Mathematical Modeling of Communication Networks

Urtzi Ayesta
BCAM and Ikerbasque

Outline of course given at:
BCAM – Basque Center for Applied Mathematics

Derio, January 18-22, 2010

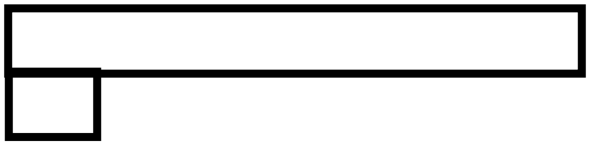
Contents:

- Introduction and motivating examples
- Probability, Random Variables, Stochastic Processes
- Discrete and Continuous time Markov Chains
- Optimal stochastic control: Principle of optimality, Dynamic Programming
- Size-Based Scheduling (FIFO, Shortest Job First, etc.)
- Fluid limits: ODE's as approximations of stochastic processes
- Game Theory: Nash equilibrium, price of anarchy, braess paradox
- The Internet as a decentralized optimization problem (based on Frank Kelly's work)
- Bibliography

Items in RED are presented in the slides

Why to schedule?

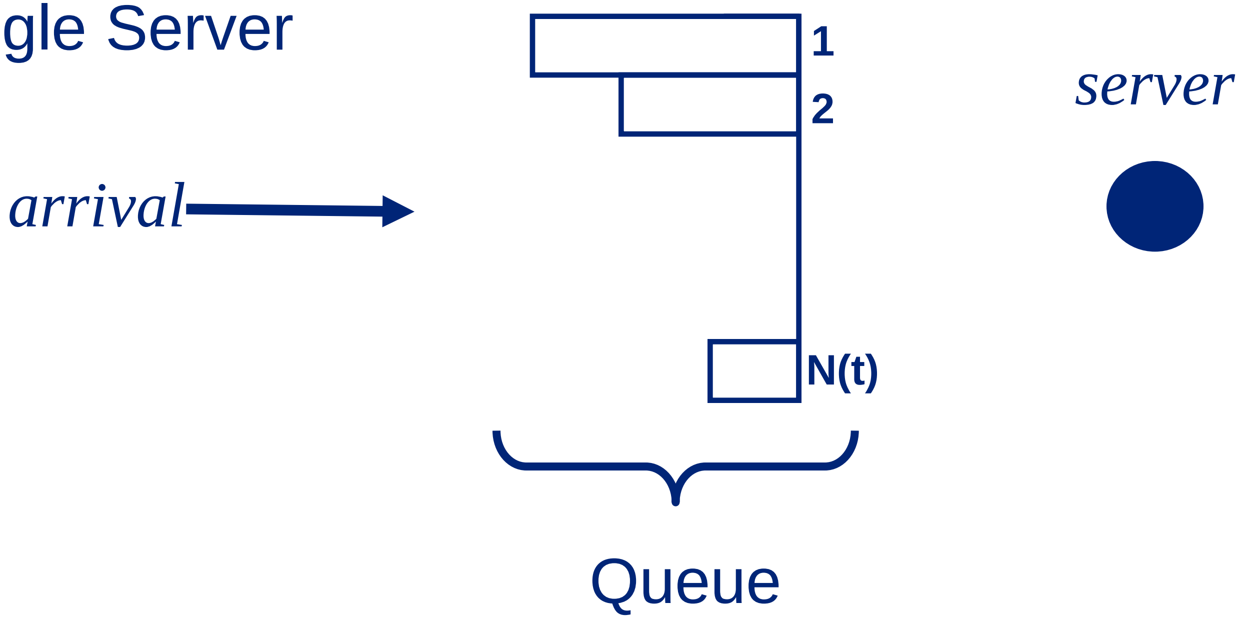
– Simple example. Two jobs available, S_1=10 S_2=1



– Policy 1,2: E[T]=(10+11)/2=10,5
– Policy 2,1: E[T]=(1+11)/2=6

– General conclusion: Giving preference to shorter jobs improves the overall performance.
  - The improvement becomes more significant as the variability of the service times increases.

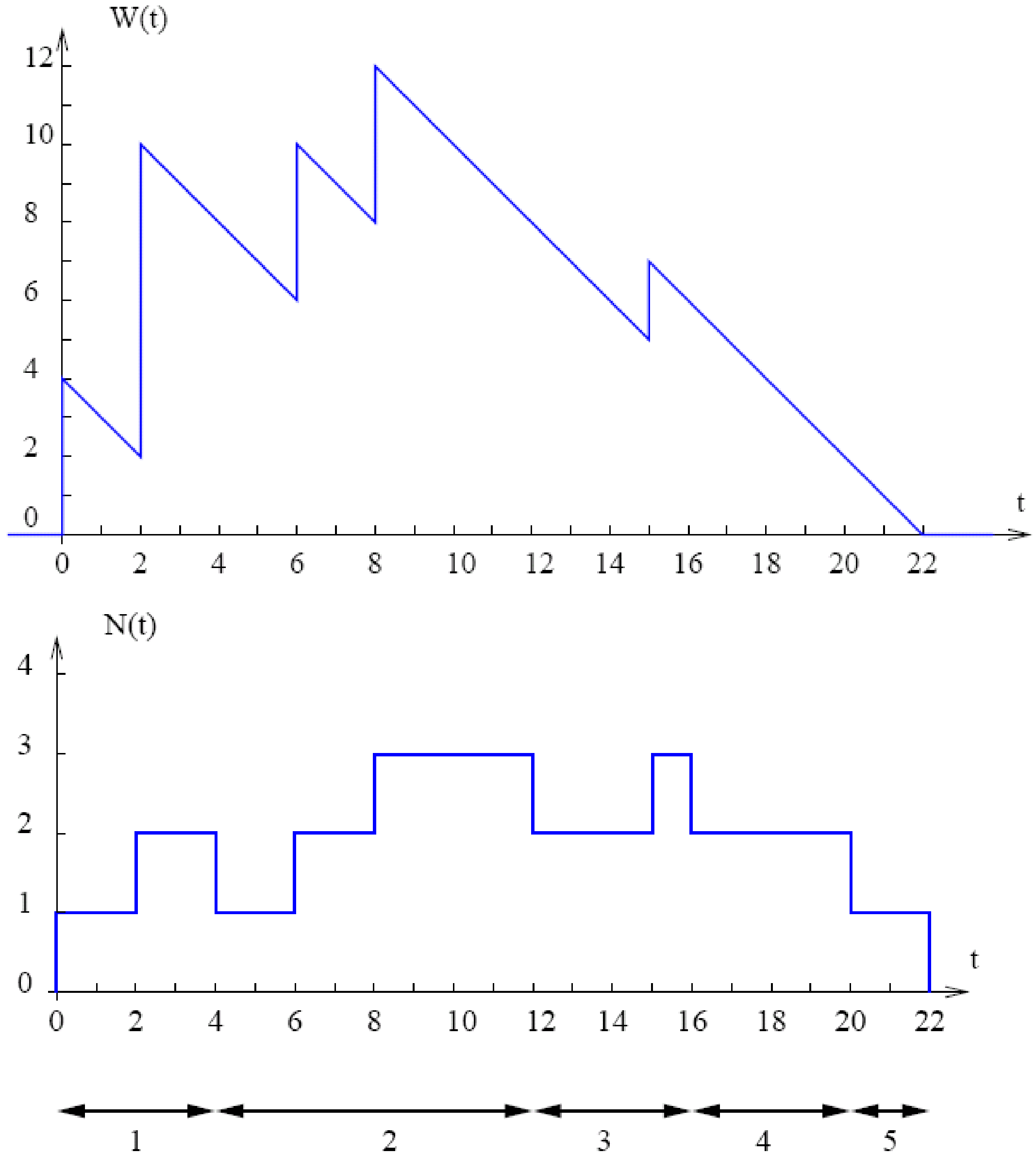
- A Single Server



- Let W(t) denote the total work in the system at time *t*
- Let N(t) denote the number of jobs in the system at time *t*.

- For general arrival process and service times, the value of W(t) at arrival epochs can be interpreted as a one dimensional random walk
- For Poisson arrival process and exponential service times, N(t) is a Markovian birth and death process.

Dynamics of the single server

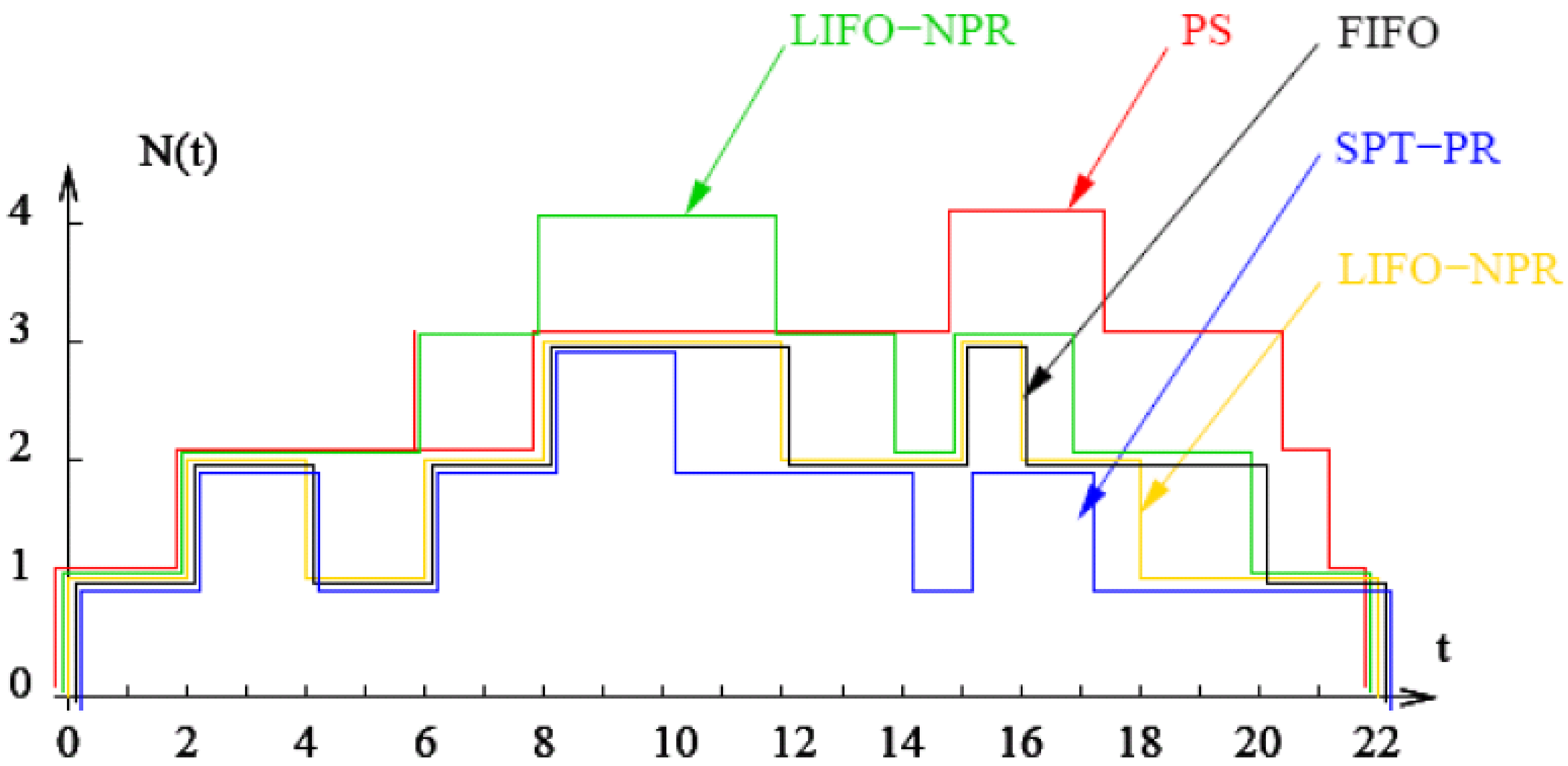


Numéro	Arrivée	Service
1	0	4
2	2	8
3	6	4
4	8	4
5	15	2

– The evolution of W(t) is independent of the scheduling policy

– The evolution of N(t) does depend on the policy

Number of jobs for various policies



Optimization criteria

- Determine the scheduling discipline that minimizes some performance criterion:
  - Single class system:
    - Sample-path: Minimize $N(t)$ for all t
    - Stochastic ordering: Minimize $P[N(t)>k]$ for all t and all k
    - Mean: Minimize $E[N]$, where N denotes the number of jobs in steady-state
  - Multi-class: Minimize $\Sigma_i c_i E[N_i]$
- Ergodic stochastic process: $E[N]$ et $E[T]$ (sojourn time) are proportional (Little's law [Little61])

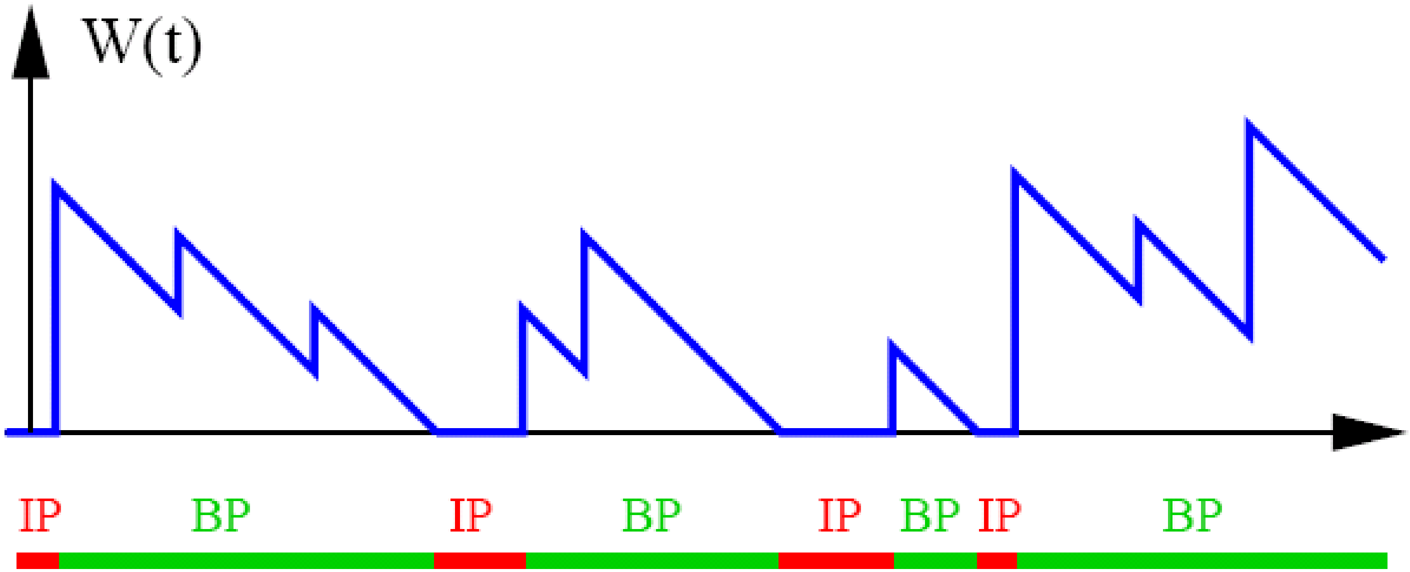
Classification of Scheduling disciplines

- Knowledge:
  - Anticipating
  - Non-anticipating

- Preemption:
  - Preemptive
  - Non-preemptive

Stability

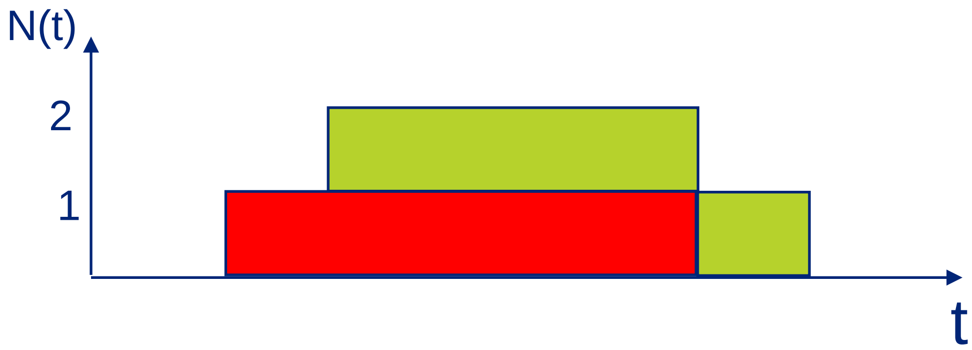
- The total workload in the system is independent of the scheduling discipline



- N(t)=0 if and only if W(t)=0
- The stability condition is independent of the scheduling discipline
- Let $\rho=E[S]/E[\alpha]<1$ denote the load in the system, where $E[\alpha]$ is the mean interarrival time, and $E[S]$ the mean service time

- Theorem: The stochastic process is stable if and only if $\rho<1$ [Lindley52,KW55,Loynes62]

Little's law



$$\int_0^t N(s)ds = T_1 + T_2$$

- In general we have $\frac{1}{t}\int_0^t N(s)ds \approx \frac{1}{t}\sum_{i=1}^{A(t)} T_i = \frac{A(t)}{t}\frac{1}{A(t)}\sum_{i=1}^{A(t)} T_i$
- As t $\longrightarrow$ ∞, A(t) / t = *λ*, where *λ* is the mean arrival rate
- As t $\longrightarrow$ ∞, $\frac{1}{A(t)}\sum^{A(t)} T_i = E[T]$
- Note that $\lim \frac{1}{t}\int_0^t N(s)ds = \mathrm{E[N]}$ so we get $\lambda E[T] = E[N]$

Anticipating disciplines: Optimality of SRPT

- SRPT : Shortest remaining processing time first
- Theorem [Schrage68,Smith78]: SRPT minimizes the number of jobs in the system sample-path wise, that is, for all *t*

$$N^{SRPT}(t) \leq N^{\pi}(t)$$

  for any other admissible scheduling discipline π.
- Pure sample-path result: Arrival process and service time requirements can be arbitrary (any correlation structure is allowed)
- [Sketch of the proof] Order the jobs such that

$$R_1(t) \geq R_2(t) \geq .... \geq R_{N(t)}(t)$$

- By definition of SRPT, for all *t*

$$R_1^{SRPT}(t) \geq R_1^{\pi}(t)$$

Anticipating: Optimality of SRPT (cont.)

- It can be shown that for any *j* and for all *t*

$$\sum_{i=1}^{j} R_i^{SRPT}(t) \geq \sum_{i=1}^{j} R_i^{\pi}(t)$$

But this together with the fact that

$$W^{SRPT}(t) = \sum_{i=1}^{N^{SRPT}(t)} R_i^{SRPT}(t) = \sum_{i=1}^{N^{\pi}(t)} R_i^{\pi}(t) = W^{\pi}(t)$$

implies that $N^{SRPT}(t) \leq N^{\pi}(t)$

Non-anticipating disciplines

- The size is not known, but we know the *attained service* of jobs.
  - The most appropriate scheduling discipline depends on the service time distribution characteristics

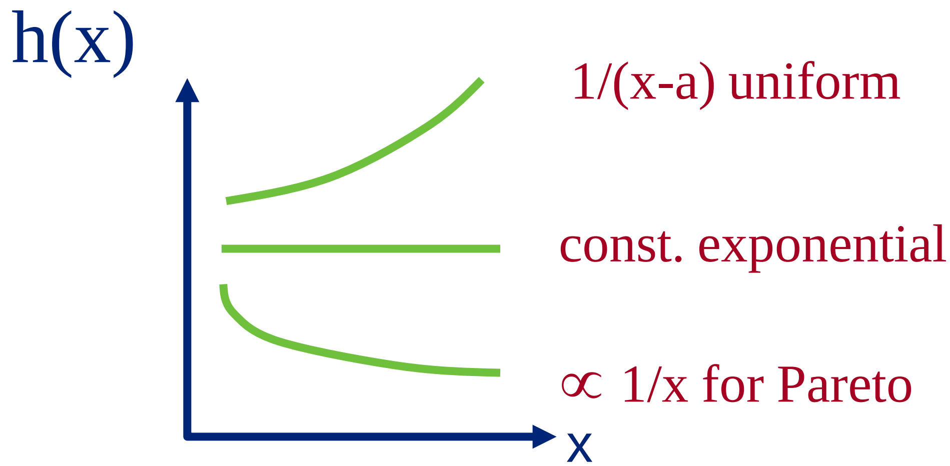
- Let $F(x)$ denote the service time distribution, that is, $F(x)=P[S\leq x]$. And let $f(x)$ denote the density function, that is, $f(x)=dF(x)/dx$.

- Hazard rate of a distribution function:
  $h(x)dx = P[x< S \leq x+dx \mid S > x]$

$$h(x)=\frac{f(x)}{1-F(x)}$$

Non-anticipating discipline (cont.)

Many human related random variables have a decreasing hazard rate: duration of chat/voice conversations, size of downloading files etc.



If the hazard rate is increasing, then the optimal policy is FCFS [RS89]

If the hazard rate is decreasing, then the optimal policy is FB (LAS)

Non-anticipating discipline (cont.)

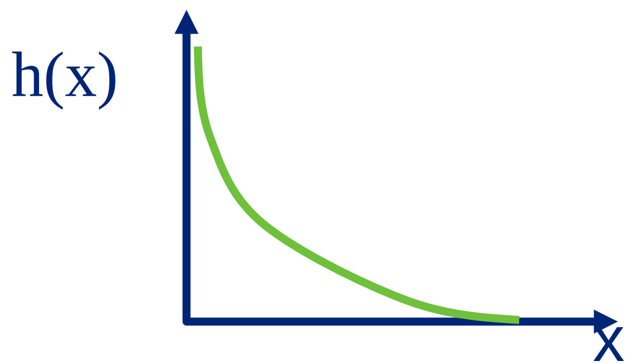
– We denote by FB (Foreground-Background) or LAS (Least Attained Service) the policy that gives service to the job with the least attained service

– Theorem [RS89,Yashkov87] If the hazard rate distribution is decreasing (DFR),

then LAS is stochastically optimal, that is $N^{LAS}(t) \leq_{st.} N^{\pi}(t)$ so

for all *t* and all *k* we have $P[N^{LAS}(t) > k] \leq P[N^{\pi}(t) > k]$

Intuition behind the result:



Proof: By Induction on time horizon and an interchange argument

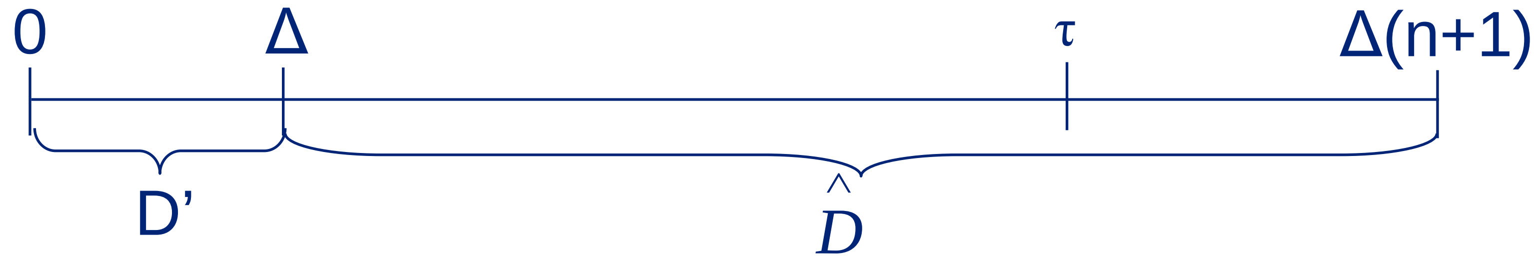
Induction: Let T be the time horizon. Stochastic optimality of LAS holds for T=0. Assume it holds for T=nΔ, and prove it holds for T'=(n+1)Δ.

Contradiction: Suppose that for horizon T', π is optimal but does not follow LAS

- So at time 0, task j is processed under π but there exists a task i such that $s_j > s_i$, so $h(s_j) < h(s_i)$ (h is decreasing)
- After time Δ, the horizon is T=nΔ, so π does LAS.
- Let τ be the first time that π schedules task i
- By the DFR assumption we have $h(s_j + \Delta) \le h(s_j) < h(s_i)$.
- Thus j will not be scheduled again before time τ

Interchange: Construct π' by scheduling task i at time 0, and task j at time τ

Let D' be the number of jobs completed during the first Δ time units and $\hat{D}$ be the number of jobs completed after time Δ



$$P_\pi(D(T')\geq k) = P_\pi(D(T')\geq k \mid T'\geq \tau)P_\pi(T'\geq \tau) + P_\pi\left(\hat{D}\geq k, T'<\tau\right)$$
$$+ P_\pi\left(\hat{D}=k-1, T'<\tau\right)P_\pi(D'=1)$$

After time τ it is irrelevant: $P_\pi(D(T')\geq k \mid T'\geq \tau) = P_{\pi'}(D(T')\geq k \mid T'\geq \tau)$

If T'< τ, $\hat{D}$ under π and π' are the same, so:

$$P_\pi\left(\hat{D}\geq k, T'<\tau\right) = P_{\pi'}\left(\hat{D}\geq k, T'<\tau\right) \text{ and } P_\pi\left(\hat{D}=k-1, T'<\tau\right) = P_{\pi'}\left(\hat{D}=k-1, T'<\tau\right)$$

For all *k*, we end up with

$$P_{\pi'}(D(T')\geq k) - P_\pi(D(T')\geq k) = P_\pi\left(\hat{D}=k-1, T'<\tau\right)\left(P_{\pi'}(D'=1) - P_\pi(D'=1)\right)$$
$$= P_\pi\left(\hat{D}=k-1, T'<\tau\right)\left(h(s_i)-h(s_j)\right) > 0$$

Starvation ?

Size-based disciplines (like SRPT and LAS) reduce the total number of jobs in the system

- Preference is given to small jobs
  - Performance of small jobs improves

- To what extent is the performance of large jobs degraded?
  - Does starvation of large jobs occur?

Compare sized-based policies with a fair policy (PS)

Fair Policy: Processor-Sharing model

- Processor-Sharing (PS):

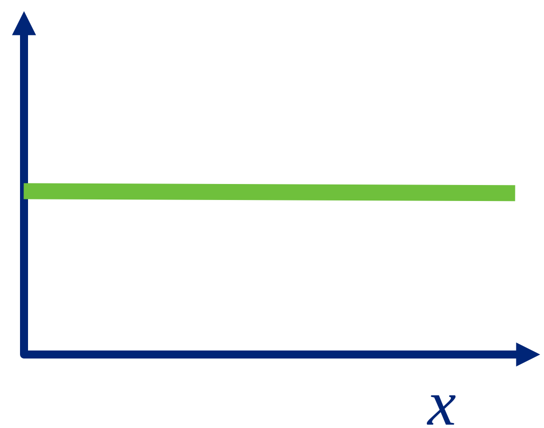
All present jobs in the system get a fair share of service.

If there are N jobs, each job gets served at rate 1/N.

- An acceptable model for (i) data networks at high load (ii) web servers and (iii) CPU

- Well-studied [Kleinrock, Cohen, Kelly, Boxma, Robert …]

$$\frac{E[T \mid S = x]}{x} = \frac{1}{1-\rho}$$



Comparison of size-based and PS

- Theorem [NQ01, HSW02] For FB and SRPT,

$$\lim_{x\to\infty} E[T \mid S = x]/x = \frac{1}{1-\rho}$$

- Theorem [BHB01] If $\rho<1/2$, then for any distribution and for all $x \geq 0$

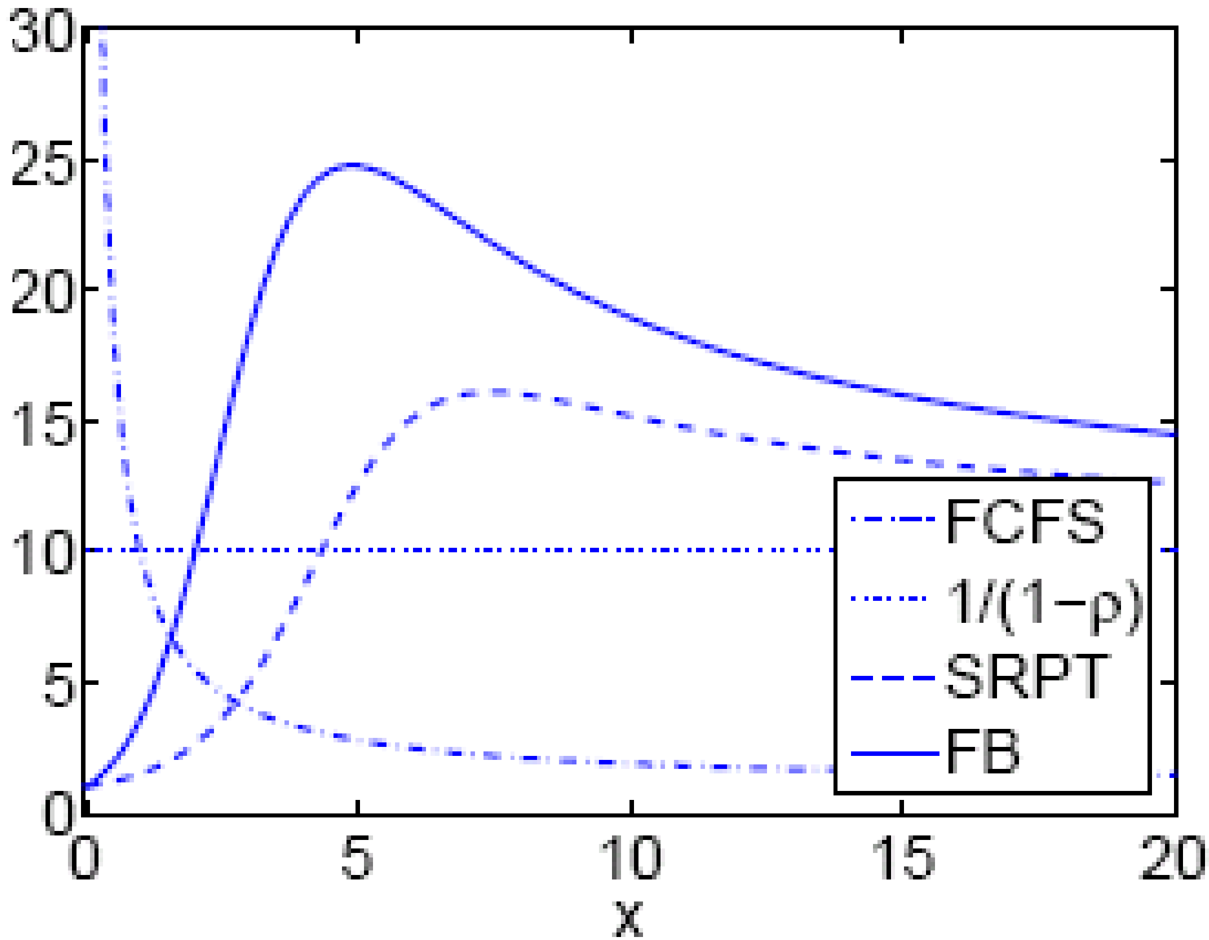
$$E[T^{SRPT} \mid S = x] \leq E[T^{PS} \mid S = x]$$

- Theorem [B06] If $E[S^2]=\infty$, then for all $x \geq 0$

$$E[T^{LAS} \mid S = x] \leq E[T^{PS} \mid S = x]$$

Despite these results, size-based scheduling is rarely implemented.

$$\frac{E[T \mid S = x]}{x}$$



Non-anticipating discipline: What when the Hazard rate is not monotone?

– Theorem [Gittins]: Let the arrival process be Poisson. The Gittins index policy minimizes the mean number of jobs in the system among all non-anticipating scheduling policies, that is

$$E[N^{Gittins}] \leq E[N^{\pi}]$$

for any other admissible scheduling discipline π.

Introduced by Sevcik [1974] for static scheduling.

Optimality with Poisson arrivals due to Gittins [1989].

– Sevcik considered M tasks and showed that the Smallest-Rank policy (equivalent to the Gittins index) minimizes $\sum_{i=1}^{M} E[T_i]$

Gittins index

Job with attained service a has the Gittins index $G(a) = \sup_{\Delta \geq 0} J(a, \Delta)$

with $$J(a,\Delta) = \frac{\int_a^{a+\Delta} f(y)\,dy}{\int_a^{a+\Delta} \overline{F}(y)\,dy} = \frac{\text{reward}}{\text{investment}}$$

- reward: $P[a \leq S \leq a+\Delta \mid S>a]$
- investment: $E[\min(S - a, \Delta) \mid S>a]$

Gittins index policy:

Pick the job with highest index value G(a) and assign him a service quota.

This job will be served until (i) it receives $\Delta^*(a)$ units of service, (ii) it departs from the system or (iii) a new job with higher Gittins index arrives to the queue

Multi-Class single server

We denote for all *i*, $\mu_i = \frac{1}{E[S_i]}$

Theorem:

Consider a single-server multi-class queue with either:

- Non-anticipating discipline and exponential service time distribution
- Non-preemptive discipline and general service time distribution

Then an objective function of the form $\min\left\{\sum_j c_j E[N_j^{\pi}], \pi \in \Pi\right\}$

is optimized by the strict-priority rule $\pi(\varphi)$, where

$$c_{\varphi_1}\mu_{\varphi_1} \geq c_{\varphi_2}\mu_{\varphi_2} \geq \cdots \geq c_{\varphi_M}\mu_{\varphi_M}$$

$\rightarrow$ Known as the cμ-rule. Equivalent to the Smith's rule [Smith56]

Achievable region [Coffman&Mitrani1980]

– Seeks solutions to stochastic optimization problems by
  1. characterize the set of all possible performances (the achievable region)
  2. optimize the overall performance objective over the achievable region
  3. identify the corresponding policy

– Used to address a wide variety of control problems:

multi-class queues (single and multiple server), indexable systems, multi-armed bandits, special classes of queueing networks etc.

$\rightarrow$ Here we focus on multi-class single server systems

Achievable region: Conservation law

– Let $N=\{1,2,\ldots M\}$ denote the set of classes

– Let $W_j^{\pi}(t)$ be the unfinished work at time t of class-j jobs under policy π

and let $W(t) = \sum_{j=1}^{M} W_j^{\pi}(t)$ denote the total work in the system

General Conservation law:

A sample-path argument shows that $W(t)$ is independent of the policy π

Example: Conservation law for non-anticipating discipline and exponential service times

- At every time t $\quad W_j^{\pi}(t) = \sum_{i=1}^{N_j^{\pi}(t)} R_i(t).$
- Memoryless property implies $E(W_j^{\pi}) = E(N_j^{\pi})\dfrac{1}{\mu_j}$
- Summing over all classes: $$E(W) = \sum_{j=1}^{M} \frac{E(N_j^{\pi})}{\mu_j}$$

Conservation law

If Poisson arrivals, then $E(W) = \dfrac{\sum_{j=1}^{M} \rho_j / \mu_j}{1-\rho}$, and the conservation law

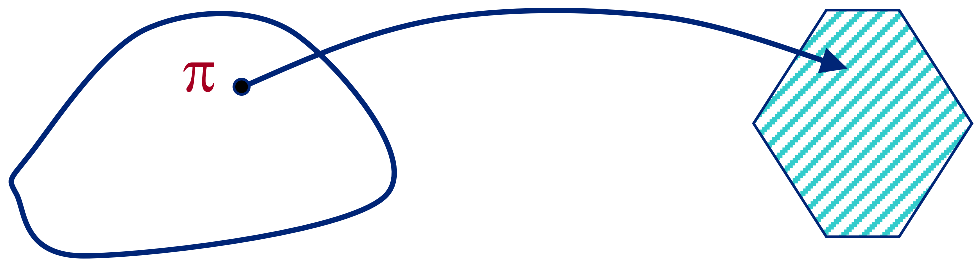
becomes $$\sum_{j=1}^{M} \frac{E(N_j^{\pi})}{\mu_j} = \frac{\sum_{j=1}^{M} \rho_j / \mu_j}{1-\rho}$$

Example (cont.): Achievable region

For any $S \subseteq M$ let $f(S) = \dfrac{\sum_{j \in S} \rho_j / \mu_j}{1 - \rho_S}$

Conservation law gives: $\sum_{j=1}^{M} \dfrac{E(N_j^{\pi})}{\mu_j} = \dfrac{\sum_{j=1}^{M} \rho_j / \mu_j}{1 - \rho}$

So for any admissible policy π: $\sum_{j \in M} \dfrac{E(N_j^{\pi})}{\mu_j} = f(M)$ and $\sum_{j \in S} \dfrac{E(N_j^{\pi})}{\mu_j} \geq f(S), \forall S$



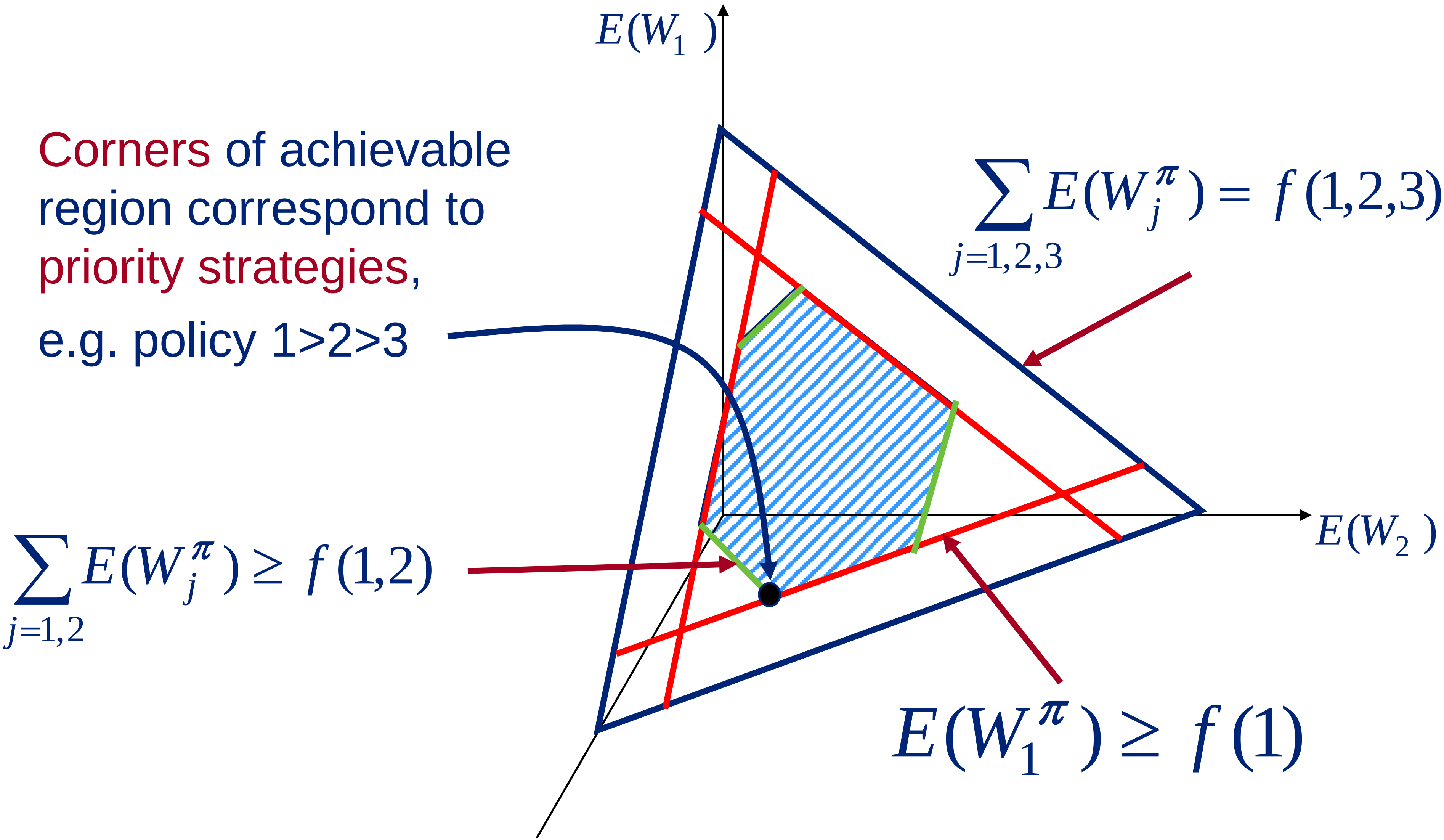
admissible policies

Polyhedron =achievable region

Example: 3 classes

Corners of achievable region correspond to priority strategies,

e.g. policy 1>2>3



In general: Achievable region is a polyhedra of dimension M-1 with M! vertices

Example: 3 classes

Optimality of cμ-rule:

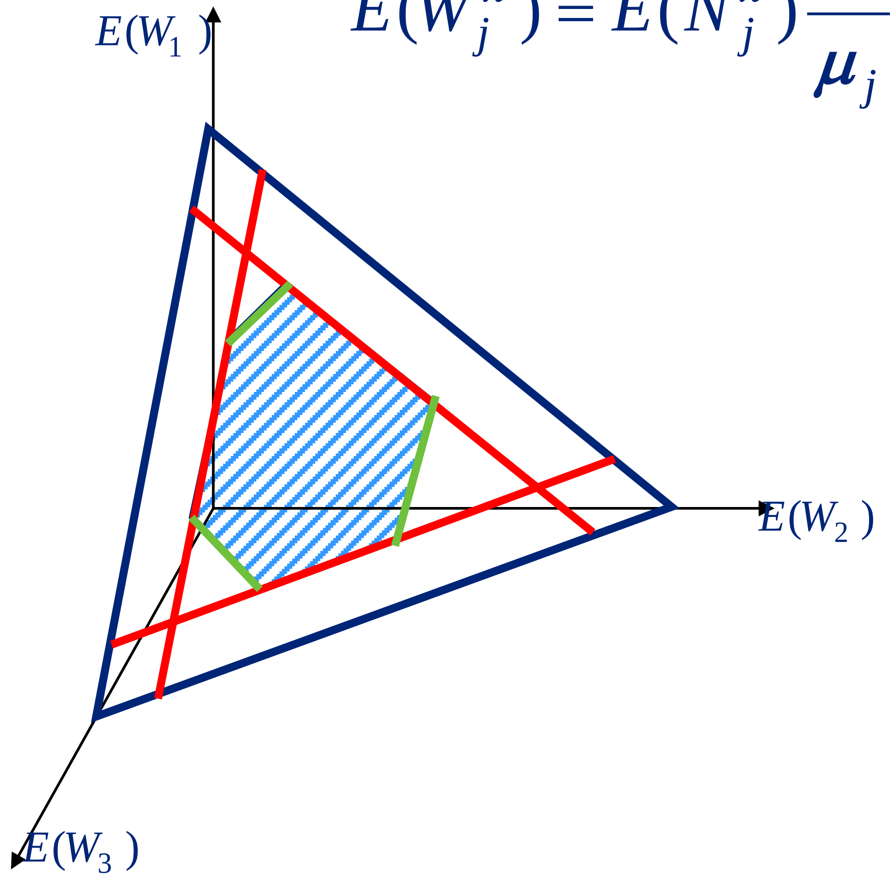
The minimum of the function

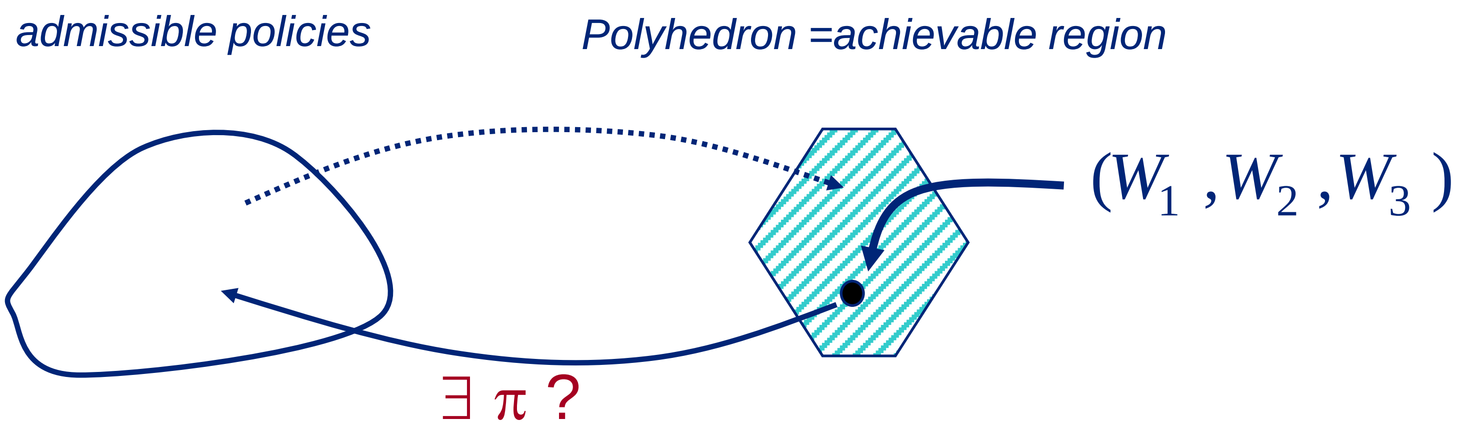
$$\sum_j c_j E[N_j^{\pi}] = \sum_j \mu_j c_j E[W_j^{\pi}]$$

will be achieved in one of the vertices (linear programming argument) of the achievable region and corresponds to the priority policy:

$$c_{\varphi_1}\mu_{\varphi_1} \geq c_{\varphi_2}\mu_{\varphi_2} \geq c_{\varphi_3}\mu_{\varphi_3}$$

$$E(W_j^{\pi}) = E(N_j^{\pi})\frac{1}{\mu_j}$$





Result: for every (W_1, W_2, W_3) in the achievable region, there exists a π such that $(E(W_1^\pi), E(W_2^\pi), E(W_3^\pi)) = (W_1, W_2, W_3)$

Proof based on the convexity of the achievable region:

– A point in the interior can be expressed as a convex combination of the vertices of the achievable region

$$\overrightarrow{W} = \sum_{j \in vertex} \alpha_j \overrightarrow{W}^{\varphi_j}, \text{ with } \sum_{j \in vertex} \alpha_j = 1$$

– Take for π the mixing strategy: At the beginning of the busy period, use priority policy φ_j with probability $\boldsymbol{\alpha_j}$

TABLE 1

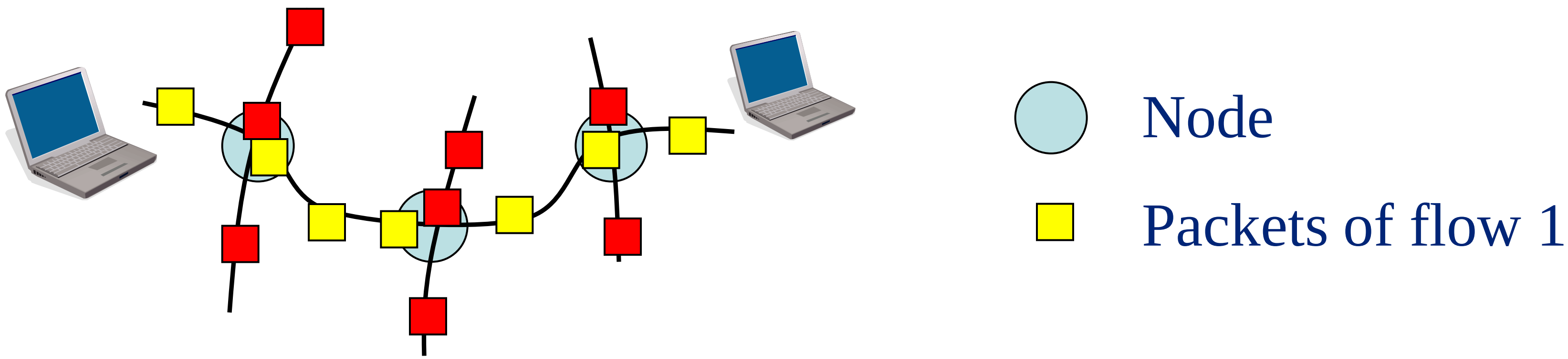
Indexable Problems and their Performance Regions

System	Criterion	Indexability	Performance region
Batch of jobs	LC[a]	Smith (1956): D[b]	Queyranne (1993): D, P[c]
		Rothkopf (1966b)	This paper: P
	DC[d]	Rothkopf (1966a): D	This paper P
		Gittins & Jones (1974)	
Batch of jobs with out-tree	LC	Horn (1972): D	This paper: EP[e]
		Meilijson & Weiss (1977)	
prec. constraints	DC	Glazebrook (1976)	This paper: EP
Multiclass $M/G/1$	LC	Cox & Smith (1961)	Coffman & Mitrani (1980): P
			Gelenbe & Mitrani (1980): P
	DC	Harrison (1975a, 1975b)	This paper: EP
Multiclass[f] $M/G/c$	LC	Federgruen & Groenevelt (1988b)	Federgruen & Groenevelt (1988b)
		Shanthikumar & Yao (1992)	Shanthikumar & Yao (1992): P
Multiclass $G/M/c$	LC	Federguen & Groenevelt (1988a)	Federgruen & Groenevelt (1988a)
		Shanthikumar & Yao (1992)	Shanthikumar & Yao (1992): P
Mutliclass Jackson network[g]	LC	Ross & Yao (1989)	Ross & Yao (1989): P
Multiclass $M/G/1$	LC	Klimov (1974)	Tsoucas (1991): EP
with feedback	DC	Tcha & Pliska (1977)	This paper: EP
Multiarmed bandits	DC	Gittins & Jones (1974)	This paper: EP
Branching bandits	LC	Meilijson & Weiss (1977)	This paper: EP
	DC	Weiss (1988)	This paper: EP

Source: Bertsimas, D. and Niño-Mora, J. (1996). Conservation laws, extended polymatroids and multiarmed bandit problems: a polyhedral approach to indexable systems. Mathematics of Operations Research, vol. 21 no. 2, 257-306.

Bandwidth-Sharing networks

- Traffic flows (web pages, email, music and movies…)
- One flow may consist of many packets

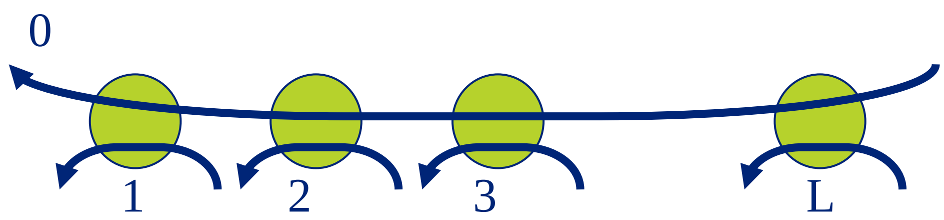


- A flow is served **simultaneously** at the same service rate **at all nodes**
- What is an efficient way to schedule these flows?

Stability is scheduling dependent

- **Class *i*** is stable iff $P(N_i=0) > 0$
- **Network** is stable if all classes are stable

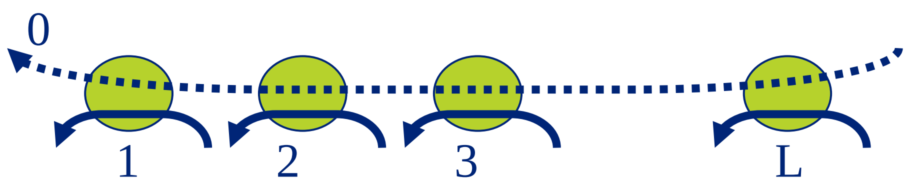
Consider a linear network



→**Necessary** condition **for stability** of network: $\boldsymbol{\rho_0+\rho_i< 1}$ **for all *i***

Stability is scheduling dependent (cont.)

– Prioritize all classes 1,…,*L*



- Class 0 is served only if all classes 1,…,*L* are empty
- Stable iff $\rho_0 < \mathrm{P}(N_1 = 0, \ldots, N_L = 0) = \prod_{i=1}^{L} (1 - \rho_i)$
- More stringent stability condition compared to $\boldsymbol{\rho_0 + \rho_i < 1}$ **for all *i***

– Theorem [VBN05]: In a linear network, size-based scheduling (like SRPT and LAS) may lead to instability at arbitrarily low loads.

α-fair bandwidth-sharing policies

Denote by $s_i(t)$ be the rate given to class *i* flows

α –fair policy: chooses $s_i(t)$ that solves $\max \sum_{i=0}^{L} N_i(t)^{\alpha} \frac{s_i(t)^{\alpha}}{1-\alpha}$

$$s.t.\quad s_0(t) + s_i(t) \leq 1,\ i = 1, \dots, L$$

- **α** = 0: Maximizes throughput: $\max \sum_{i=0}^{L} s_i(t)$
- **α** $\longrightarrow$ 1: Proportional fairness
- **α** = 2: TCP
- **α** $\longrightarrow$ ∞: Max-min fairness

Stability of **α**-fair allocation

The process $\left(\vec{N}(t)\right)_{t\geq 0} = \left(N_1(t),\ldots,N_L(t)\right)_{t\geq 0}$ is Markovian

with transition rates
$$\begin{cases} \left(\vec{N}(t)\right) \to \left(\vec{N}(t)\right) + \vec{e}_i : & \lambda_i \\ \left(\vec{N}(t)\right) \to \left(\vec{N}(t)\right) - \vec{e}_i : & \mu_i s_i(t) \end{cases}$$

Theorem [BM01]: The process $\left(\vec{N}(t)\right)_{t\geq 0}$ corresponding to the **α** –fair policy is stable under the necessary conditions:
$\boldsymbol{\rho_0 + \rho_i < 1}$ **for all *i*.**

→ The result that **α** –fair policies are stable is not restricted to linear network, but holds for general network topology!

MATHEMATICS OF TRAFFIC IN NETWORKS

Internet as an optimization problem

Outline

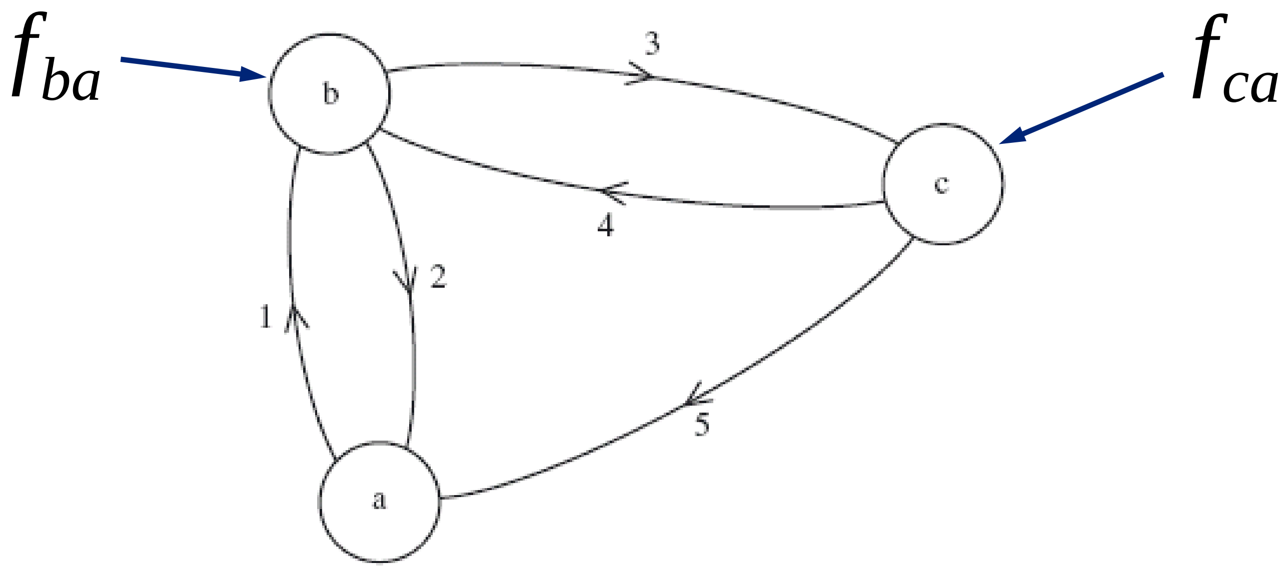
- Network structure

- Queueing/road networks
  - Non-cooperative: Wardrop equilibrium.
    - Potential games
  - Global optimum
  - Braess's paradox

- Data networks
  - Allocation obtained with TCP

- Price of Anarchy: Measuring inefficiency of non-cooperative allocation

Different between routing/road planning and allocation in networks

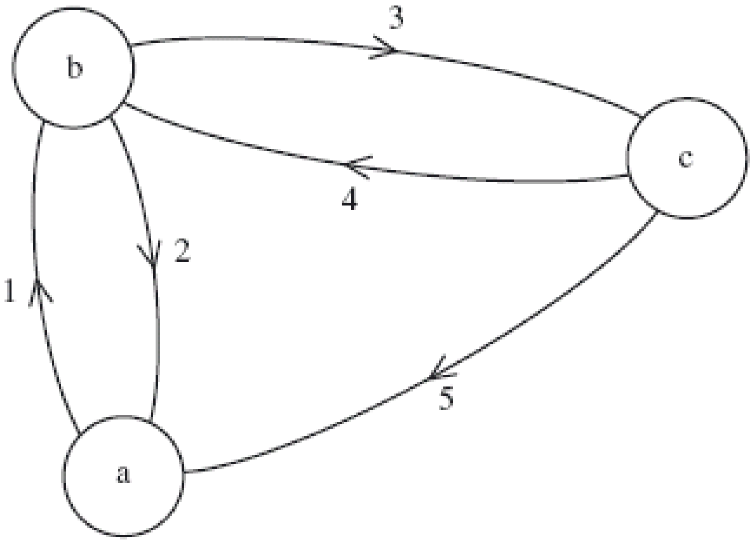


- Routing/road: For fixed source-destination flows what is the optimal way to distribute the total flow among the routes?
- Allocation: Given the capacity constraints, what is the optimal way to share the capacity among the source-destination pairs?

Network structure

- Let J denote the set of links
- Let R denote the set of routes
- Set $A_{jr}=1$ if link j lies on route r. Set $A_{jr}=0$ otherwise.
- This defines a Matrix $A=(A_{jr},\ j \in J,\ r \in R)$ called the link-route incidence matrix

Example



$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccccc} \text{ab} & \text{ac} & \text{ba} & \text{bc} & \text{ca1} & \text{ca2} & \text{cb1} & \text{cb2} \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

- Let x_r be the flow on route r.
- Let $x=(x_r, r \in R)$
- The total flow in a link is

$$y_j = \sum_{r \in R} A_{jr} x_r , \qquad j \in J$$

Delay

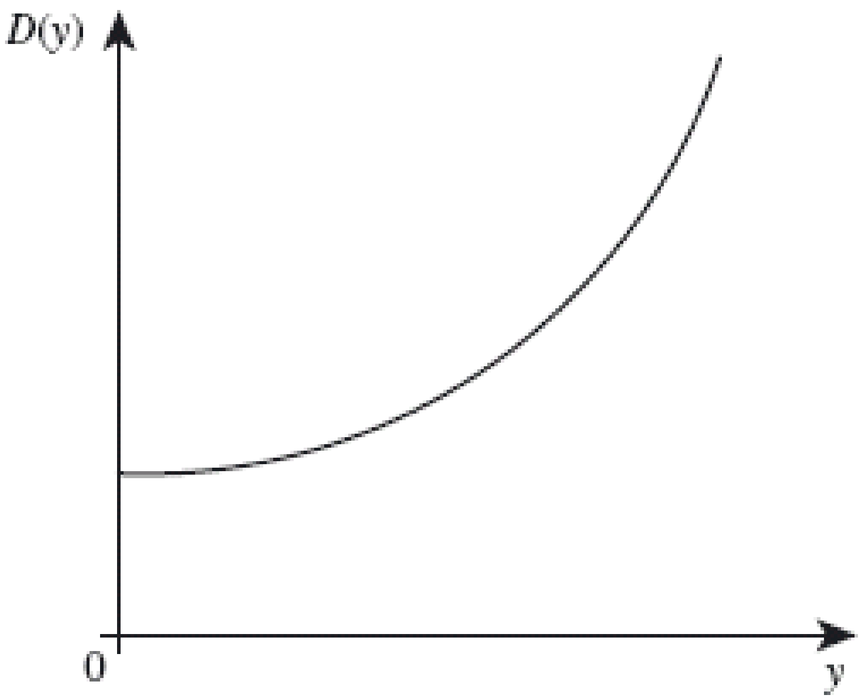
- Let $y=(y_r, r \in R)$
- In Matrix form we can write

$$y = Ax$$

- The level of congestion in a link is a function of the flow. And this will affect the time taken to travel along the link

Delay cont.

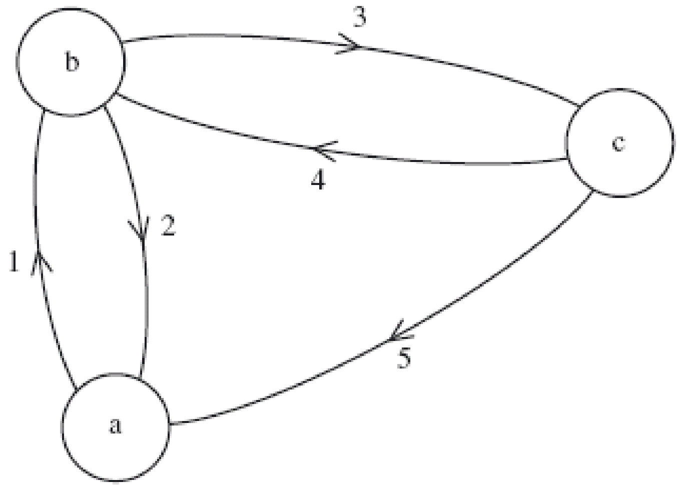
- Let $D_j(y_j)$ be the delay along link j
  - For small values of y_j, $D(y_j)$ is the time to travel along an empty road/link
  - For larger values of y_j, $D(y_j)$ is larger, and possible much larger



Routing choices

- Given 2 nodes, there can be several routes
- Let *s* denote a source-destination pair, and let *S* be the set of all pairs.
- Let $H_{sr}=1$ if *s* can be served by route *r*. Set $H_{sr}=0$ otherwise.
- This defines a Matrix $H=(H_{sr}, s \in S, r \in R)$

Routing choices *cont.*



$$H = \begin{array}{c} \\ ab \\ ac \\ ba \\ bc \\ ca \\ cb \end{array} \begin{array}{c} \begin{array}{cccccccc} ab & ac & ba & bc & ca1 & ca2 & cb1 & cb2 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array}$$

– For each route *r*, let us write *s(r)* the source-destination pair served. For example, *s*(ac)=ac and *s*(ca1)=ca.

– Let f_s denote the total flow of traffic between the source-destination *s*. Then

$$f_s = \sum_{r \in R} H_{sr} x_r, \qquad s \in S$$

– The vector f=(f_s, s∈S) can be expressed as: $f = Hx$

Wardrop Equilibria

- How do the traffic flows between the various sources and destinations distribute themselves over the links of the network?

- Each driver will try to use whatever route is quickest

- Time to travel along route *r* is: $\sum_{j \in J} D_j(y_j) A_{jr}$

- So the driver will be pleased if: $\sum_{j \in J} D_j(y_j) A_{jr} \leq \sum_{j \in J} D_j(y_j) A_{jr'}$

- Define a Wardrop equilibrium as a vector x=(x_r, r$\in$*R*) such that for every pair of route r and r' serving the same source destination

$$x_r > 0 \Rightarrow \sum_{j \in J} D_j(y_j) A_{jr} \leq \sum_{j \in J} D_j(y_j) A_{jr'},$$

where y=Ax

Wardrop equilibrium *(cont.)*

- Does a Wardrop equilibrium exist?
- Consider the optimization problem:

$$\begin{aligned} \text{Minimize} \quad & \sum_{j \in J} \int_0^{y_j} D_j(u)\, \mathrm{d}u \\ \text{over} \quad & x \geqslant 0, \; y, \\ \text{subject to} \quad & Hx = f, \; Ax = y. \end{aligned}$$

- Let us see in outline how the above optimization problem has a solution *(x,y)*, and why, if *(x,y)* is a solution, the vector *x* is a Wardrop equilibrium.

- To find a solution we use the method of Lagrange multipliers

$$L(x, y; \boldsymbol{\lambda}, \boldsymbol{\mu}) = \sum_{j \in J} \int_0^{y_j} D_j(u)du + \boldsymbol{\lambda}(f - Hx) - \boldsymbol{\mu}(y - Ax) - \boldsymbol{\gamma} x$$

- From general results on convex optimization it is known that there exist Lagrange multipliers (*λ, μ,γ*) and a vector (*x, y*) such that (*x, y*) minimizes *L*(*x, y*; *λ, μ, γ*), satisfies the constraints *Hx* = *f*, *Ax* = *y*, and solves the original optimization problem.
- A minimum of L(x,y;**λ,μ**) over all (x,y) occurs when

$$\boldsymbol{\mu}_j = D_j(y_j)$$

$$\lambda_{s(r)} = \sum_{j \in J} \mu_j A_{jr} \quad \text{if } x_r > 0$$

$$\leqslant \sum_{j \in J} \mu_j A_{jr} \quad \text{if } x_r = 0.$$

- Thus if traffic in the network distributes itself in accordance with the self-interested choices of drivers, the equilibrium flows (x, y) will solve an optimization problem.

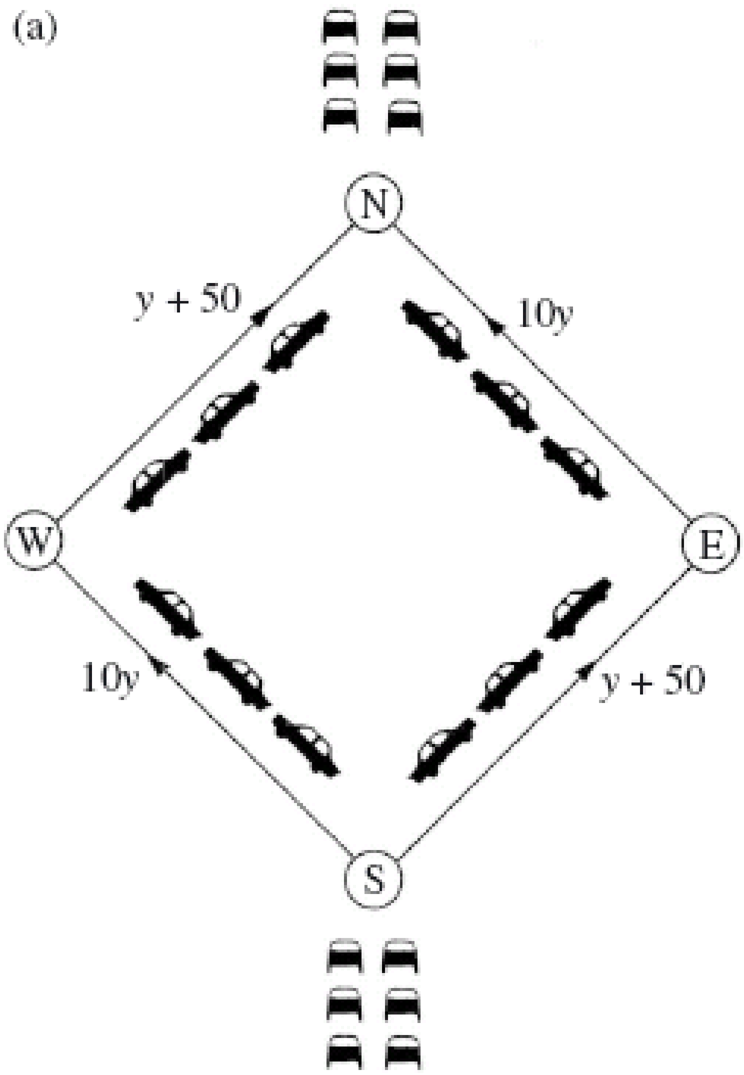
- Result originally due to [Beckmann,1956]

- The pattern of traffic resulting from the individual decisions of a large number of self-interested drivers behaves as if a central intelligence were directing flows to optimize a certain (rather strange) objective function.

- Games whose outcome solve an optimization problem are called Potential games.
  - Existence of a Potential function is a sufficient condition such that the dynamics of greedy users converges to equilibrium [Rosenthal,1973]

Braess's Paradox

- Total flow from S to N is 6
- Wardrop equilibrium 3 units travel via W, and 3 via E
- Total Delay (10x3)+(3+50)=83

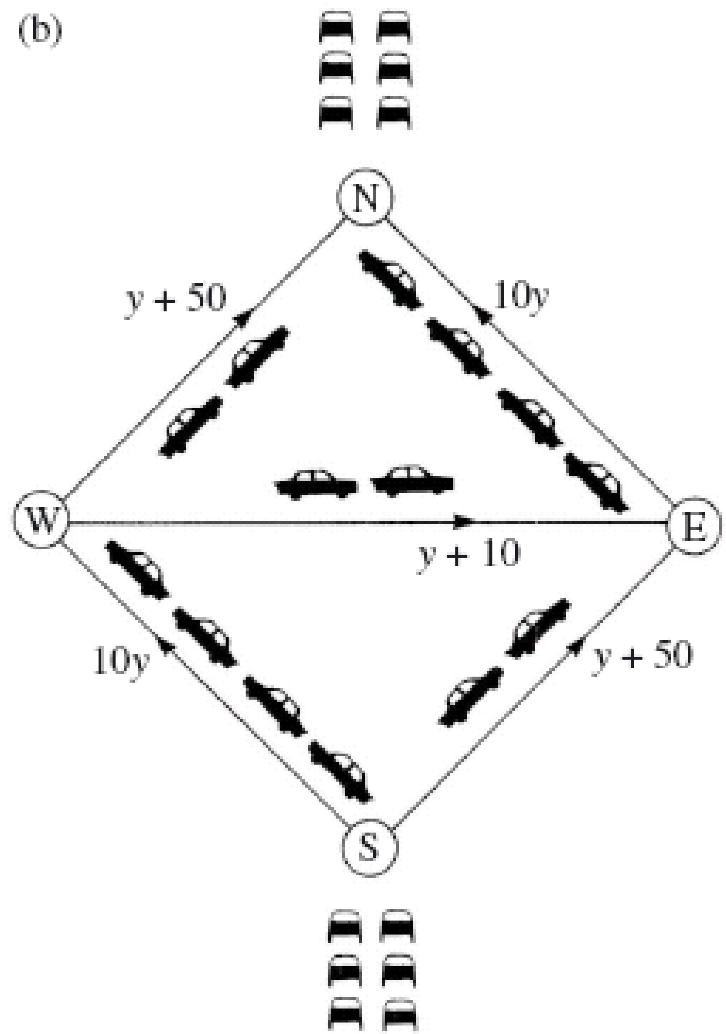


Braess's Paradox *(cont.)*

- A new link is added.
- After everyone knows about the new link, a new Wardrop equilibrium will be established.
- In the new equilibrium, the 3 links are used.
- There are 3 possible routes with the same delay:

  (10x4)+(2+50)=92

  (10x4)+(2+10)+(10x4)=92
- Adding a new link increases everyone's delay!



Braess's paradox *(cont.)*

- Similar paradoxes have been found in Electrical circuits, network with springs and strings etc.

- In real life:
  - In Seoul, South Korea, a speeding-up in traffic around the city was seen when a motorway was removed as part of the Cheonggyecheon restoration project.
  - Stuttgart, Germany after investments into the road network in 1969, the traffic situation did not improve until a section of newly-built road was closed for traffic again.
  - In 1990 the closing of 42nd street in New York City reduced the amount of congestion in the area.

Optimal traffic pattern

- The product $y_j D(y_j)$ denotes the delay incurred at link j per unit of time

- The optimal traffic pattern minimizes the total delay per unit of time:

$$\begin{aligned} \text{Minimize} \quad & \sum_{j \in J} y_j D_j(y_j) \\ \text{over} \quad & x \geqslant 0, \; y, \\ \text{subject to} \quad & Hx = f, \; Ax = y. \end{aligned}$$

- The problem is of the same form as in the Wardrop equilibrium

Optimal traffic pattern *(cont.)*

- To find a solution we use the method of Lagrange multipliers

$$L(x, y; \boldsymbol{\lambda}, \boldsymbol{\mu}) = \sum_{j \in J} y_j D_j(y_j) + \boldsymbol{\lambda}(f - Hx) - \boldsymbol{\mu}(y - Ax)$$

- A minimum of L(x,y;λ,μ) over all (x,y) occurs when

$$\boldsymbol{\mu}_j = D_j(y_j) + y_j D_j^{'}(y_j)$$

$$\begin{aligned} \lambda_{s(r)} &= \sum_{j \in J} \mu_j A_{jr} \quad \text{if } x_r > 0 \\ &\leqslant \sum_{j \in J} \mu_j A_{jr} \quad \text{if } x_r = 0. \end{aligned}$$

Optimal traffic pattern *(cont.)*

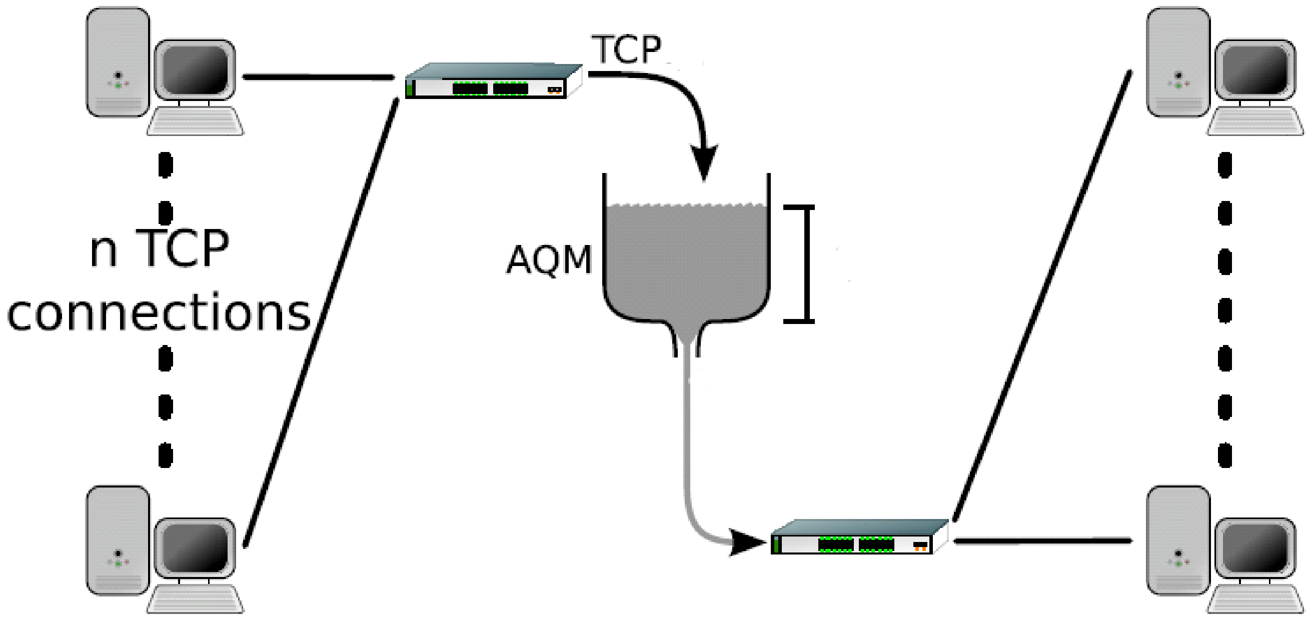
- Suppose that in addition to the delay $D(y_j)$, users of link *l* incur a traffic-dependent toll

$$T_j(y_j) = y_j D_j^{'}(y_j)$$

- Then μ_j is the generalized cost of using link *j* and λ_s is the minimum generalized cost over all routes serving the source-destination *s*
- With the imposition of appropriate tolls, it is possible for the self-interested behavior of drivers to lead to an equilibrium pattern of flow that minimizes total delay
  - Application: Downtown London
- With selfish behavior, it is not guaranteed that adding a road improves the situation

Flow control in the Internet

- *n* TCP connections sharing a bottleneck



- TCP:
  - Distributed Congestion Control algorithm
  - The TCP sender detects congestion on the network through the feedback received from the TCP receiver

Mean throughput of TCP

The system problem

$$\begin{array}{ll} \text{maximize} & \sum_{r \in R} U_r(x_r) \\ \text{subject to} & Ax \leq C \\ \text{over} & x \geq 0. \end{array}$$

The user problem

$$\begin{array}{ll} \text{maximize} & U_r\left(\dfrac{w_r}{\lambda_r}\right) - w_r \\ \text{over} & w_r \geq 0. \end{array}$$

User *r* chooses an amount to pay per unit time w_r and receives in return a flow $x_r = w_r / \lambda_r$

The network problem

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{r \in R} w_r \log x_r \\ \text{subject to} & Ax \leq C \\ \text{over} & x \geq 0. \end{array}$$

As if the network maximizes a logarithmic utility function but with constants (w_r, $r \in R$)

Decomposition theorem

Theorem (problem decomposition)

There exist vectors $\lambda = (\lambda_r, r \in R)$,
$w = (w_r, r \in R)$ and $x = (x_r, r \in R)$ such that

(i) $w_r = \lambda_r x_r$ for $r \in R$

(ii) w_r solves $\mathrm{USER}_r(U_r; \lambda_r)$

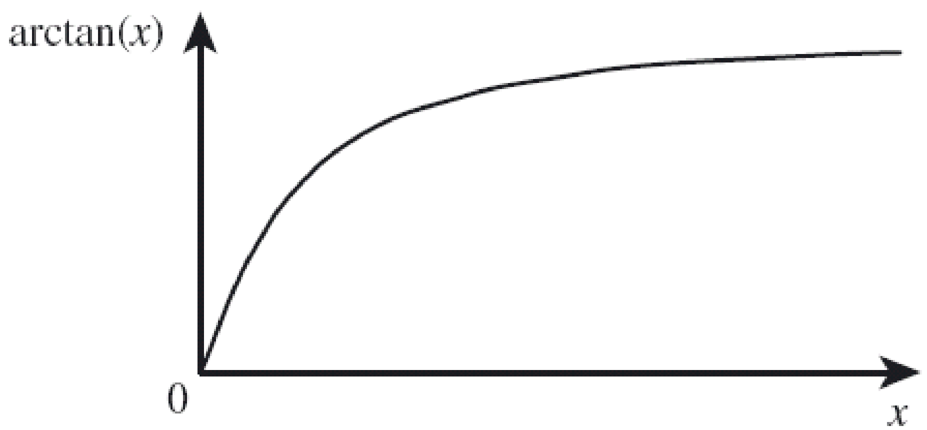
(iii) x solves $\mathrm{NETWORK}(A, C; w)$.

The vector x then also solves $\mathrm{SYSTEM}(U, A, C)$.

- Analyzing the dynamics of real protocols, the utility function is:

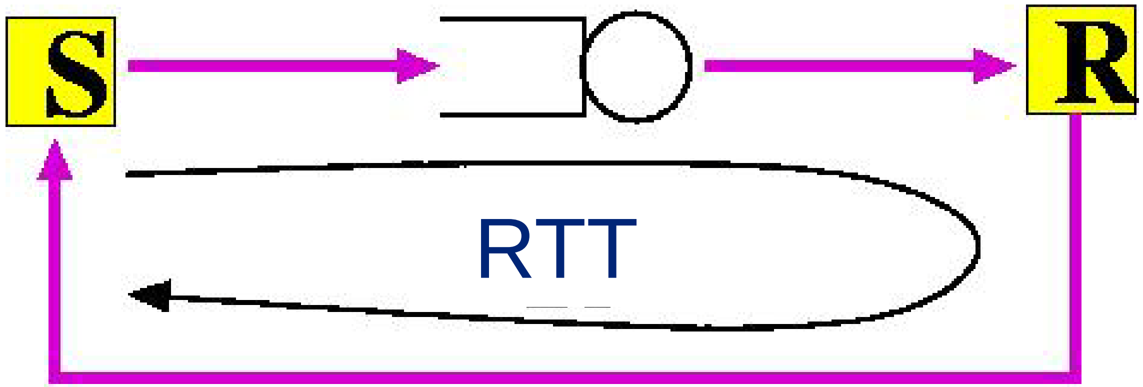
$$\begin{aligned} \text{Maximize} \quad & \sum_{r \in R} \frac{\sqrt{2}}{T_r} \arctan\left(\frac{x_r T_r}{\sqrt{2}}\right) \\ \text{over} \quad & x \geqslant 0, \\ \text{subject to} \quad & Ax \leqslant C. \end{aligned}$$

- Concave function: if two of more connections share an overloaded link, the rates achieved will be approximately the same



TCP

- The packet sending rate (congestion window) is adapted dynamically
  - Increase in absence of packet losses
  - Decrease upon reception of congestion notification



- Current TCP Implementations:
  - AIMD paradigm : The increase is linear in time
  - Problem in fully utilizing the link capacity in the presence of High-Speed links

Mean throughput of TCP

- Let p be the probability that a packet is lost in the network
- Probability (1-p) congestion window increases linearly
- Probability p congestion window is halved.
- Expected change is $cwnd^{-1}(1-p)-\frac{p}{2}cwnd$
- Thus $cwnd=\sqrt{\frac{2(1-p)}{p}}$

- Let p_j be the proportion of packets lost in link *j*

- Assume that if y_j=C_j then p_j . If y_j<C_j then p_j=0.

- Approximately, the probability that a packet is lost on route *r* is

$$p_r = \sum_{j \in J} p_j A_{jr}$$

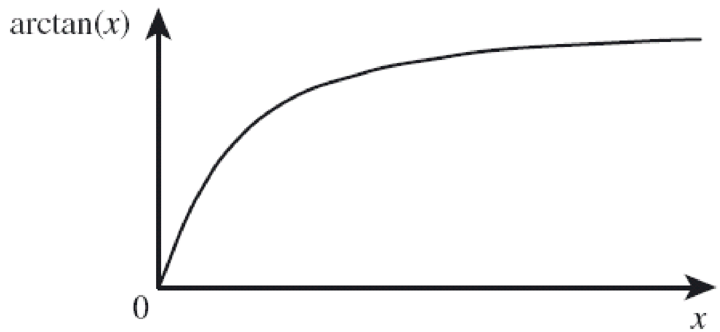
- The throughput on route *r* is: $x_r = \frac{cwnd}{T_r} = \frac{1}{T_r}\sqrt{\frac{2(1-p)}{p}}$

- Is it possible to choose rates $x=(x_r, r \in R)$ and drop probabilities $p=(p_r, j \in J)$ in a consistent fashion?

- The answer is yes, and is given by the solution of:

$$\begin{aligned} \text{Maximize} \quad & \sum_{r \in R} \frac{\sqrt{2}}{T_r} \arctan\left(\frac{x_r T_r}{\sqrt{2}}\right) \\ \text{over} \quad & x \geqslant 0, \\ \text{subject to} \quad & Ax \leqslant C. \end{aligned}$$

- Concave function: if two of more connections share an overloaded link, the rates achieved will be approximately the same



Bibliography

Stability

– J.Kiefer and J.Wolfowitz, On the Theory of queues with many servers, Transactions of the American Mathematical Society, 78, 1—18, 1955
– R. Loynes. The stability of a queue with non-independent inter-arrival and service times. *Proc. Cambridge Phil. Soc.*, 58:497-520, 1962.
– D.V. Lindley. The theory of queues with a single server. *Proc. Cambridge Phil. Soc.*, 48:277-289,1952.

Size based scheduling:

– L.E. Schrage and L.W. Miller, The Queue {M/G/1} with the Shortest Remaining Processing Time Discipline, Operations Research, 14, 670—684, 1966.
– D.R. Smith. A new proof of the optimality of the shortest remaining processing time discipline. *Operations Research*, 26:197-199, 1978.
– R. Righter and J.G. Shanthikumar, Scheduling multiclass single server queueing systems to stochastically maximize the number of successful departures, Probability in the Engineering and Informational Sciences, 3, 323-334, 1989
– S.F. Yashkov. Processor sharing queues: Some progress in analysis. *Queueing Systems*, 2:1-17, 1987.

Stochastic comparison

– W.A. Massey. Stochastic orderings for markov processes on partially ordered spaces. *Mathematics of Operations Research*, 12:350{367, 1987.
– F.J. Lopez and G. Sanz. Markovian couplings staying in arbitrary subsets of the state space. *Journal Applied Probability*, 39:197{212, 2002.

cμ-rule

– W.E. Smith. Various optimizers for single stage production. *Naval Res. Logist. Quart.*, 3:59-66, 1956.

Interchange arguments

– C. Buyukkoc, P. Varaya, and J. Walrand. The $c\mu$ rule revisited. *Adv. Appl. Prob.*, 17:237-238,1985.
– Z. Liu, P. Nain, and D. Towsley. Sample path methods in the control of queues. *Queueing Systems*, 21,1995.
– P. Nain, P. Tsoucas, and J. Walrand. Interchange arguments in stochastic scheduling. *Journal of Applied Probability*, 26(4):815-826, 1989.

Gittins index

– J.C. Gittins. *Multi-armed Bandit Allocation Indices*. Wiley, Chichester, 1989.
– K. Sevcik. Scheduling for minimum total loss using service time distributions. *Journal of the ACM*, 21:66{75, 1974
– I. Meilijson and G. Weiss. Multiple feedback at a single server station. *Stochastic Processes and Applications*, 5:195-205, 1977.

Asymptotic/Starvation

– M. Harchol-Balter and K. Sigman and A. Wierman, Asymptotic convergence of Scheduling policies with respect to slowdown, Performance Evaluation, 49, 241—256, 2002
– N. Bansal and M. Harchol-Balter. Analysis of SRPT scheduling: investigating unfairness. *ACM SIGMETRICS Perform. Eval. Rev.*, 29(1):279-290, 2001.
– R. Nunez-Queija, Processor-Sharing Models for Integrated-Services Networks. PhD Thesis, Eindhoven University of Technology, January 2000, ISBN 90-646-4667-8
– P. Brown. Comparing FB and PS scheduling policies. *SIGMETRICS Perform. Eval. Rev.*,34(3):18-20, 2006.

Scheduling in wireless

– S.C. Borst. User-level performance of channel-aware scheduling algorithms in wireless data networks. In *Proceedings of IEEE INFOCOM*, September 2003.

Achievable region and conservation laws

– E.G. Coffman and I. Mitrani. A characterization of waiting time performance realizable by single-server queues. *Operations Research*, 3(28):810--821, 1980.

– E. Gelenbe and I. Mitrani. *Analysis and Synthesis of Computer Systems*. London: Academic Press, 1980.

– J. Nino-Mora. Restless bandits, partial conservation laws and indexability. *Adv. Appl. Prob.*, 33(1):76-98, 2001.

– U. Ayesta. A unifying conservation law for single server queues. *Journal of Applied Probability*, 44(4):1078{1087, 2007

Stability in networks

– J.G. Dai. On positive harris recurrence of multiclass queueing networks: a unified approach via fluid limit models. *Annals of Applied Probability*, 5:49-77, 1995.

– T. Bonald and L. Massoulié. Impact of fairness on Internet performance. In *Proceedings of ACM SIGMETRICS/Performance*, pages 82-91, 2001.

– I.M. Verloop, S.C. Borst, and R. Nunez-Queija. Stability of size-based scheduling disciplines in resource-sharing networks. *Performance Evaluation*, 62:247-262, 2005.

Textbooks

- D.P. Heyman and M.J. Sobel. *Stochastic Models in operations research, Volume I: Stochastic Processes and operating characteristics*. McGraw-Hill, 1982.
- L. Kleinrock. *Queueing Systems, vol. 2*. John Wiley and Sons, 1976.
- R.W. Wolff. *Stochastic Modeling and the theory of Queues*. Prentice-Hall, 1989.
- M. El-Taha and S. Stidham. *Sample-path analysis of queueing systems*. Kluwer Academic Publishers, 1999.
- F. Baccelli and P. Bremaud, Elements of Queuing Theory: Palm Martingale Calculus and Stochastic Recurrences, Springer 2003