Networks: Design, Analysis and Optimization

Urtzi Ayesta

Séminaire MOCOSY, 27 March 2009

Research Domain

- Scheduling Theory
- Queueing Theory
- Stochastic optimal control
- Game Theory
- and their application to the performance evaluation, conception and dimensioning of communication networks and distributed systems.

Outline of the talk

- **Introduction to Stochastic Processes**

- **Three examples of on-going research**

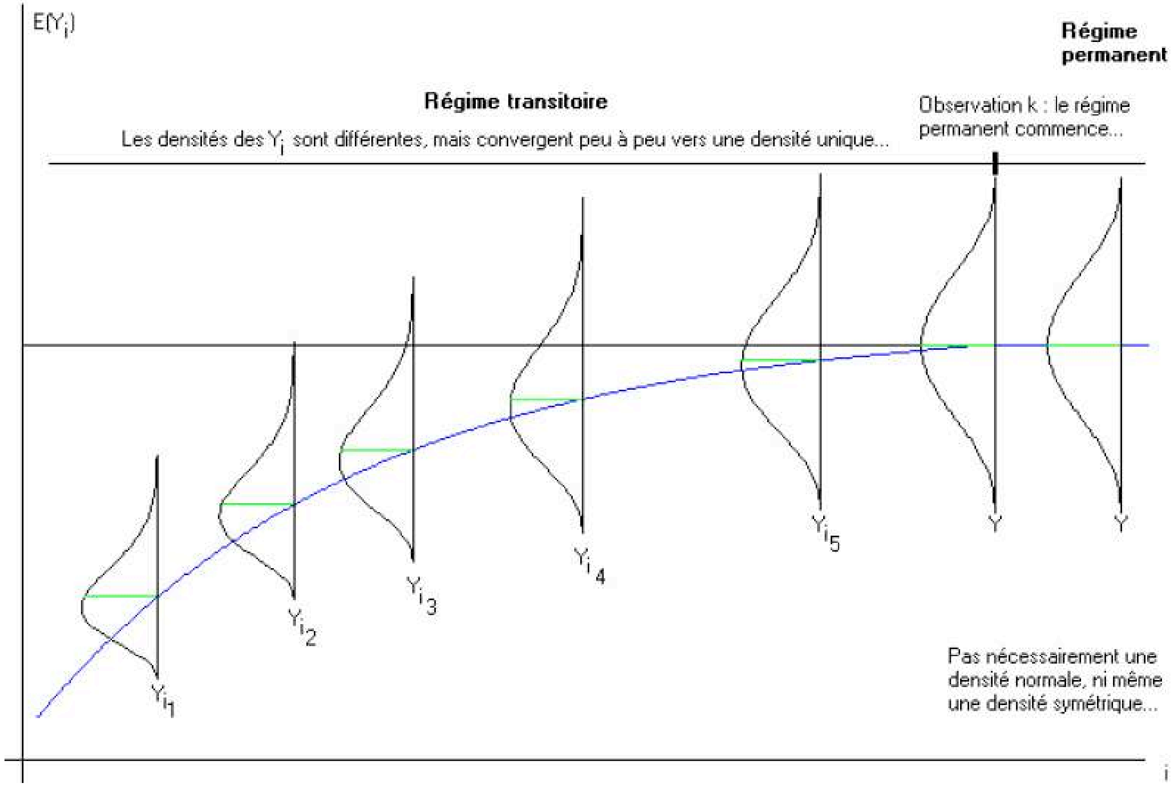
Stochastic Process

- A stochastic process $(N(t))_{t\geq 0}$ is a sequence of random variables indexed by t.

- Randomly evolving dynamical system

- Characterization by first order statistics
  - distribution $\mathbb{P}(N(t) \leq y)$ as a function of t
  - mean $\mathbb{E}[N(t)]$ and variance $\mathbb{E}[N(t)^2]$
  - Simulation, Analysis, Comparison, Optimization, Control...?

Transient vs. Steady-State

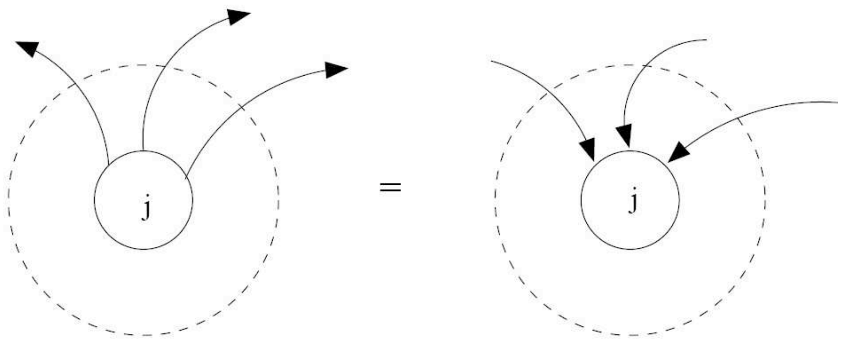


Let $N = \lim_{t\to\infty} N(t)$ denote number of customers in **steady-state**.

Simplest random-walk: $N \to N+1$ at rate λ and $N \to N-1$ at rate μ. Then $\mathbb{P}(N = n) = (\lambda/\mu)^n(1 - \lambda/\mu)$.

Analysis of Steady-State

- Let $\pi_j = \lim_{t\to\infty} \mathbb{P}(N(t) = j)$ denote the steady-state probability
- The number of times the **process departs** from state j is equal to the number of times the process arrives to this state.

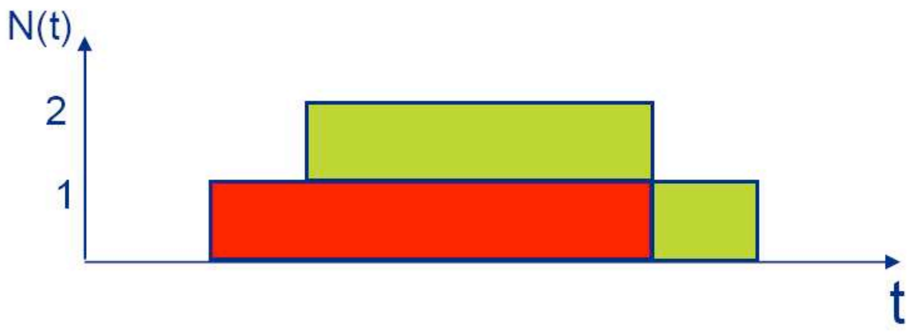


- In equilibrium it holds $\pi_j = \sum_i \pi_i p_{ij}$
- **Questions:** Existence, uniqueness, closed-form, numerical resolution

Little's law: Relation between mean number of jobs and Mean response time

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

Little's law: Relation between mean number of jobs and Mean response time



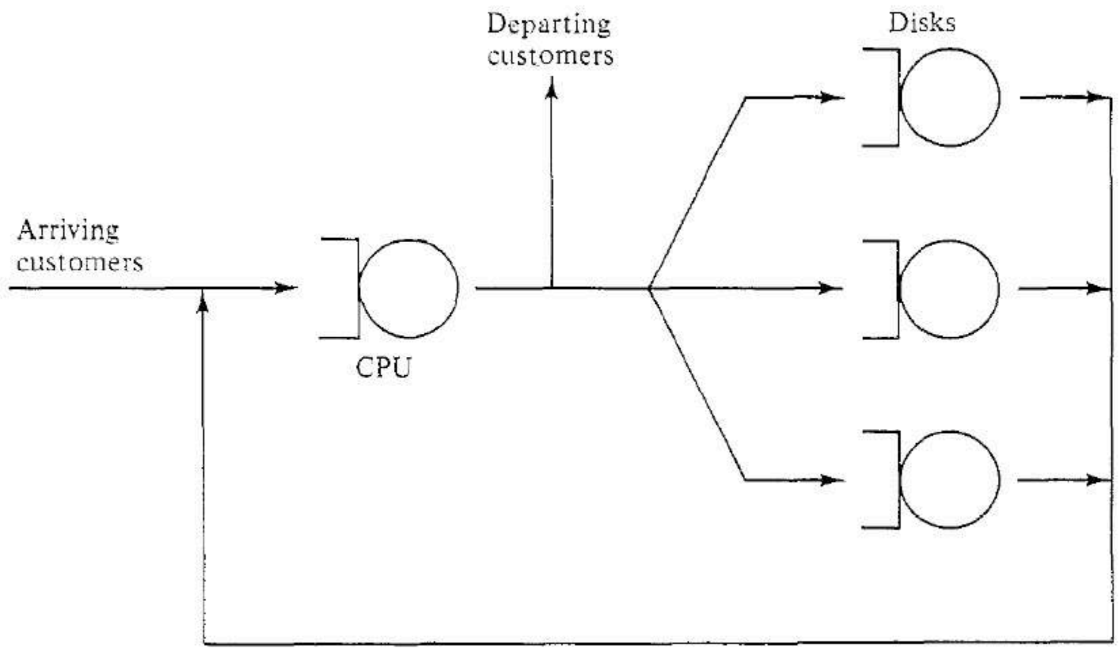
$\int_0^t N(s)ds = T_1 + T_2$

In general we have

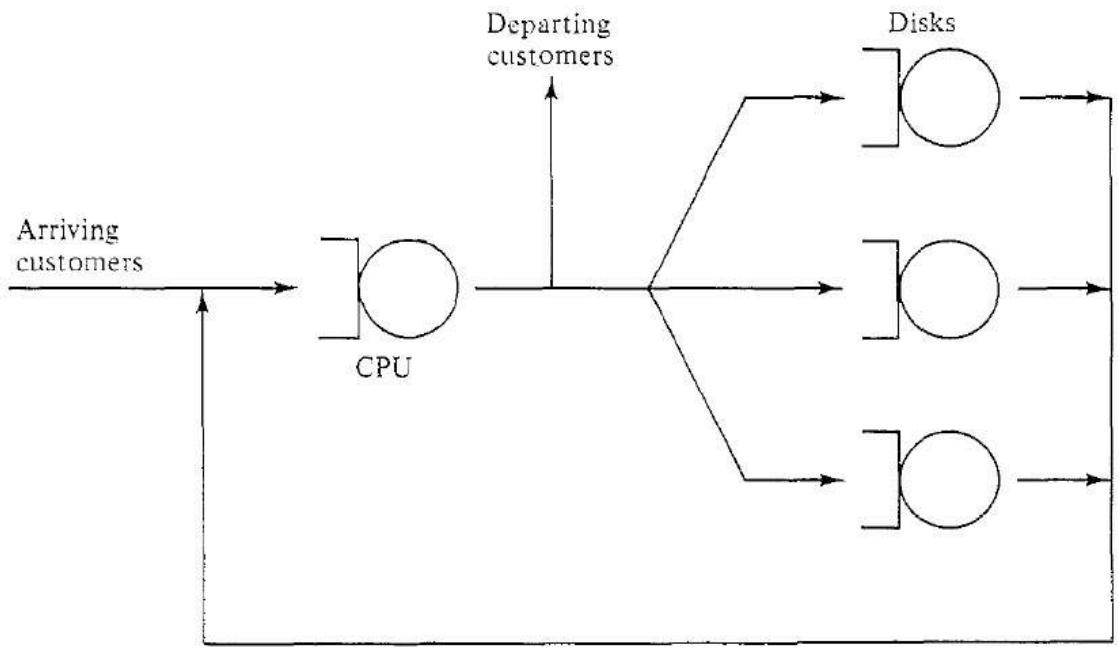
$$\frac{1}{t}\int_0^t N(s)ds \approx \frac{1}{t}\sum_{i=1}^{A(t)} T_i$$

and thus **Little's Law** states: $\mathbb{E}[N] = \lambda \mathbb{E}[T]$

Jackson, BCMP and Kelly networks



Jackson, BCMP and Kelly networks



$$\mathbb{P}(N_1 = n_1, N_2 = n_2, \ldots, N_K = n_K) = \Pi_{i=1}^{K} \mathbb{P}(N_i = n_i)$$

where $\mathbb{P}(N_i = n_i) = (\lambda/\mu)^{n_i}(1 - \lambda/\mu)$.

In **steady-state** the queues behave as if they were **isolated** and **independent** from each other.

Limiting regimes and Comparisons

- **Heavy-Traffic**, when the system is in saturation [VAN09]

$$\lim_{\lambda \to \mu} (\mu - \lambda)\mathbb{P}(N_1 = n_1, N_2 = n_2, \ldots, N_K = n_K) \stackrel{d}{=} X \cdot (\rho_1, \rho_2, \ldots, \rho_K)$$

Limiting regimes and Comparisons

- **Heavy-Traffic**, when the system is in saturation [VAN09]

$$\lim_{\lambda \to \mu} (\mu - \lambda)\mathbb{P}(N_1 = n_1, N_2 = n_2, \ldots, N_K = n_K) \stackrel{d}{=} X \cdot (\rho_1, \rho_2, \ldots, \rho_K)$$

- **Fluid limit**. For large Δ, $N(\Delta t) \approx N(\Delta(t-\epsilon)) + \lambda\Delta\epsilon - \mu\Delta\epsilon$. In certain cases it holds $\lim_{\Delta\to\infty} \frac{N(\Delta t)}{\Delta} = n(t)$ where $n(t)$ is the solution of an ordinary differential equation. For the previous example $\frac{dn(t)}{dt} = \lambda - \mu$.
  Performance Evaluation and Optimal Control [APZ08]

Limiting regimes and Comparisons

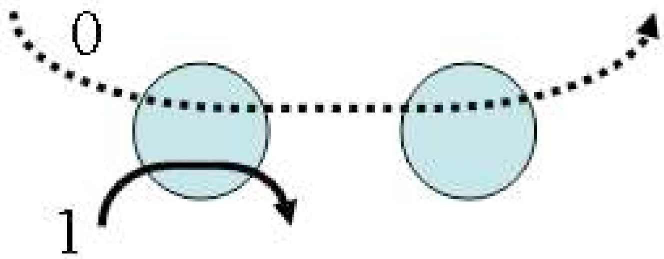
- **Heavy-Traffic**, when the system is in saturation [VAN09]

$$\lim_{\lambda \to \mu} (\mu - \lambda)\mathbb{P}(N_1 = n_1, N_2 = n_2, \ldots, N_K = n_K) \stackrel{d}{=} X \cdot (\rho_1, \rho_2, \ldots, \rho_K)$$

- **Fluid limit**. For large Δ, $N(\Delta t) \approx N(\Delta(t-\epsilon)) + \lambda\Delta\epsilon - \mu\Delta\epsilon$. In certain cases it holds $\lim_{\Delta\to\infty} \frac{N(\Delta t)}{\Delta} = n(t)$ where $n(t)$ is the solution of an ordinary differential equation. For the previous example $\frac{dn(t)}{dt} = \lambda - \mu$.

  Performance Evaluation and Optimal Control [APZ08]

- **Sample-path Comparison** [VAB09]



If $\vec{W}^{\pi}(0) = \vec{W}^{\tilde{\pi}}(0)$, then

i) $N_0^{\pi}(t) \geq N_0^{\tilde{\pi}}(t)$ and $W_0^{\pi}(t) \geq W_0^{\tilde{\pi}}(t)$,

ii) $W_0^{\pi}(t) + W_i^{\pi}(t) \geq W_0^{\tilde{\pi}}(t) + W_i^{\tilde{\pi}}(t)$

Limiting regimes and Comparisons

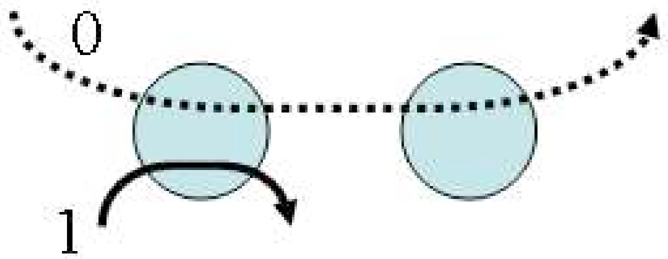
- **Heavy-Traffic**, when the system is in saturation [VAN09]

$$\lim_{\lambda \to \mu} (\mu - \lambda)\mathbb{P}(N_1 = n_1, N_2 = n_2, \ldots, N_K = n_K) \stackrel{d}{=} X \cdot (\rho_1, \rho_2, \ldots, \rho_K)$$

- **Fluid limit**. For large Δ, $N(\Delta t) \approx N(\Delta(t-\epsilon)) + \lambda\Delta\epsilon - \mu\Delta\epsilon$. In certain cases it holds $\lim_{\Delta\to\infty} \frac{N(\Delta t)}{\Delta} = n(t)$ where $n(t)$ is the solution of an ordinary differential equation. For the previous example $\frac{dn(t)}{dt} = \lambda - \mu$.

  Performance Evaluation and Optimal Control [APZ08]

- **Sample-path Comparison** [VAB09]



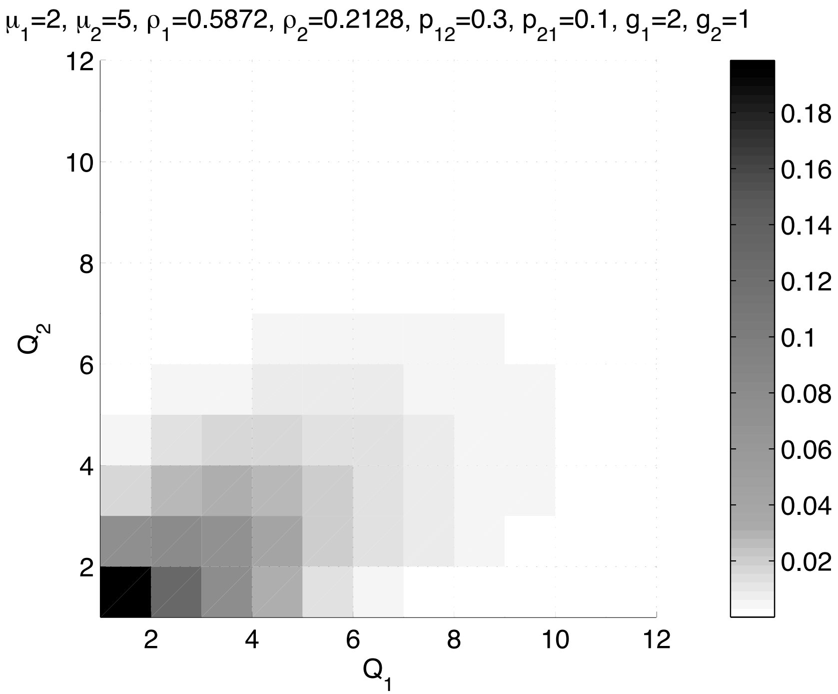
If $\vec{W}^{\pi}(0) = \vec{W}^{\tilde{\pi}}(0)$, then

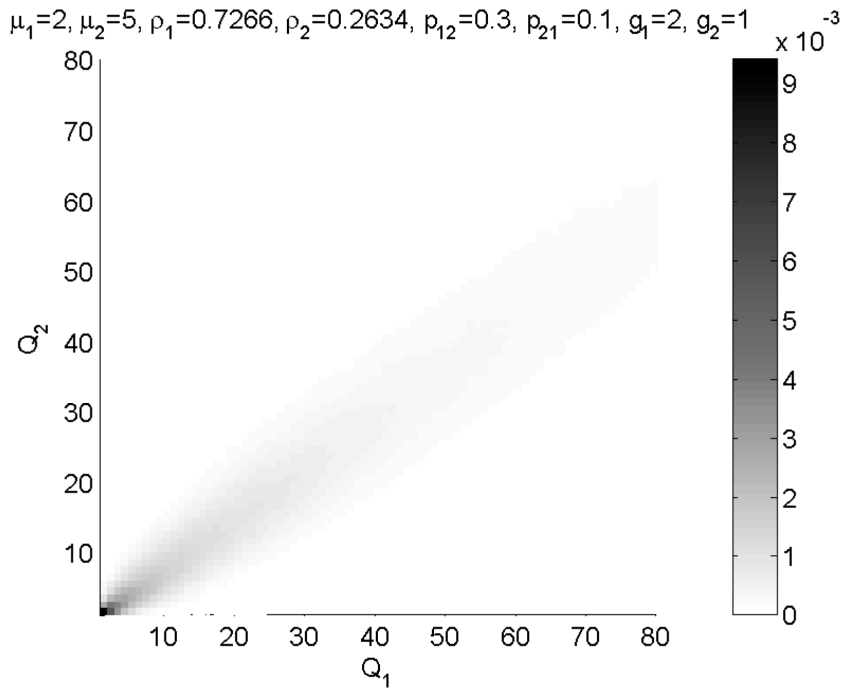
i) $N_0^{\pi}(t) \geq N_0^{\tilde{\pi}}(t)$ and $W_0^{\pi}(t) \geq W_0^{\tilde{\pi}}(t)$,

ii) $W_0^{\pi}(t) + W_i^{\pi}(t) \geq W_0^{\tilde{\pi}}(t) + W_i^{\tilde{\pi}}(t)$

- **Mean Field Limit**, **Large deviations** and **Differential Traffic Theory**

Heavy-Traffic: State space collapse





Three examples

- Size-based Scheduling
- Conservation Law in queues
- Server Farms

Fair Policy: Processor Sharing Policy

- **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, Yashkov, Cohen, Kelly, Boxma, Robert]

Example 1: Size-based scheduling

- **Experimental evidence:** Mice and Elephants traffic pattern, 80% of the connections are short, 5% of largest flows make up for 95% of the load

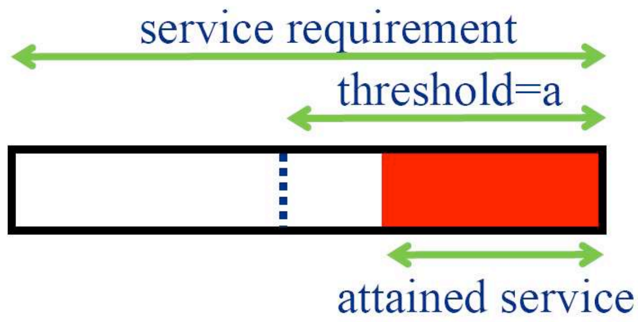
- Preferential treatment to short connections?

- **Evaluate the performance consequences:**
  - To what extent is the performance of large connections degraded?
  - What happens to the average number of connections? What are the consequence?

2PS(a)

Jobs are classified into two groups depending on the amount of service they have received.

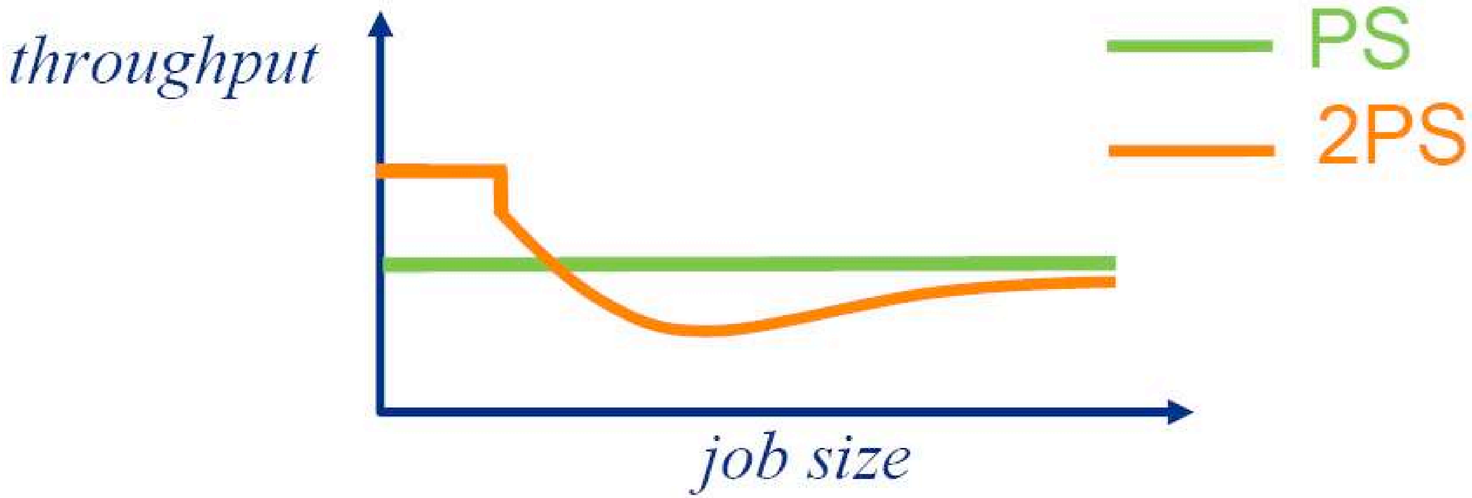
- **High Priority:** Jobs that have obtained less units of service than a.
- **Low Priority:** Jobs that have obtained more units of service than a. Within one priority level, jobs are served according to PS.



Asymptotic throughput of 2PS(a)

Theorem [AAB06]: The throughput obtained by large jobs is the same under both systems

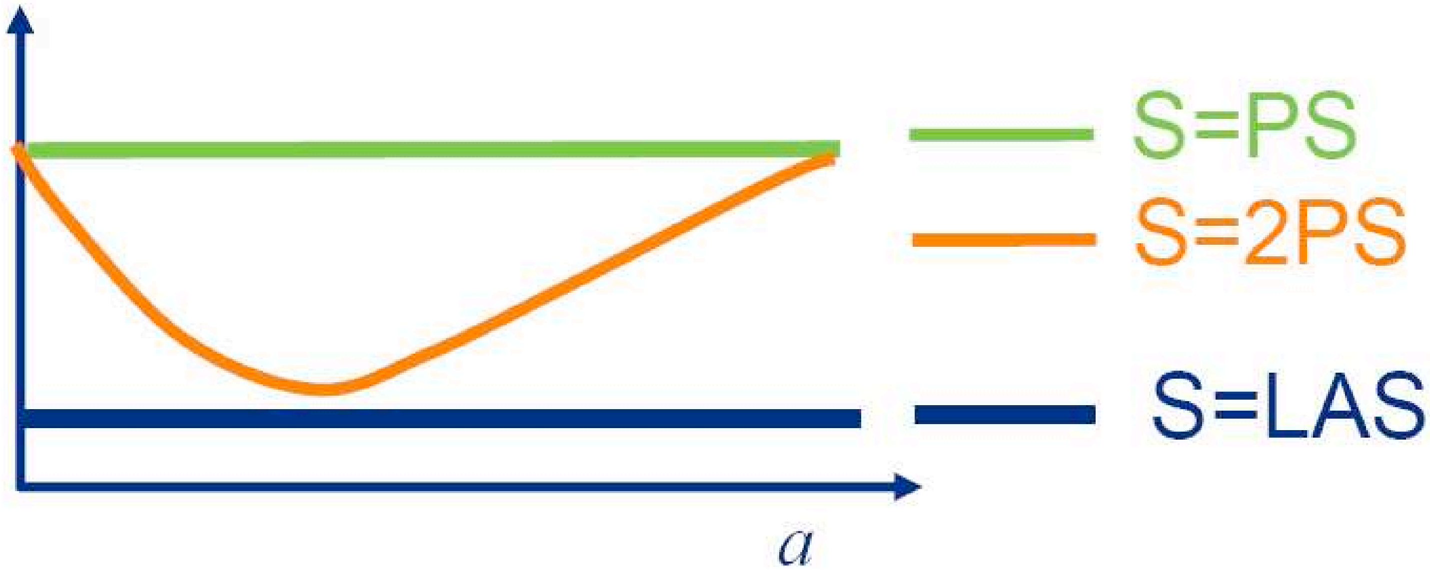
$$\lim_{x\to\infty} \frac{x}{\mathbb{E}[T^{2PS}|X=x]} = \frac{x}{\mathbb{E}[T^{PS}|X=x]}$$



Comparison between 2PS(a) and PS

Theorem [AA05]: If the hazard rate of the distribution function is decreasing:

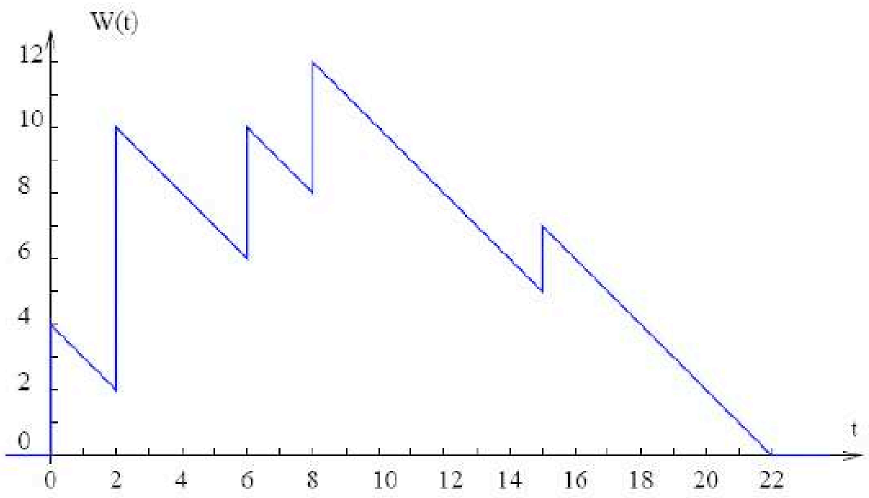
$$\mathbb{E}[N^{2PS}] \leq \mathbb{E}[N^{PS}]$$



Example 2: Conservation Law for single server queues

Example 2: Conservation Law for single server queues

Let $W(t)$ denote the total work in the system at time t. The evolution of $W(t)$ is **independent** of the scheduling policy.



number	arrival	service
1	0	4
2	2	8
3	6	4
4	8	4
5	15	2

Conservation Law for single server queues

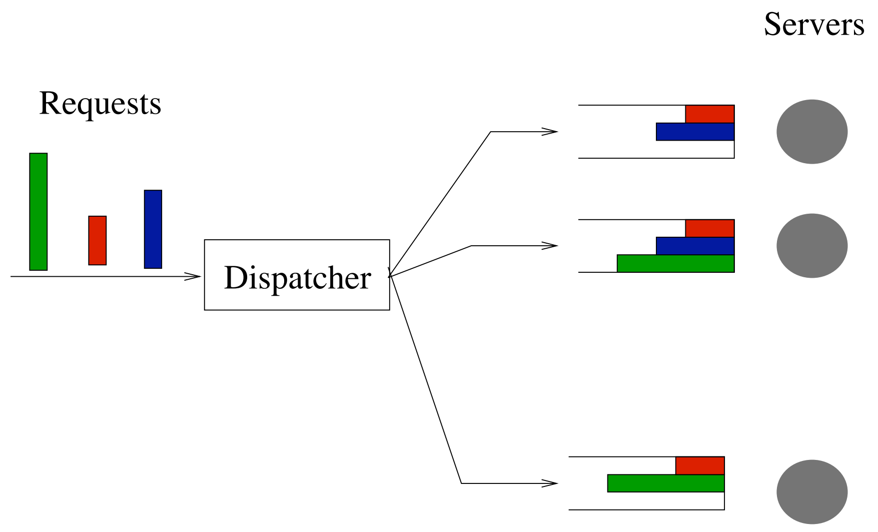
Theorem [A07]: In a single server queue with M classes with arbitrary scheduling discipline π:

$$\sum_{j=1}^{M} \lambda_j \int_0^\infty \mathbb{E}[T_j^\pi | X_j = x] \mathbb{P}(X_j > x) dx = \mathbb{E}[W]$$

- Application to comparison of policies $\mathbb{E}[T^{\pi_1}] \leq \mathbb{E}[T^{\pi_2}]$ [A07]
- Characterization of large sojourn times $\lim_{x\to\infty} \mathbb{E}[T_j^\pi | X_j = x]$ [AAB08]

Example 3: Server farms

- Diverse applications : e-service industry, database systems, grid computing clusters



Design problem: What is the optimal routing policy?

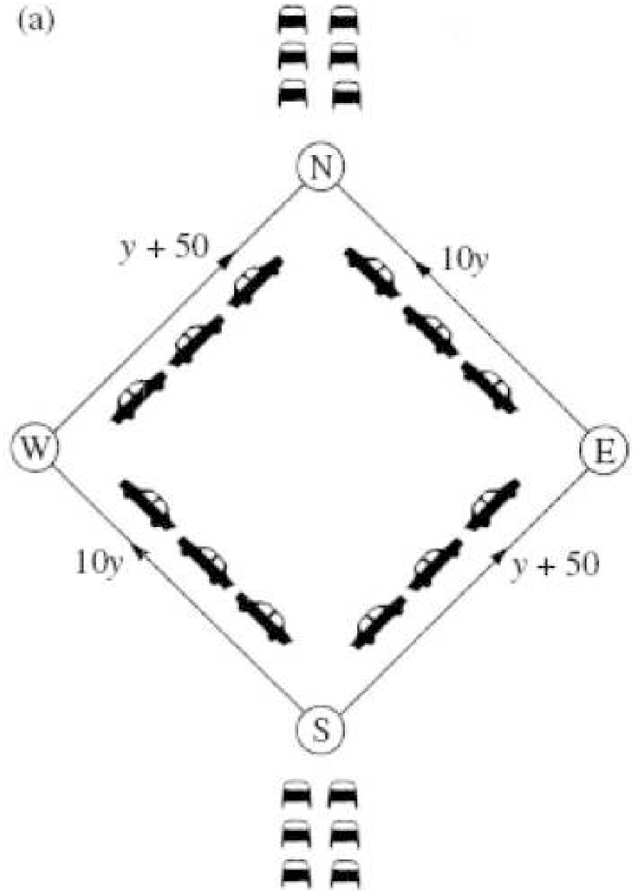
- Centralized setting: dispatcher takes decisions
- Decentralized setting: requests take decisions

Decentralized setting: Wardrop equilibrium

Total flow from S to N is 6

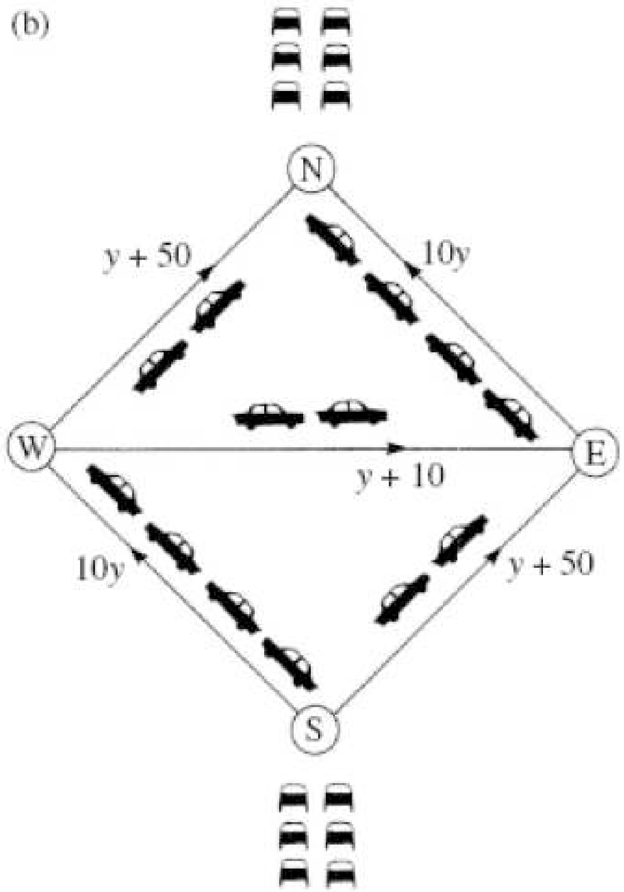
Wardrop equilibrium: 3 units travel via W, and 3 via E

Total Delay: $(10\times 3)+(3+50) = 83$



Comparing the Global and Individual: Braess' Paradox

A new link is added:



Comparing the Global and Individual: Braess' Paradox

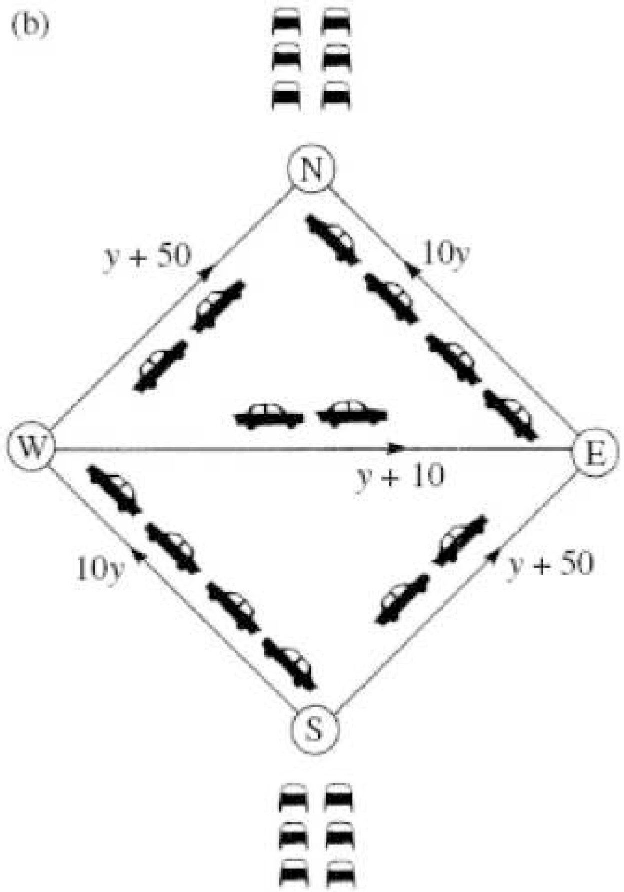
A new link is added:

There are 3 possible routes with the same delay:

$$(10 \times 4) + (2 + 50) = 92$$

$$(10\times4)+(2+10)+(10\times4) = 92$$

Adding a new link increases everyone's delay!



Example application

Internet based source code repositories - SourceForge, Google Code: Source files are hosted on several mirror sites

Filename	Size	Downloads
(2008-10-15 09:09)		
Azureus4.0.0.0.jar	12281931	1997
Azureus4.0.0.0.jar.torrent	7978	725
Vuze_4.0.0.0_linux.tar.bz2	13264417	994
Vuze_4.0.0.0_linux-x86_64.tar.bz2	13358925	145
Vuze_4.0.0.0_macosx.dmg	9052160	5853
Vuze_4.0.0.0_pluginapi.jar	540611	35
Vuze_4.0.0.0_source.zip	8143146	287
Vuze_4.0.0.0_windows.exe	9080760	33937

Select a different mirror:

- Asia
- Australia
- Europe
  - Lausanne, Switzerland
  - Duesseldorf, Germany
  - Paris, France
  - Berlin, Germany
  - Dublin, Ireland
  - Bologna, Italy
  - Amsterdam, The Netherlands
  - Kent, UK
- North America
- South America
- Auto-select

- Decision is taken either by the central unit or by the downloader
- Downloads progress in parallel $\Rightarrow$ Processor Sharing (PS) at each server

Centralized setting

- Solve the following mathematical program :

$$
\begin{aligned}
\text{minimize} \quad & \sum_{j \in \mathcal{S}} c_j \mathbb{E}[N(\mathbf{p})] \\
\text{subject to} \quad & \sum_{j \in \mathcal{S}} p_{ij} = 1, \ \text{ for all } i \in \mathcal{K}; \\
& \mathbf{p} \succeq \mathbf{0};
\end{aligned}
$$

Decentralized setting

Equilibrium: A strategy $\mathbf{p}$ is **an equilibrium** if for each class $i = 1, ..., K$ and each queue k used by class i,

$$\mathbb{E}[c_k T_k(\mathbf{p})|i] = \min_{j=1,...,K} \mathbb{E}[c_j T_j(\mathbf{p})|i]$$

Decentralized setting

Equilibrium: A strategy $\mathbf{p}$ is **an equilibrium** if for each class $i = 1, ..., K$ and each queue k used by class i,

$$\mathbb{E}[c_k T_k(\mathbf{p})|i] = \min_{j=1,...,K} \mathbb{E}[c_j T_j(\mathbf{p})|i]$$

Potential Games. The distributed non-cooperative game can be transformed into the standard convex optimization problem

$$\begin{array}{ll} \min\limits_{\mathbf{p}} & \displaystyle\sum_{k=1}^{C} c_k \log\left(\frac{1}{1-\rho_k(\mathbf{p})}\right) \\ \text{subject to} & 0 < \rho_j < 1, \quad \displaystyle\sum_j r_j \rho_j = \overline{\eta}. \end{array}$$

$\Rightarrow$ The game belongs to a particular type of games known as **"Potential Game"** [Shapley et al. 1996]

Comparing the Global and Individual

Price of Anarchy: [Papadimitriou 98] Defined as the ratio between the performance (mean delay) obtained by the Wardrop equilibrium and the global optimal solution.

$\Rightarrow$ A measure for the inefficiency of the decentralized scheme.

$$PoA = \sup_{\vec{\lambda},\vec{c},\vec{r}} \left(\frac{\text{Perfomance Decentralized Setting}}{\text{Global optimum}} \right); \quad PoA \in [1, \infty)$$

Comparing the Global and Individual

$$PoA = \sup_{\vec{\lambda},\vec{c},\vec{r}} \left(\frac{\text{Perfomance Decentralized Setting}}{\text{Global optimum}} \right); \quad PoA \in [1, \infty)$$

Theorem [AAP08]. For every θ, there exist c_j and r_j, $j \in \mathcal{S}$, such that $PoA > \theta$.

$\Rightarrow$ The PoA is unbounded.

If $c_k = 1$, then $PoA \leq C$ [Haviv and Roughgarden, 2007].

Comparing the Global and Individual

$$PoA = \sup_{\vec{\lambda},\vec{c},\vec{r}} \left(\frac{\text{Perfomance Decentralized Setting}}{\text{Global optimum}} \right); \quad PoA \in [1, \infty)$$

Theorem [AAP08]. For every θ, there exist c_j and r_j, $j \in \mathcal{S}$, such that $PoA > \theta$.

$\Rightarrow$ The PoA is unbounded.

If $c_k = 1$, then $PoA \leq C$ [Haviv and Roughgarden, 2007].

Theorem [ABP09]. If there are K selfish users, then $PoA = \sqrt{K}$

Missing result...

$$\begin{array}{ll} \text{Max}_{\vec{\lambda}:\sum_k \lambda_k = \Lambda} & \sum_{i=1}^{K} D_i(\mathbf{p_1^*}(\vec{\lambda}), \ldots, \mathbf{p_K^*}(\vec{\lambda})) \\ \text{where} & \mathbf{p_1^*}, \ldots, \mathbf{p_K^*} \text{ is s.t.} \end{array}$$

$$D_i(\mathbf{p_1^*}, \ldots, \mathbf{p_K^*}) = \min_{\mathbf{p_i}:\sum_{j=1}^{L} p_{ij} = \lambda_i} D_i(\mathbf{p_1^*}, \ldots, \mathbf{p_{i-1}^*}, \mathbf{p_i}, \mathbf{p_{i+1}^*}, \ldots, \mathbf{p_K^*})$$

Conjecture: The solution is $\vec{\lambda} = (\Lambda/K, \ldots, \Lambda, K)$?

Conclusions and Future work

- Interaction between **Game Theory** and **Queueing**
- **Wireless Systems**. Capacity Changes over time.
  - Need for new mathematical model and paradigms
- **Wired Networks**. Internet will be everywhere. Elastic (web, email, ...) and Streaming (VoIP, video-on-demand) applications with very different QoS requirements
  - Need for new mathematical models for the design and performance evaluation of such networks.
- **Power** might be the key performance criteria!
- **Peer-to-Peer** Networks, **AdHoc** Networks