Fairness, Flow Control & Multi-User Games

Peter Key
peterkey@microsoft.com

Partners

- Frank Kelly, Richard Gibbens (Stats Lab)
- Derek McAuley, Dave Stewart (Microsoft Res Ltd)

Current Solutions

eg *ATM*

- ‘Police users’
- Tight QoS guarantees
- Too complex for users
- C.O. (inelastic bias)

- eg *Internet*
- ‘Best effort’ (!)=1 QoS
- Low usage users subsidise heavy users
- Relies on ‘nice’ users (eg compliant TCP)
- Fails to account for congestion externality

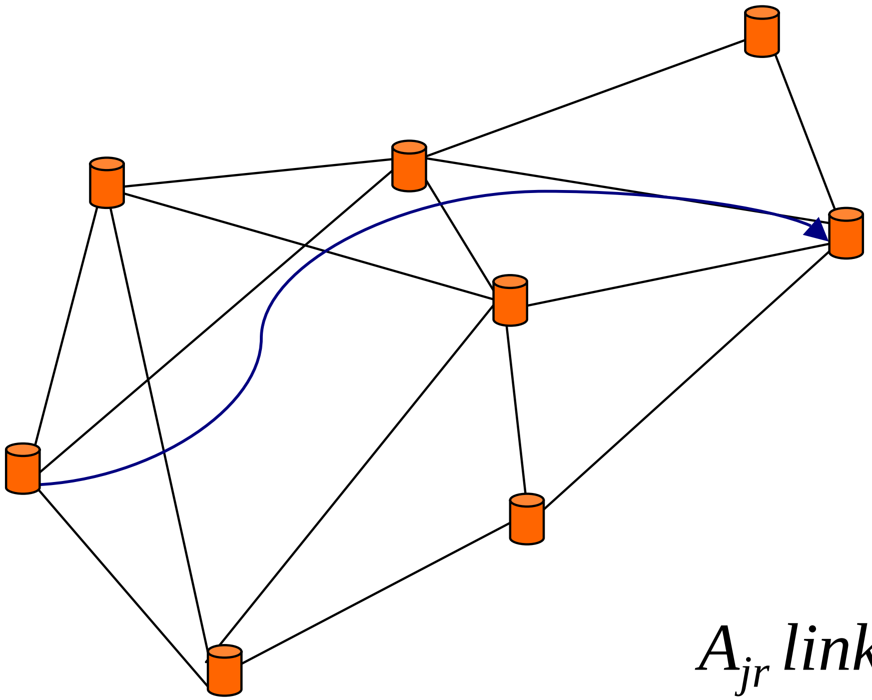
What is QoS?

- Current thinking tends toward definitions involving mappings on a multidimensional space (eg CCITT series of recommendations have a whole layered structure)
- Most SLAs for telecommunications define quantities that are not measurable (especially by provider).
- Is this complexity necessary - other services are differentiated by a few criteria?
- Only need to be have non-identical measures to apply Cake-slicing theorem!

Resource system ('network')

Resource $j \in \mathcal{J}$

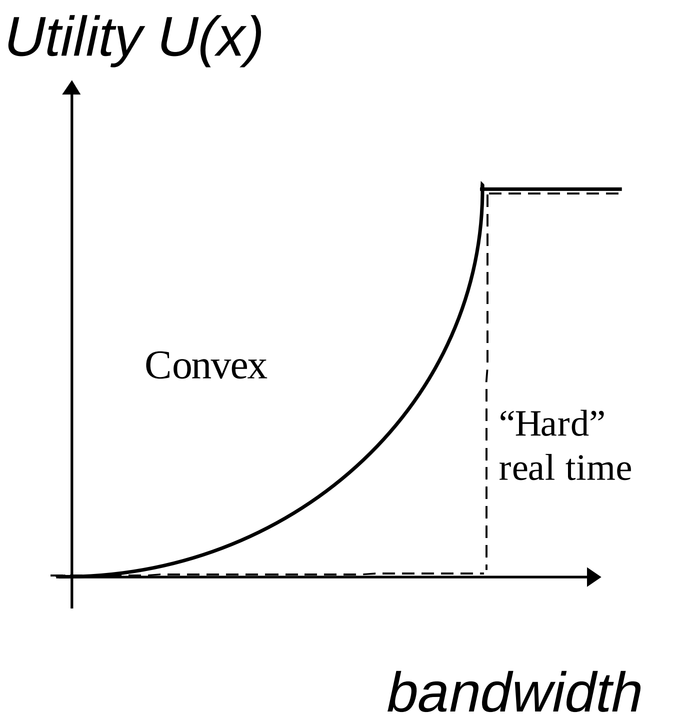
Capacity C_j

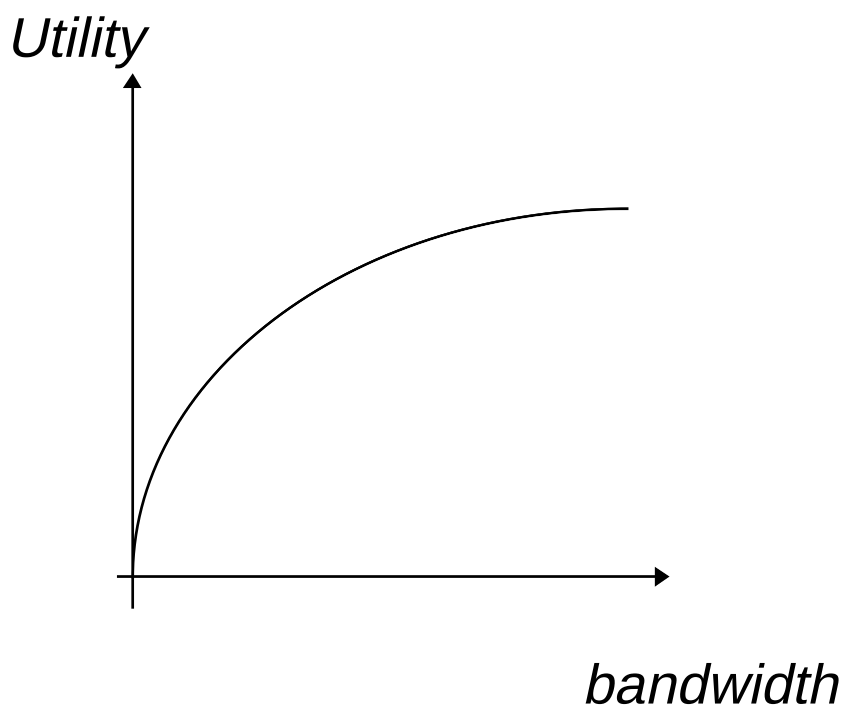


User /route $r \in \mathcal{R}$

A_{jr} links users to resources

Service Requirements (Shenker)

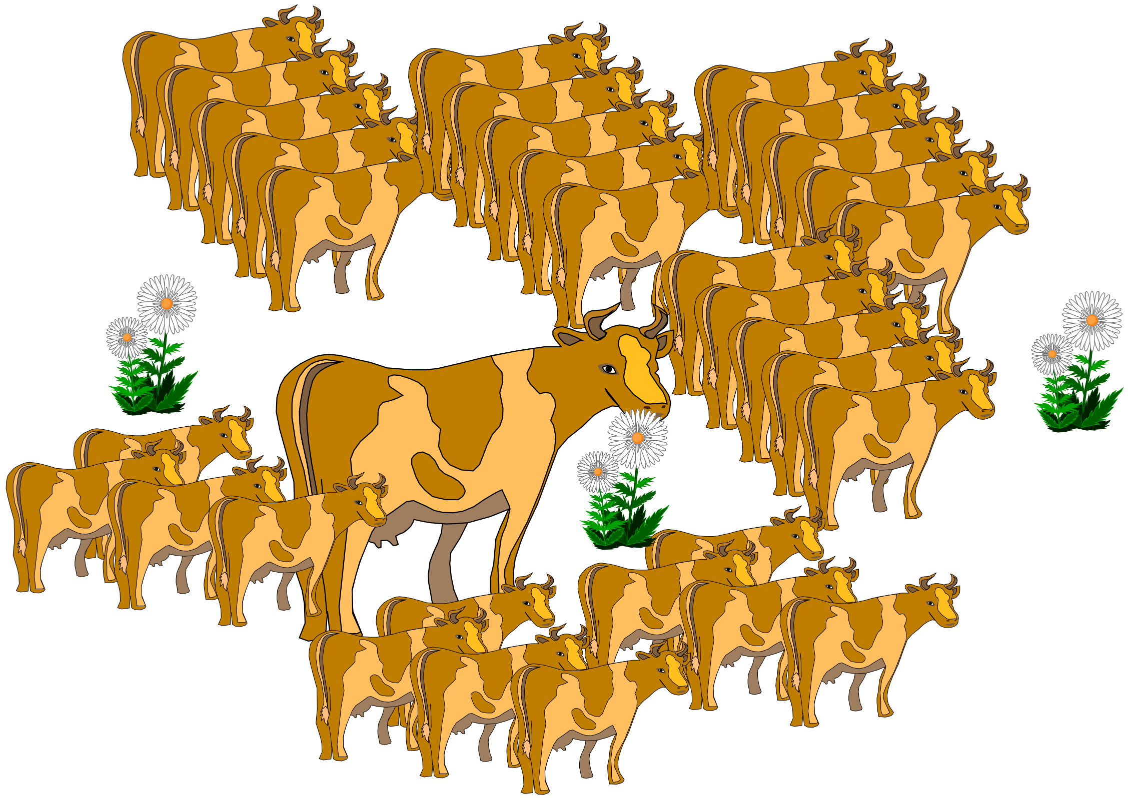




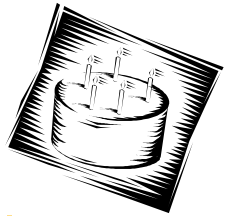
Real Time
Limited capacity → Rejection
Telcos

Data
No calls rejected
Share bandwidth
The Internet!

The problem of the commons



Fairness



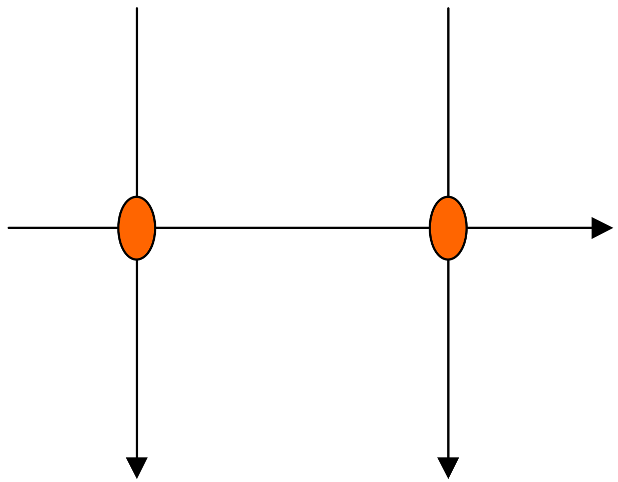
- There are lots of ways to be fair!
- In a network context, Max-Min and Max-Utilisation are like "boundary" values
- Proportional fairness has links with Nash arbitration scheme

What is fair??

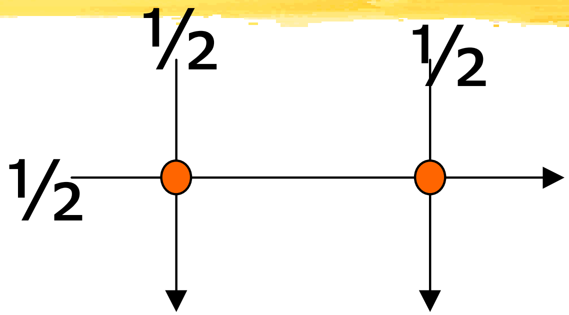
- As many definitions as utility functions (preference functions on x s.t $Ax \leq C$)
- Max-min fair popular
- Proportional fairness advocated by eg Kelly
- Revenue maximisation another option

Fairness Examples

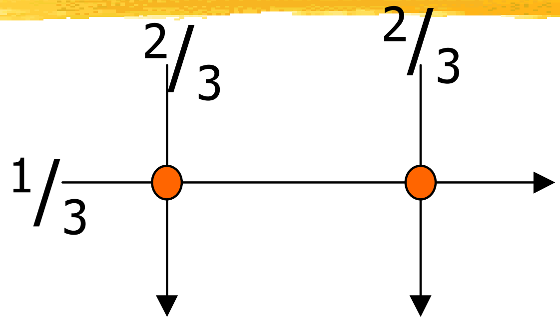
- Two resources
- Each with capacity normalised to 1
- Vertical streams use one resource
- Horizontal streams use two resources



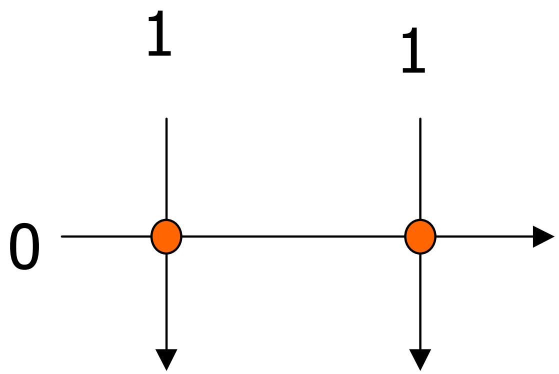
Fairness Examples, eg



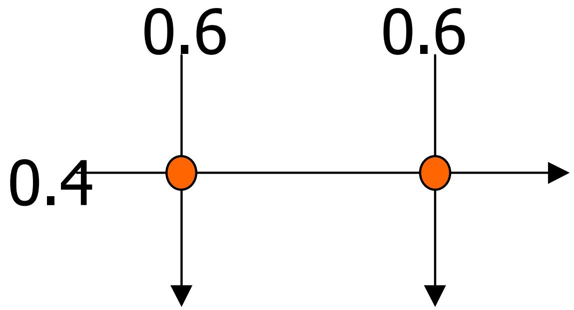
Max-min



Proportional

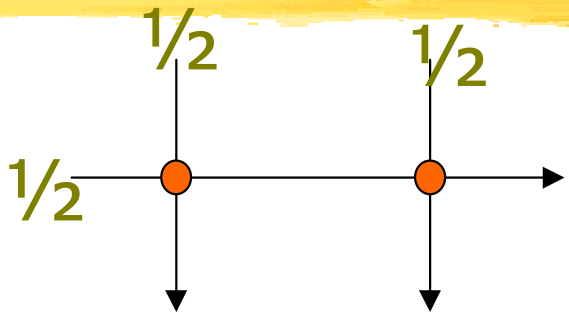


Max load



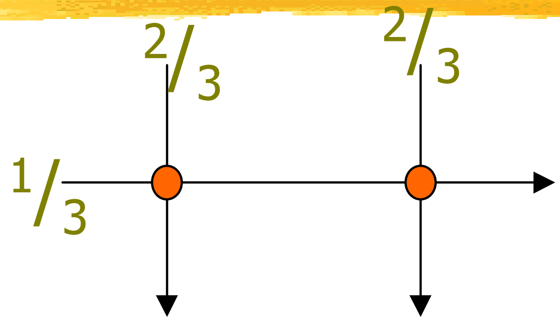
TCP approx

Fairness Examples, eg



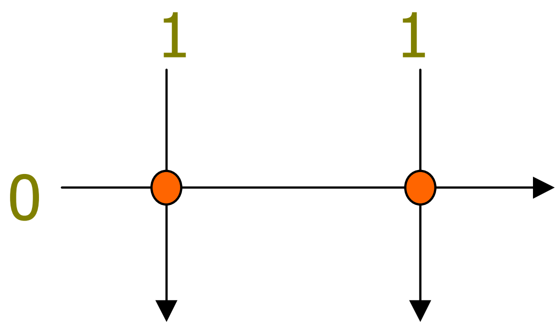
Max-min

$$U(x) = -(-\mathrm{Log}(x))^{\infty}$$



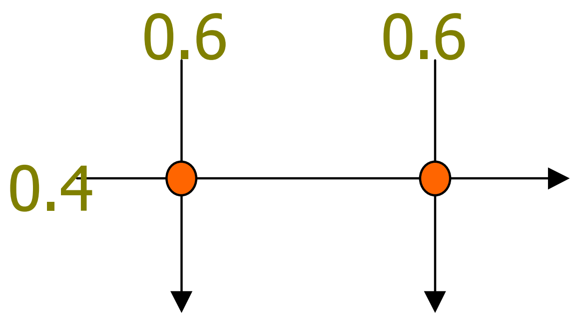
Proportional

$$U(x) = \mathrm{Log}(x)$$



Max load

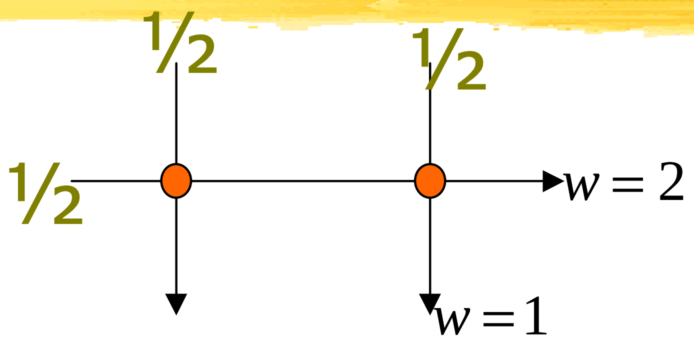
$$U(x) = \mathrm{Log}(1 + x)$$



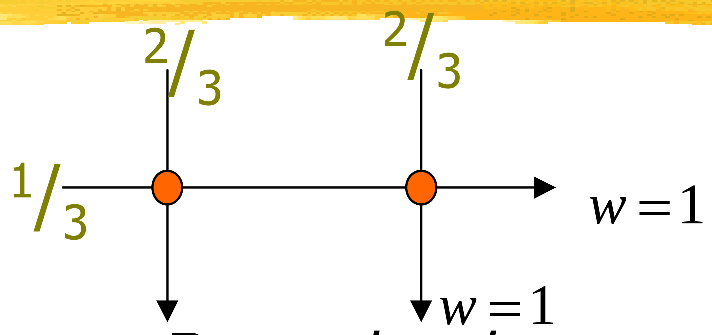
TCP approx

$$U(x) = -(-\mathrm{Log}(x))^{3/2}$$

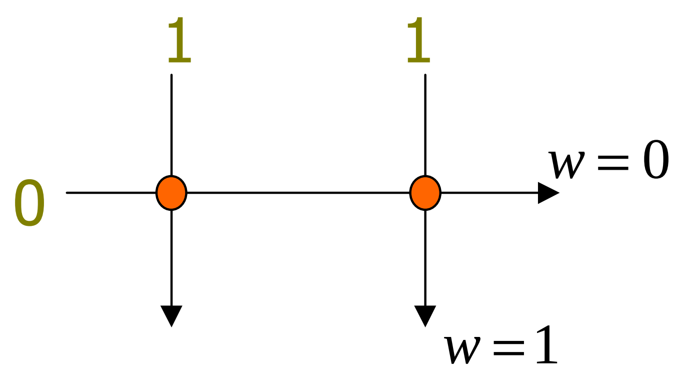
Fairness Examples, prop. fair prices



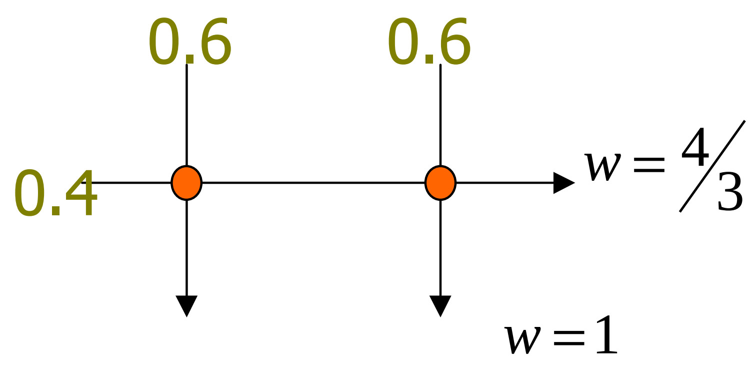
Max-min



Proportional



Max load



TCP approx

Some problems with TCP (CA mode)

- Lose cells
  - hence keeps network at high load (but …)
  - slow to react (queue size rather than rate)
- If p is probability lose cell, throughput $\propto 1/\sqrt{p}$

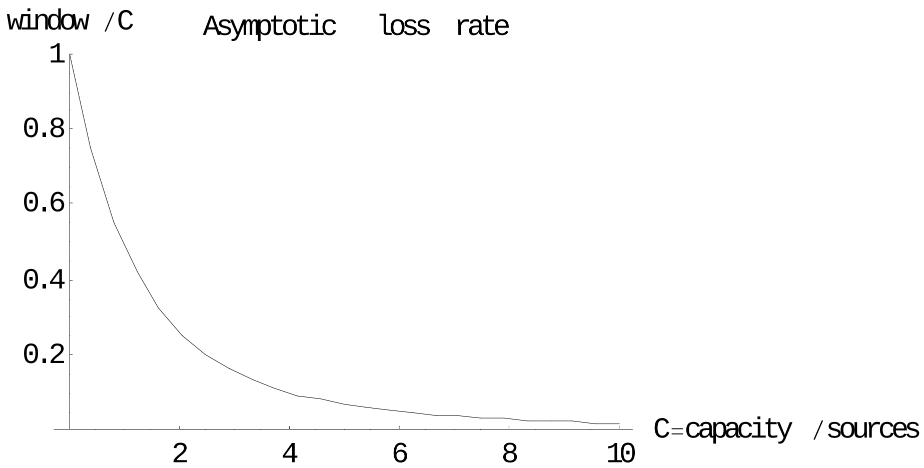
  (should be 1/p)
- Does not scale, eg if double capacity
  - wrong shadow prices
  - only works if “capacity” > number of sources, where capacity = (bottleneck x rtt/mss)

Simple TCP model with feedback

Resource loses excess load, *n* number of sources,

nC capacity (=n x resource capacity*RTT/MSS),

RTT=1ms, cap=1MB/s, C=244



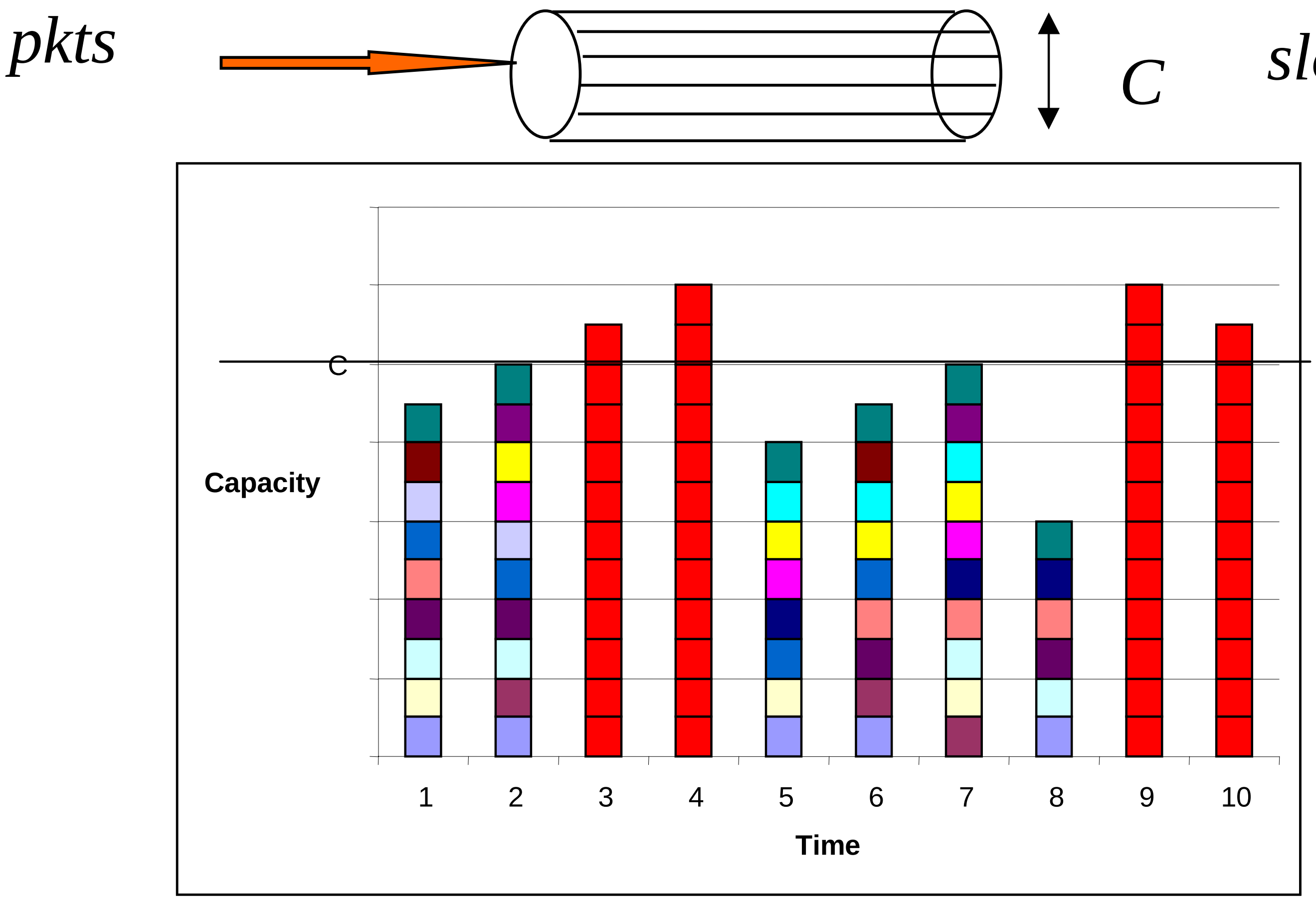
window=

$$\frac{1}{2}\left(C+\sqrt{8+C^2}\right)$$

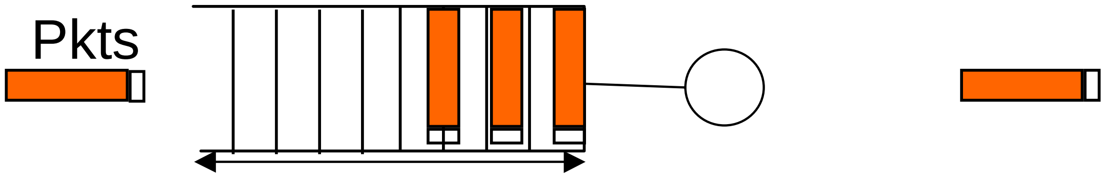
Framework (Gibbens &Kelly)

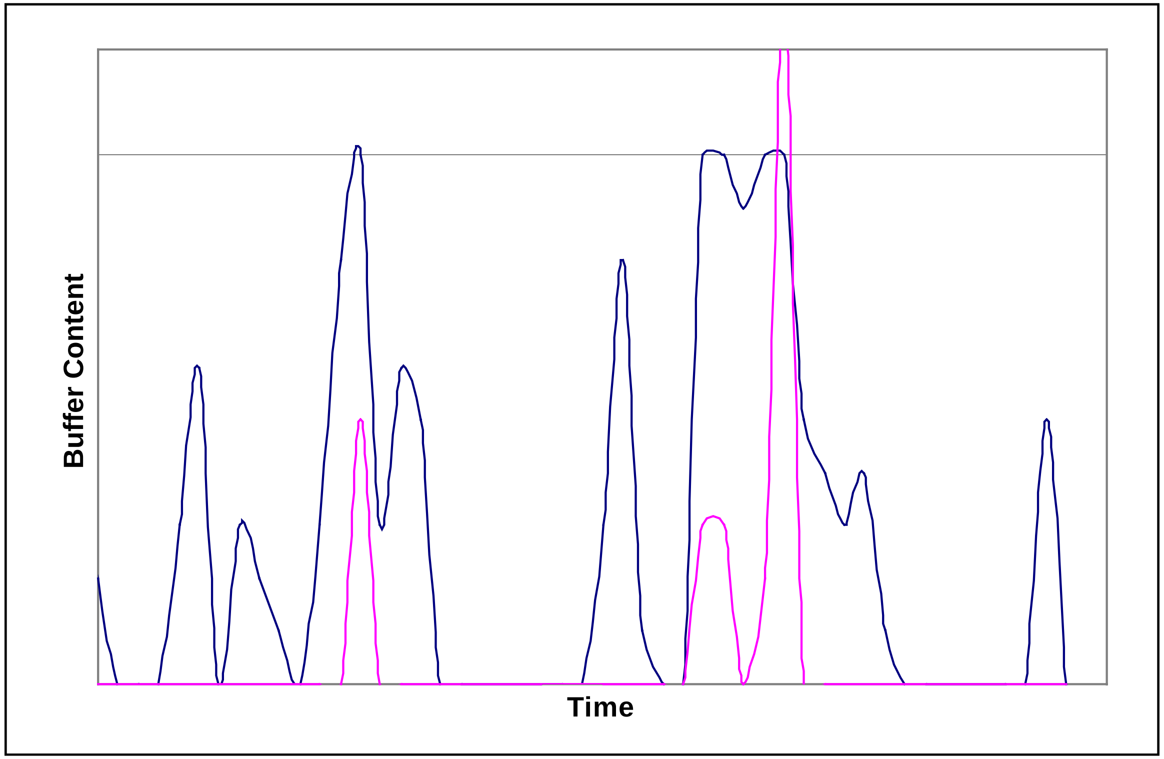
- Users generate load
- Feedback signal is “sample shadow price”
  - eg in simple slotted model, mark every packet if load exceeds capacity: *shadow price* as describes effect of adding extra packet
  - if a buffered system, mark packets from start of busy period to last packet lost in this period

Sample Path Shadow Prices

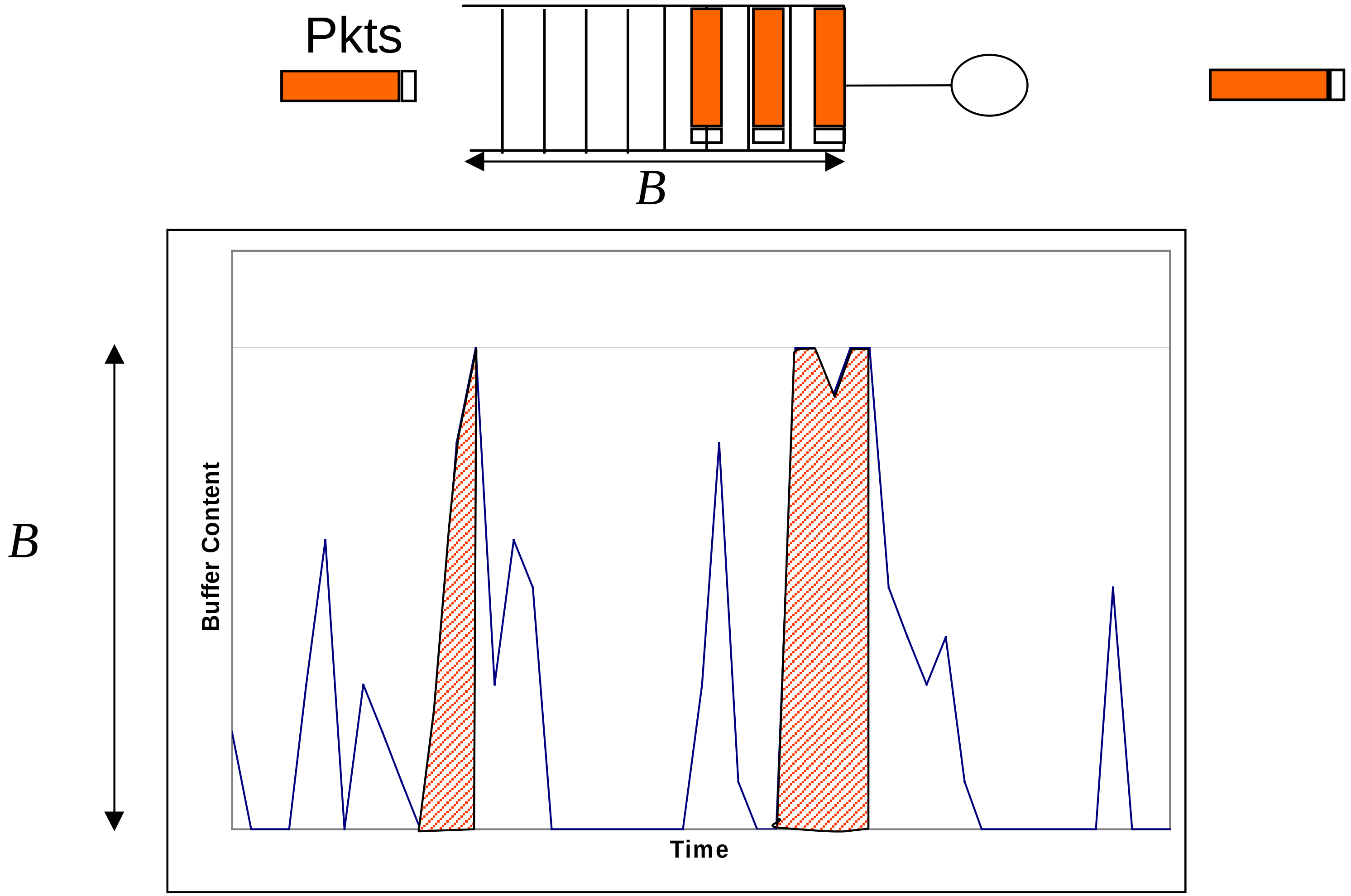


Sample path Shadow prices, buffered model





Shadow path shadow prices -buffered model



Features of Efficient Pricing Scheme

- Those with the highest cost of delay get served first
- Prices send the right signals for capacity expansion
  - If all of the congestion revenues are reinvested in new capacity,then capacity will be expanded to the point where its marginal value is equal to its marginal cost
- eg “Smart market” approach of Mackie-Mason/Varian, though some issues with their model
- Prices related to congestion

Optimisation Framework
(for fairness)

$$Max \quad \mathbf{U} = \sum_r U(x_r)$$

$$s.t. \; Ax \le C$$

System optimum

U is utility

$$Max \quad \mathbf{U} = \sum_r U(x_r) - \sum_j C_j(y)$$

$$y = \sum_r A_{jr} x_r$$

C is cost function,eg

$$C_j = (y - C_j)^+$$

$$Max \quad U(x_r) - t x_r$$

User optimum

Solution

Consistent set of taxes (prices) and load exist s.t.

$$U'(x_r) = \sum_{j \in r} \mu_j \quad \text{if } x_r > 0$$

$$\mu_j = \frac{d\mathbf{U}}{dC_j} \; \textit{constrained} \text{ case}$$

$$\mu_j = \frac{dC_j}{dy} \quad \text{unconstrained}$$

$$U'(x_r) = t_r$$

Eg Network chooses taxes, user chooses load, solution is network, user and System optimal. But dependent on Utility function, so ….

Example

$$C_j(y) = E(Y - c_j)^+ \quad \Rightarrow \mu_j = \frac{dC_j}{dy} = \Pr\{Y \geq c_j\}$$

if Y Poisson and $E(Y) = y$.

Also,

$$E(x_r I\{Y \geq c\}) = x_r \Pr\{Y \geq c_j\}$$

Hence mark packet if exceed capacity

$$\text{If } U(x_r) = w_r \operatorname{Log}(x_r), \quad \text{then}$$

$$w_r = x_r \Pr\{Y \geq c_j\}$$

Adaptive (prop fair) scheme

Suggests the adaptive approach

$$\frac{d}{dt} x_r(t) = \kappa_r \left(w_r(t) - x_r(t) \sum_{j \in r} \mu_j(t) \right)$$

where $\mu_j(t) = p_j\left(\sum_{j \in r} x_r(t) \right), \quad \frac{d}{dy} C_j(y) = p_j(y)$

$C_j(y)$ is cost function on resource j when load is y

maximises $\sum_r U_r(x_r) - \sum_j C_j\left(\sum_{j \in r} x_r \right)$

Example - elastic control

$$x_{t+1} = x_t + \kappa\left(w_t - f\left(t\right)\right)$$

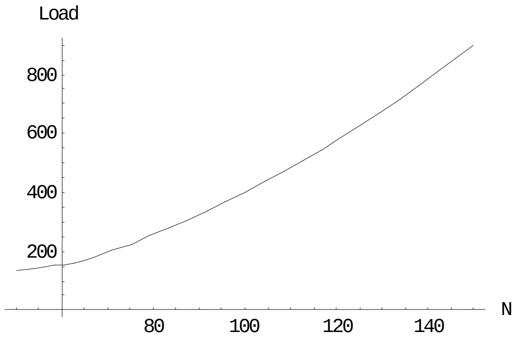
w_t reflects willingness to pay,

$f\left(t\right)$ is feedback received from the network

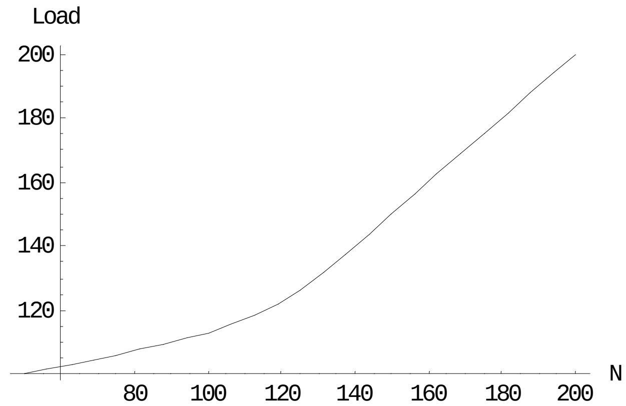
eg

$f\left(t\right) = x_t$ if (resource/ bottleneck overloaded) else $= 0$

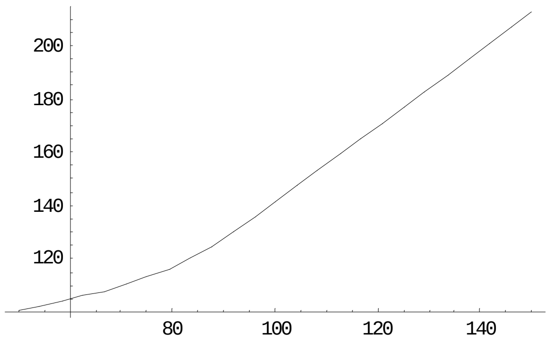
TCP & TCP like schemes



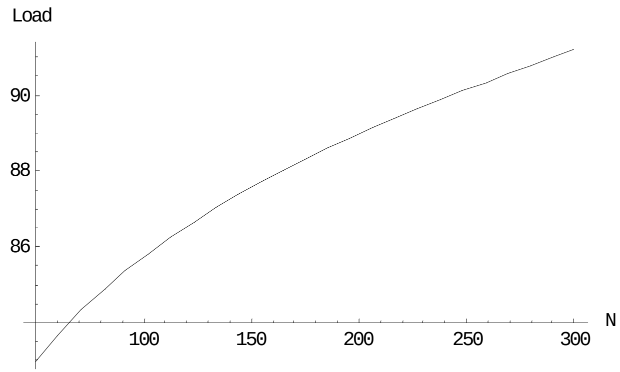
TCP



w=1

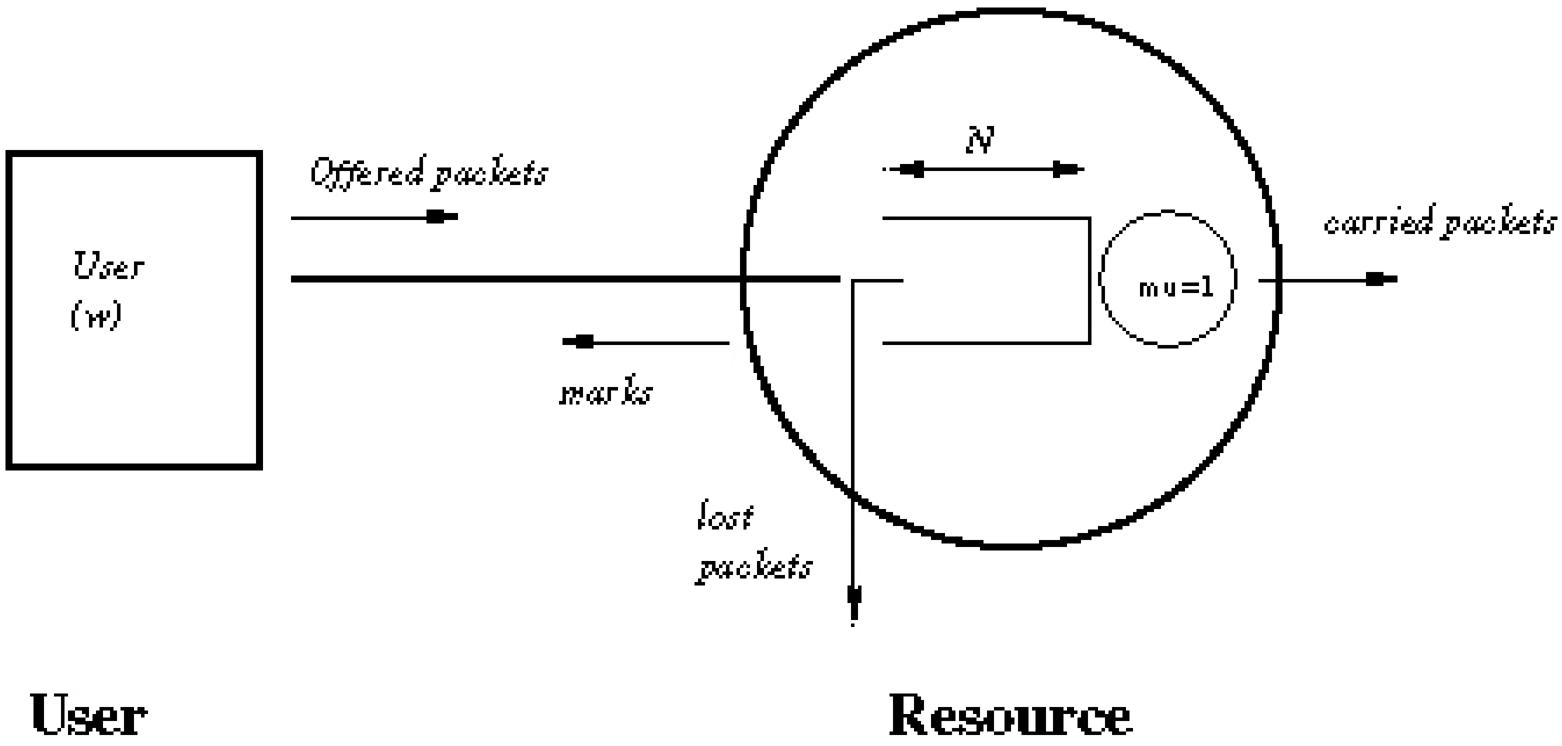


TCP with shadow price

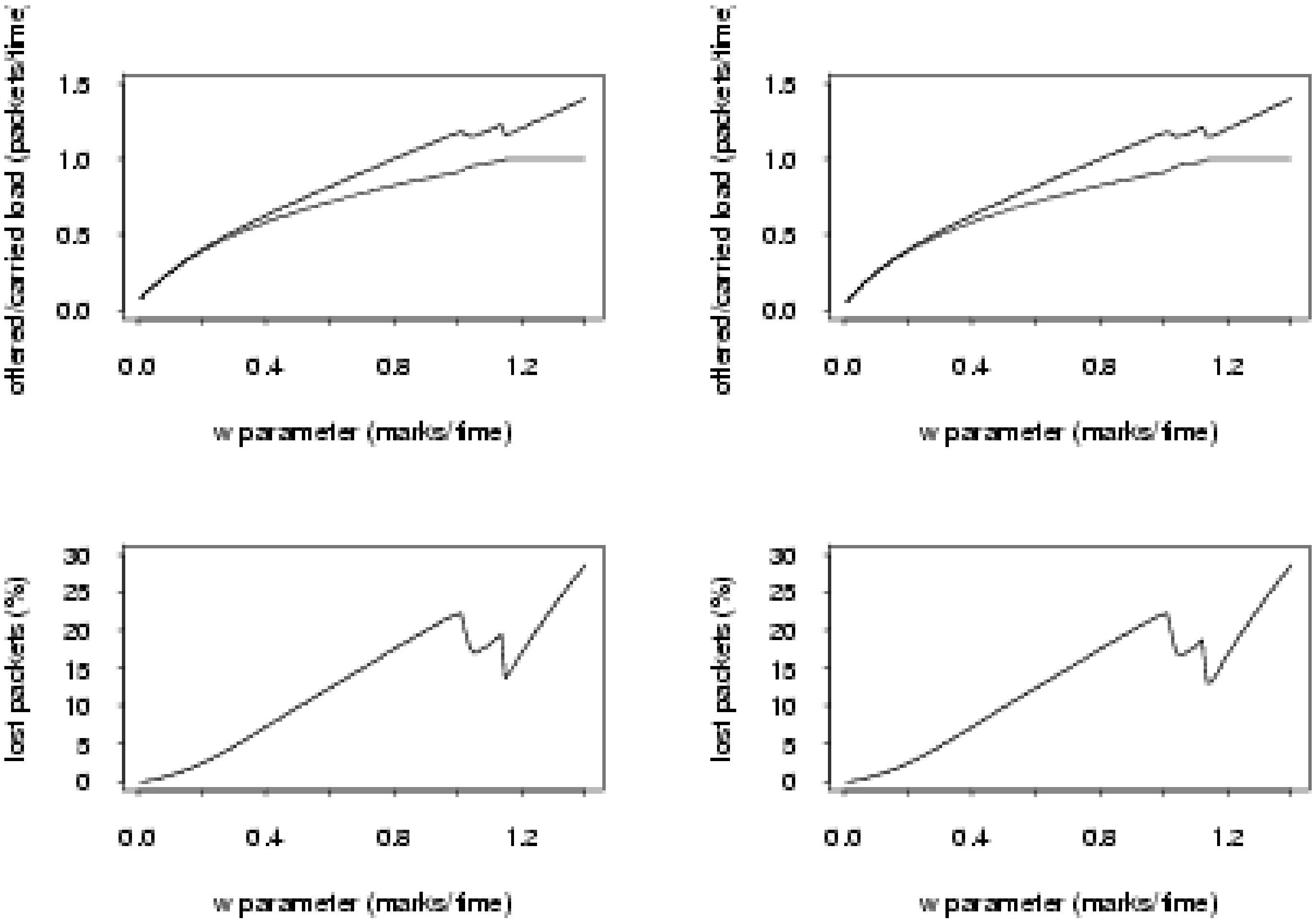


w=0.05

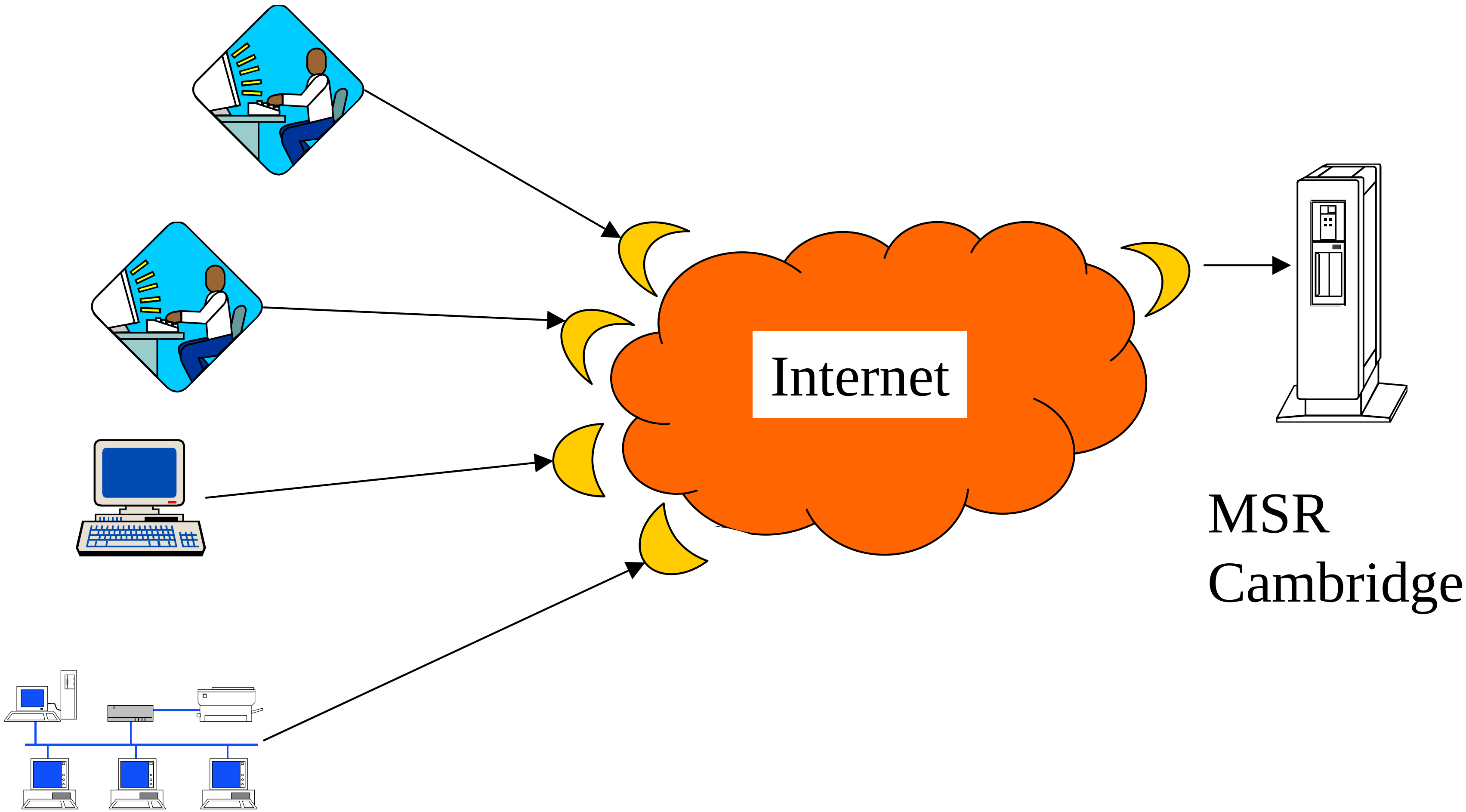
Single node network



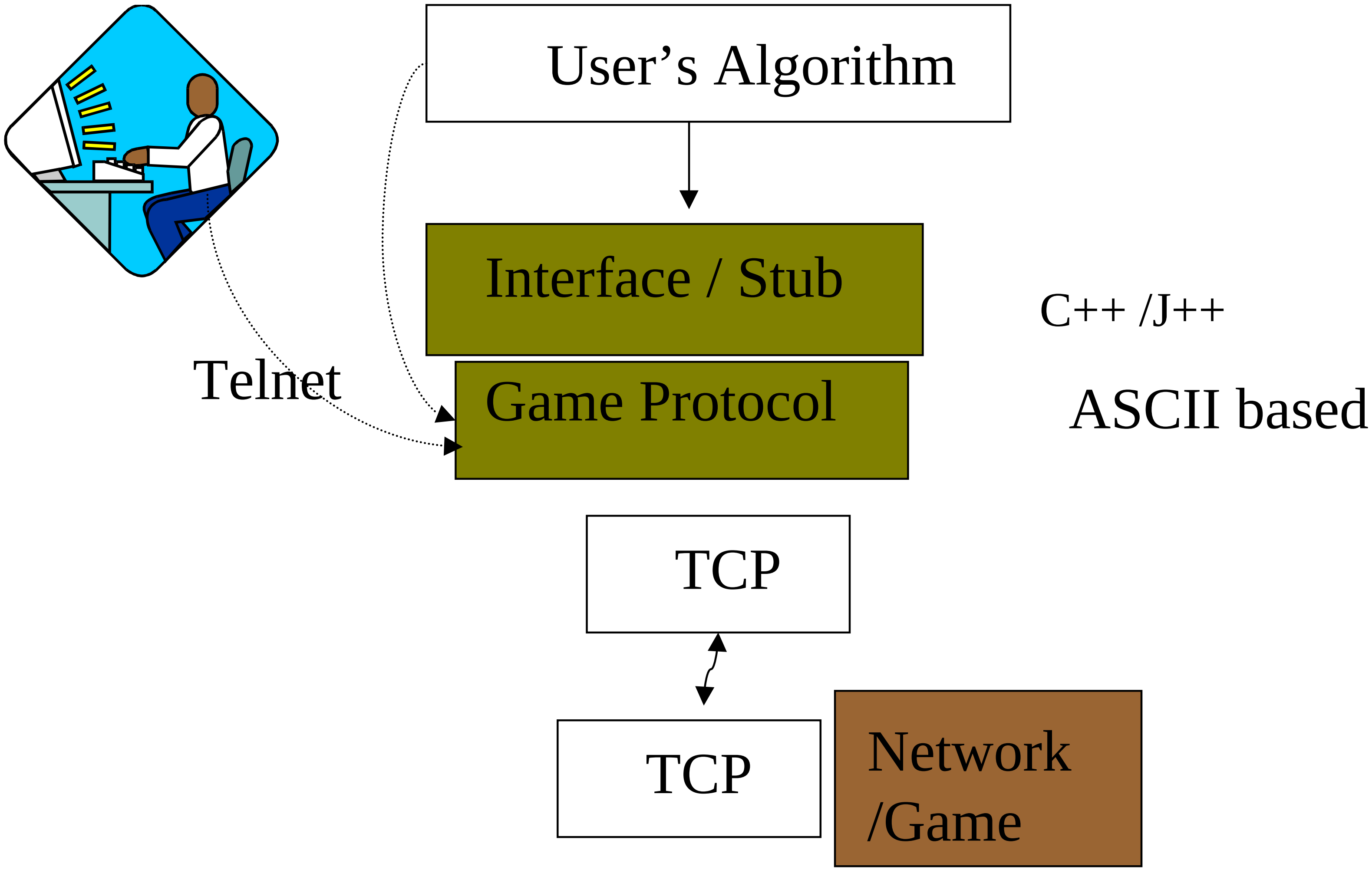
Buffered Model



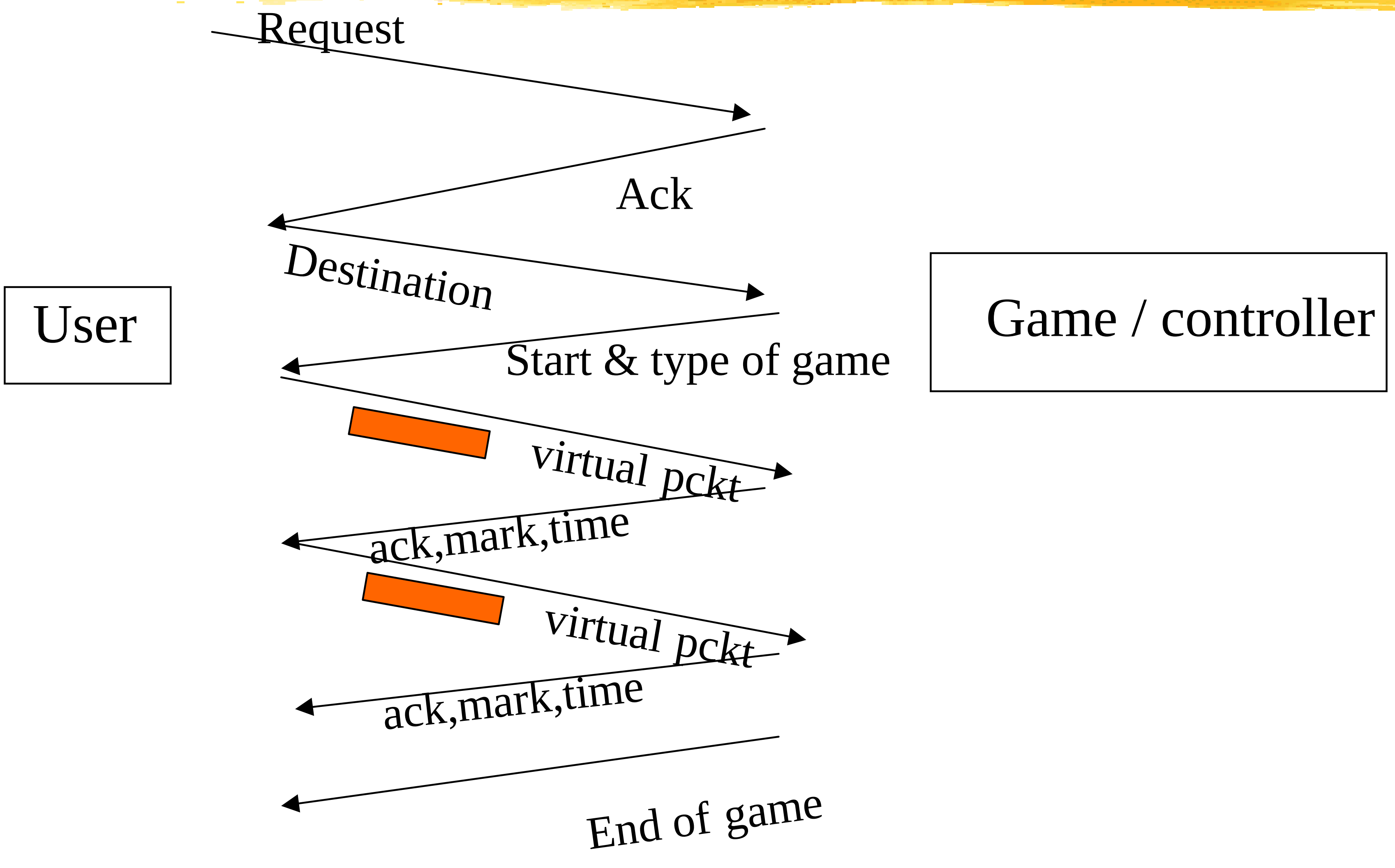
Distributed multi-player game



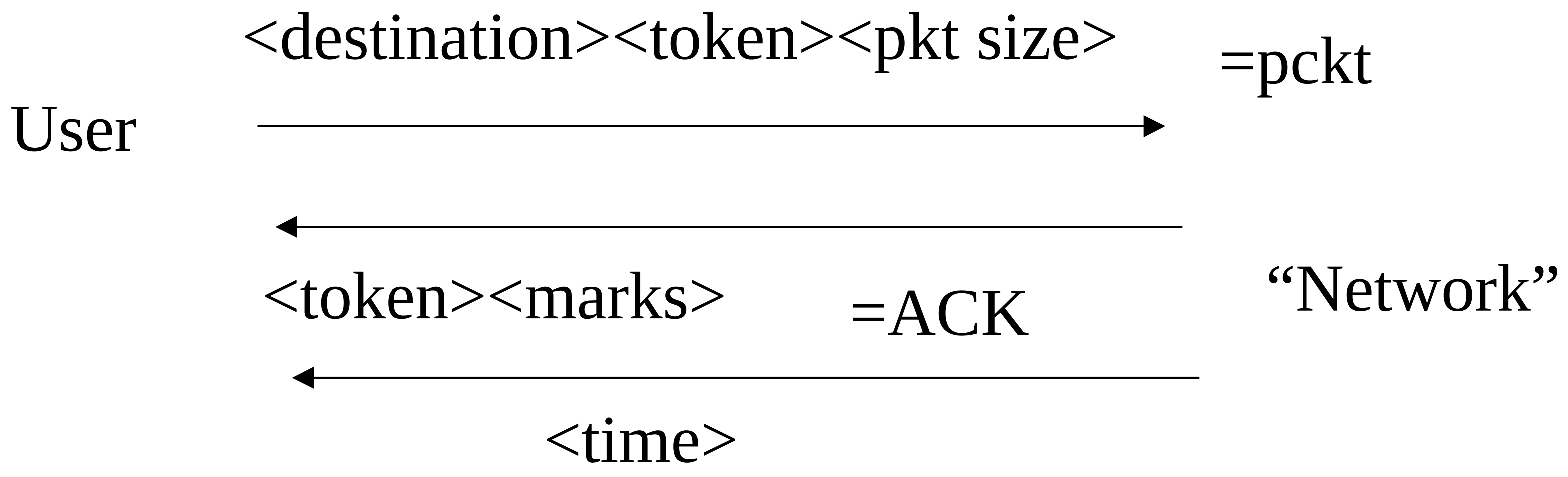
Protocol structures



Information flow



Protocol



Notes:

- *All single word (32bit) unsigned,*
- *pkt size integer*
- *marks integer*
- *Time=sec.μsec*
- *Token generated by user*
- *Corrupted/lost token=packet drop*

Example Objectives

Assumes notified cost per mark

- Maximise (ave. thruput - ave. cost)
- Max Discounted $\sum$(thruput - cost)
- For given utility function, max $\sum$(utility - cost)
- Transfer an amount of data *F* (file) at min cost
- Transfer *F* in set time *T* at min cost
- Transfer *F* as quickly as possible at min cost
- Given fixed budget, maximise transfer

Iterative Approach

- New User plays on test harness
- Plays against controlled load
    - (eg against copies of single game or against sample from random population)
- Plays against other users each with same objective
- Plays against others with multiple objectives

Disciplines

- Computer science
- Control Theory
- Game Theory / econometrics
- Stochastic Decision Theory
- Optimisation / Dynamic Programming

Let the games begin!