  - give higher throughput to flows that use less resources
  - give smaller throughput to flows that use more resources
- Proportional fairness

Proportional Fairness

- An allocation of rates x_s is *proportionally fair* if and only if, for any other feasible allocation y_s we have (S sources)

$$\sum_{s=1}^{S} \frac{y_s - x_s}{x_s} \leq 0$$

- Any change in the allocation must have a negative average change
- Parking lot example with $n_s = 1$
  - max-min fair allocation $x_s = C/2$ for all s
  - let decrease x_0 by δ : $y_0 = C/2 - \delta$, $y_s = C/2 + \delta$, $s = 1, \ldots, I$
  - average rate of change is positive - not proportionally fair for $I \geq 2$!

$$\left(\sum_{s=1}^{I} \frac{2\delta}{c}\right) - \frac{2\delta}{c} = \frac{2(I-1)\delta}{c}$$

Proportional Fairness

- There exists one unique proportionally fair allocation. It is obtained by maximizing

$$J(\vec{x}) = \sum_{s} \ln(x_s)$$

over the set of feasible allocations for all sources s

Parking lot example

- For any choice of x_0 we should set x_i such that
  - $n_0 x_0 + n_i x_i = C$, $i = 1, \ldots, I$
- Maximize

$$f(x_0) = n_0 \ln(x_0) + \sum_{i=1}^{I} n_i(\ln(C - n_0 x_0) - \ln(n_i))$$

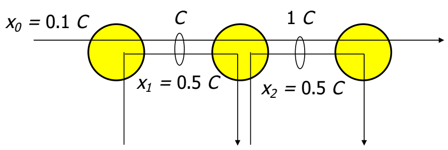
  over the set $0 \leq x_0 \leq C / n_0$.
- The maximum is for

$$x_0 = \frac{C}{\sum_{i=0}^{I} n_i} \qquad x_i = \frac{C - n_0 x_0}{n_i}$$

- If $n_i = 1$, $x_0 = C/(I+1)$, $x_i = CI/(I+1)$
- Max-min allocation is $C/2$ for all rates - sources of type 0 get a smaller rate, since they use more network resources

Comparisons

- $I = 2$, $n_i = 1$
- max throughput:
  - $x_0 = 0$, throughput = $2C$
- equal-share and max-min:
  - $x_0 = C/2$, $x_i = C/2$, throughput = $1.5C$
- proportional fairness:
  - $x_0 = C/3$, $x_i = 2C/3$, throughput = $5C/3$



End-to-end congestion control

- End-to-end congestion control
  - binary feedback from the network: congestion or not
  - rate adaptation mechanism: decrease or increase
- Modeling
  - I sources, rate $x_i(t)$, $i = 1, \ldots, I$
  - link capacity: C
  - discrete time, feedback cycle = one time unit
  - during one time cycle, the source rates are constant, and the network generates a binary feedback signal $y(t) \in \{0, 1\}$
  - sources: increase the rate if $y(t) = 0$ and decrease if $y(t) = 1$
  - feedback

$$y(t) = [if\ (\sum_{i=1}^{I} x_i(t) \le c)\ then\ 0\ else\ 1]$$

Linear adaptation algorithm

- Find constants u_0, u_1, v_0, v_1, such that

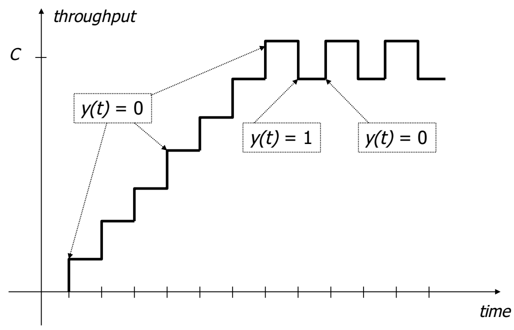
$$x_i(t+1) = u_{y(t)}\, x_i(t) + v_{y(t)}$$

  - we want to converge towards a fair allocation
  - one single bottleneck, so all fairness criteria are equivalent
  - we should have $x_i = C/I$
  - the total throughput

$$f(t) = \sum_{i=1}^{I} x_i(t)$$

should oscillate around C : it should remain below C until it exceeds it once, then return below C

Linear adaptation algorithm



Necessary conditions

$$f(t+1) = u_{y(t)}\, f(t) + v_{y(t)}$$

- we must have

  $u_0 f + v_0 > f$, increase rate if feedback 0

  $u_1 f + v_1 < f$, decrease rate if feedback 1
- this gives the following conditions

  $u_1 < 1$ and $v_1 \le 0$ (A)

  or

  $u_1 = 1$ and $v_1 < 0$ (B)

  and

  $u_0 > 1$ and $v_0 \ge 0$ (C)

  or

  $u_0 = 1$ and $v_0 > 0$ (D)

Ensure fairness

$u_1 < 1$ and $v_1 \le 0$ (A)

or

$u_1 = 1$ and $v_1 < 0$ (B)

and

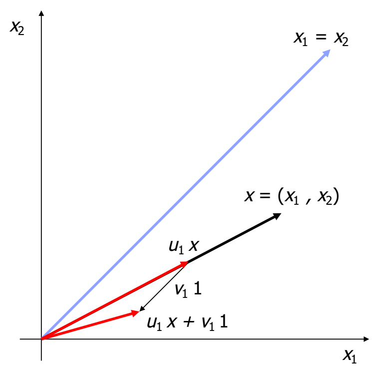
$u_0 > 1$ and $v_0 \ge 0$ (C)

or

$u_0 = 1$ and $v_0 > 0$ (D)

how to decrease rate?

$v_1 = 0$, $u_1 < 1$

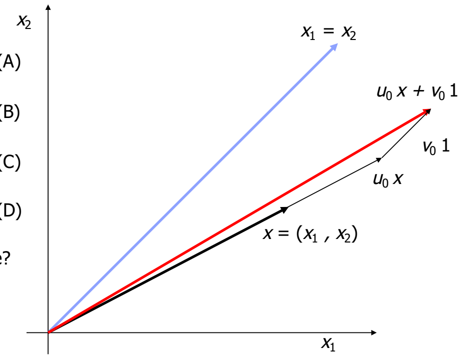


Ensure fairness

$u_1 < 1$ and $v_1 \le 0$ (A)
or
$u_1 = 1$ and $v_1 < 0$ (B)
and
$u_0 > 1$ and $v_0 \ge 0$ (C)
or
$u_0 = 1$ and $v_0 > 0$ (D)

how to increase rate?

$v_0 > 0,\ u_0 > 1$ or
$v_0 > 0,\ u_0 = 1$

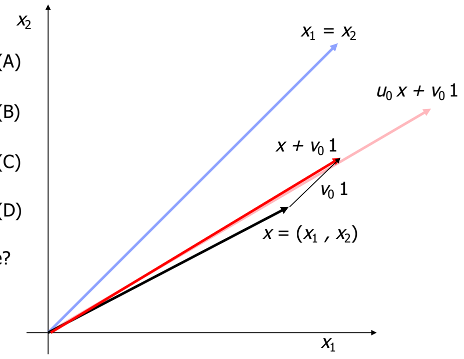


Ensure fairness

$u_1 < 1$ and $v_1 \le 0$ (A)
or
$u_1 = 1$ and $v_1 < 0$ (B)
and
$u_0 > 1$ and $v_0 \ge 0$ (C)
or
$u_0 = 1$ and $v_0 > 0$ (D)

how to increase rate?

$v_0 > 0,\ u_0 = 1$



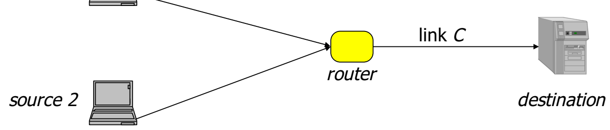
Ensure fairness

- When we apply a multiplicative increase or decrease, the unfairness is unchanged
- An additive increase decreases the unfairness, whereas an additive decrease increases the unfairness
- To obtain that unfairness decreases or remains the same, and such that in the long term it decreases
  - $v_1 = 0$ decrease must be **multiplicative**
  - $u_0 = 1$ increase must be **additive**

Result

- Fact
  - In order to satisfy efficiency and convergence to fairness, we must have a multiplicative decrease (namely, $u_1 < 1$ and $v_1 = 0$ and a non-zero additive component in the increase (namely, $u_0 \ge 1$ and $v_0 > 0$).
  - If we want to favour a rapid convergence towards fairness, then the increase should be additive only (namely, $u_0 = 1$ and $v_0 > 0$).
- **Additive increase, Multiplicative decrease**

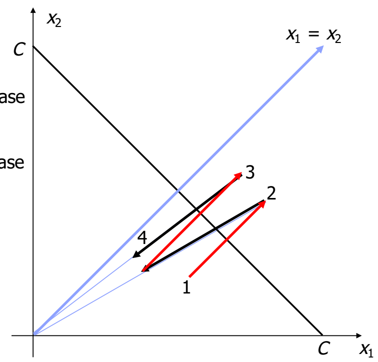
Why AI-MD works?



- Simple scenario with two sources sharing a bottleneck link of capacity C

Throughput of sources

1. Additive increase
2. Multiplicative decrease
3. Additive increase
4. Multiplicative decrease



Different types of CC

- Router/Switch centric (ATM)
  - switch decides which packet transmit or discard
  - switch notifies the source at which rate it should send
- Open loop (ATM)
  - resource reservation
  - admission control
- Host centric (TCP)
  - host observes the network and adjust the rate
- Closed loop with feedback
  - information on congestion state
    - implicit - packet loss (TCP)
    - explicit (RTCP)

Different types of CC

- Rate-based control
  - negociated with network
  - adjusted if needed
  - ATM, RTP
- Window-based control
  - defines the volume of data to send
  - TCP
- Open loop implies
  - Router/Switch centric
  - rate-based control

Facts to remember

- In a packet network, sources should limit their sending rate by taking into consideration the state of the network
- Maximizing network throughput as a primary objective may lead to large unfairness
- Objective of congestion control is to provide both efficiency and some form of fairness
- Fairness can be defined in various ways: equal share, max-min, proportional
- End-to-end congestion control in packet networks is based on binary feedback and the adaptation mechanism of additive increase, multiplicative decrease.