Switched Networks

Switched networks are a useful paradigm for studying scheduling and resource allocation in networks.

* They have applications to call centres, Internet routers, Internet congestion control, wireless data transmission, road transport, ...

* Even when modelling details (distributions etc.) are made as simple as possible, they still present challenges

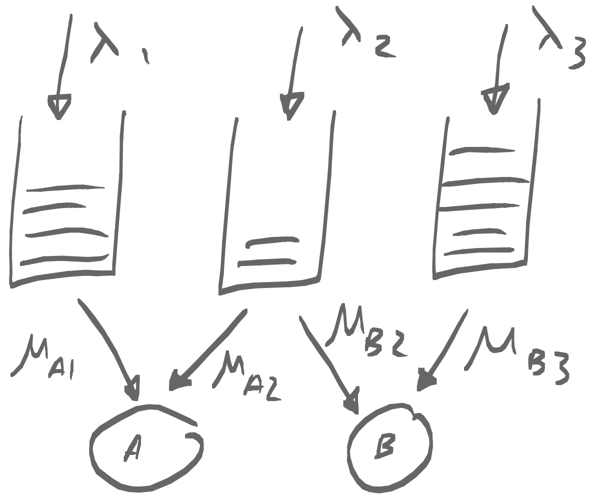
* The key idea dates to 1992; since 2000 a substantial body of modern theory has been developing. [Tassiulas + Ephremides 1992]

* There is lots left to do!

Outline

Monday :	Modelling applications	heuristics
	Fluid model	
	Stability region	
	Scheduling algorithms	
	Lyapunov function	
Wednesday :	Fluid limit	rigor
Friday :	Workloads	general idea
	Underload — stability analysis	
	Critical load — heavy traffic	
	Overload	
	Underload — large deviations	

A model of a call centre



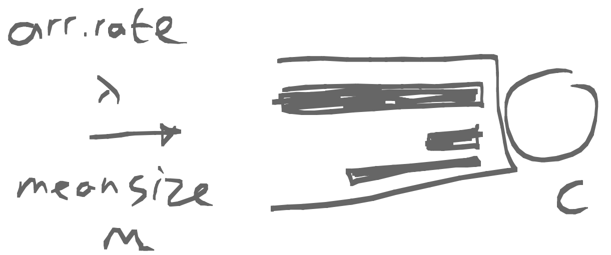
Three classes of customer, each with its own queue.
Customers in class i arrive as a Poisson process of rate λ_i.

Two servers, A and B. When server A serves a customer of class i, the service time is $\sim \text{Exp}(\mu_{Ai})$. Similarly for B. No interruption.

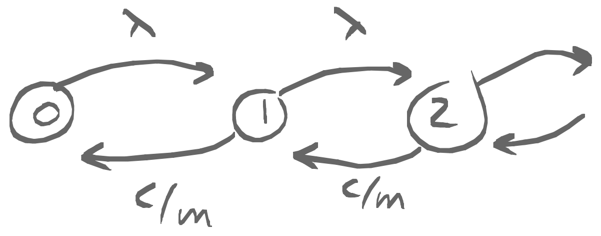
QUESTION. When a server finishes a job, who should it serve next? When a customer arrives and finds idle servers, which should be chosen?

$\longrightarrow$ what is a sensible objective?

Internet congestion control: background



RECALL the processor-sharing link:
jobs arrive as a Poisson process of rate λ,
job sizes are i.i.d. $\sim \text{Exp}(1/m)$,
when there are q jobs active, each gets
throughput $\theta(q) = C/q$

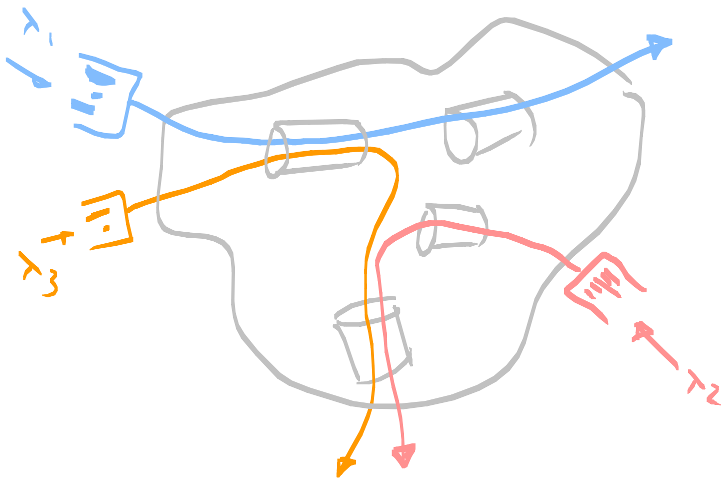

$$\begin{tikzcd}
0 \arrow[r, bend left, "\lambda"] & 1 \arrow[l, bend left, "C/m"] \arrow[r, bend left, "\lambda"] & 2 \arrow[l, bend left, "C/m"] \arrow[r, bend left] & {} \arrow[l, bend left]
\end{tikzcd}$$

Then, the number of active jobs at time t,
Q_t, is a Markov process:

up-rate λ

down-rate $\sigma(q) = q\dfrac{\theta(q)}{m} = \dfrac{C}{m}$.

A model of Internet congestion control



Let there be a collection of routes R.

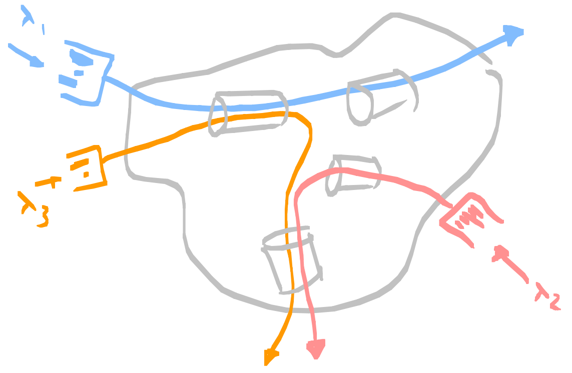
On route r, jobs arrive as a Poisson process of rate λ_r; job sizes are i.i.d. $\sim \mathrm{Exp}(\frac{1}{m_r})$.

When there are $q = (q_r)_{r \in R}$ jobs active, let each job on route r get throughput $\theta_r(q)$. Let $\sigma_r(q) = q_r \theta_r(q)/m_r$.

Then, the number of active jobs $Q(t)$ is a Markov process:

$$\begin{tikzcd}
(q_1-1, q_2, q_3) & & (q_1+1, q_2, q_3) \\
(q_1, q_2-1, q_3) & (q_1, q_2, q_3) \arrow[ul, "\sigma_1(q)"'] \arrow[l, "\sigma_2(q)"'] \arrow[dl, "\sigma_3(q)"] \arrow[ur, "\lambda_1"] \arrow[r, "\lambda_2"] \arrow[dr, "\lambda_3"] & (q_1, q_2+1, q_3) \\
(q_1, q_2, q_3-1) & & (q_1, q_2, q_3+1)
\end{tikzcd}$$

A model of Internet congestion control



TCP is the Internet's algorithm for controlling how fast each job of each user is sent, ie for choosing $\sigma(q)$. Specifically,

$\sigma(q)$ is chosen from $\{ p \in \mathbb{R}_+^N : Ap \leq C \}$,
to maximize $\frac{1}{1-\alpha} \sum p_r^{1-\alpha} q_r^{\alpha}$.

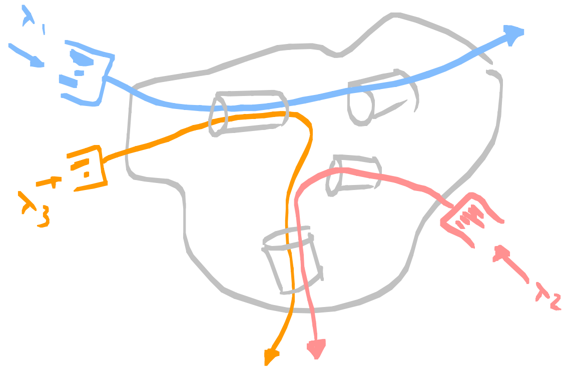
N = # routes

$Ap \leq C$ means: total throughput on each link is no more than the link speed

$\alpha \in (0, \infty)$ is an arbitrary parameter, set to $\alpha = 2$ by default.

[Roberts + Massoulié 1998]

Drift Model



$$\mathbb{E}\left(Q_r(t+\delta) \,\middle|\, Q(t) \right) = Q_r(t) + \#\text{arrivals} - \#\text{departures}$$
$$= Q_r(t) + \lambda_r \delta - \sigma_r(Q_r(t))\,\delta.$$

Heuristically,

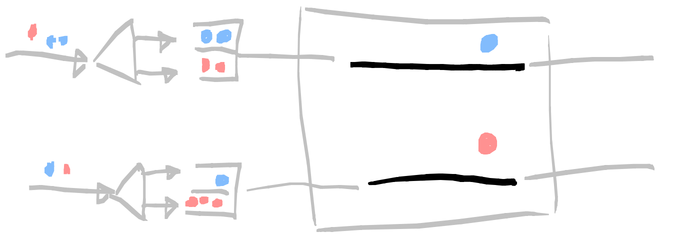
$$q_r(t+\delta) = q_r(t) + \lambda_r \delta - \sigma_r(t)\,\delta$$

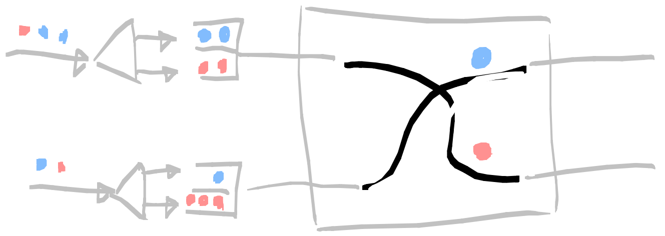
$$\Rightarrow \quad \frac{d}{dt} q(t) = \lambda - \sigma(t).$$

$\sigma(t)$ is chosen from $\{ \rho \in \mathbb{R}_+^N : A\rho \le C \}$, given $q(t)$, according to the TCP rule.

I call this a heuristic "drift model" to distinguish it from a rigorous "fluid model". Philosophically & practically, it is just as good as a stochastic model.

A model for an input-queued switch





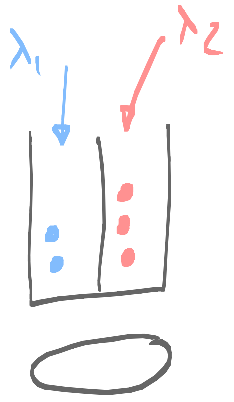
Consider an Internet router with two line cards, i.e. two inputs & two outputs. It operates in slotted time. Each timeslot, packets arrive at the inputs, and are classified according to their destination (hence $N=4$ queues in total).

Let arrivals to queue n be a Bernoulli process of rate λ_n.

Then the router chooses a schedule, ie a set of queues to which it offers service. There are constraints on which queues can be served simultaneously.

[Dai + Prabhakar 2000]

A generalized switch model



Consider a collection of N queues, in slotted time.

Let $A_n(t)$ = # arrivals to queue n in timeslot t,

$\mathbb{E} A_n(t) = \lambda_n$, arrivals are i.i.d.

$S = \{ (1,1), (1{\cdot}5, 0), (0, 2) \}$

Each timeslot, the switch chooses a service action $\rho \in S$, for some finite set $S \subset \mathbb{R}_+^N$.

Here, ρ_n = amount of service offered to queue n.

<u>Drift model</u>

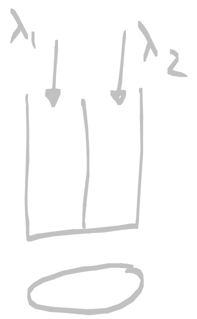
$$\mathbb{E}\big(Q_n(t+1) \mid Q(t)\big) = Q_n(t) + \lambda_n - \sigma_n(t)$$

$$\longrightarrow \quad \frac{d}{dt} q(t) = \lambda - \sigma(t)$$

$\sigma(t)$ is chosen from the convex hull $\langle S \rangle$, depending on $q(t)$, according to some rule.

[Stolyar 2004]

The static planning problem



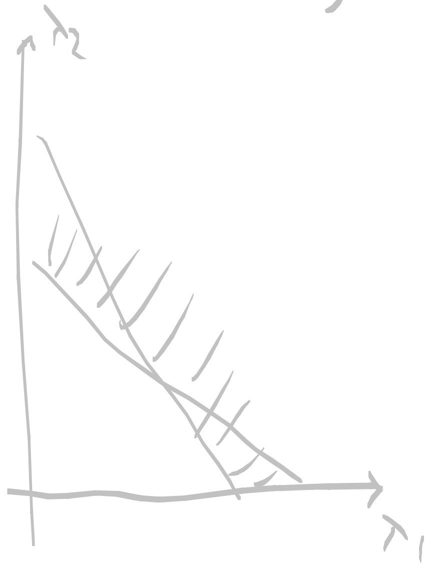
$S = \{ (1,1), (1.5, 0), (0, 2) \}$

For what $\lambda \in \mathbb{R}_+^N$ is it possible to serve all incoming work? Consider a static service policy which spends a fraction of time a_p on service action $p \in S$, so the total service rate is $\sigma = \sum a_p p \in \langle S \rangle$.

Define the <u>stability region</u> Λ to be

$$\Lambda = \{ \lambda \in \mathbb{R}_+^N : \ \lambda \leq \sigma \text{ for some } \sigma \in \langle S \rangle \}$$

If $\lambda \in \Lambda$, an omniscient planner could choose to serve according to σ; then the queues will not grow (according to the drift model).



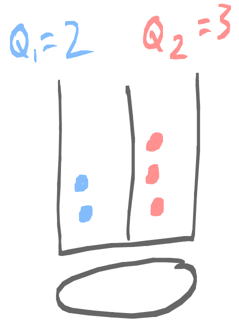
$\Lambda = \{ \lambda : \ \lambda_1 + \lambda_2 \leq 2 \text{ and } 2\lambda_1 + \lambda_3 \leq 3 \}.$

The goal of this area of study is to find easy-to-implement scheduling algorithms — and to do enough performance analysis to find out which scheduling algorithms work reasonably well.

So far, we understand only two* algorithms: TCP-inspired, and MaxWeight. What makes them tractable is their link to Lyapunov functions.

* and a few algorithms that apply to specific examples of switched networks

Max-Weight



$S = \{ (1,1),$ ← weight $2+3=5$

$(1{\cdot}5, 0),$ ← weight $2\times 1{\cdot}5 = 3$

$(0,2) \}$ ← weight $2\times 3 = 6.$

So serve $(0,2)$.

The Max-Weight algorithm picks

$$\sigma(t) \in \operatorname*{argmax}_{\sigma\in\langle S\rangle} \sum_n q_n(t)\,\sigma_n .$$

This is appealing because it is myopic, i.e. it makes its choice based entirely on the current state, not on any past measurements. It doesn't attempt to explicitly measure the arrival rates. This means it tends to adapt well when arrival rates fluctuate.

Max-Weight and Lyapunov function

The Max-Weight algorithm picks $\sigma(t) \in \operatorname*{argmax}_{\sigma \in \langle S \rangle} \sum_n q_n(t)\,\sigma_n$.

Consider $L(q) = \frac{1}{2}\sum_n q_n^2$: and suppose $\lambda \in \Lambda$

$$\begin{aligned}
\frac{d}{dt} L(q(t)) &= \sum_n q_n(t) \frac{d}{dt} q_n(t) \\
&= \sum_n q_n(t)\,(\lambda_n - \sigma_n(t)) && \text{drift model for } q_n(t) \\
&\leq \sum_n q_n(t)\rho - \sum_n q_n(t)\,\sigma_n(t) && \text{since } \lambda \in \Lambda,\ \lambda \leq \rho \\
&\qquad \text{for some } \rho \in \langle S \rangle \\
&\leq 0. && \text{since } \rho \in \langle S \rangle, \text{ and } \sigma(t) \text{ has max.weight over } \langle S \rangle.
\end{aligned}$$

[Tassiulas + Ephremides 1992]

"Theorem"

If $\lambda \in \Lambda$, then $\lambda \leq \sigma$ for some $\sigma \in \langle S \rangle$,
ie an omniscient planner could choose to serve at rate σ,
and $\frac{d}{dt} q_n(t) = \lambda_n - \sigma_n \leq 0$ so the queues will be stable.

A myopic controller could use MaxWeight, i.e. serve at rate

$$\sigma(t) \in \operatorname*{argmax}_{\sigma \in \langle S \rangle} \sum_n q_n(t) \sigma_n(t)$$

and this too will keep the queues stable. We say
"MaxWeight has 100% throughput".

On the other hand, if $\lambda \notin \Lambda$, then it is impossible
to prevent some queue or collection of queues from growing
linearly in time.

In many related models, we can devise scheduling algorithms by starting with a Lyapunov function and choosing a service action to make it decrease as fast as possible.

The trick is finding a good Lyapunov function.

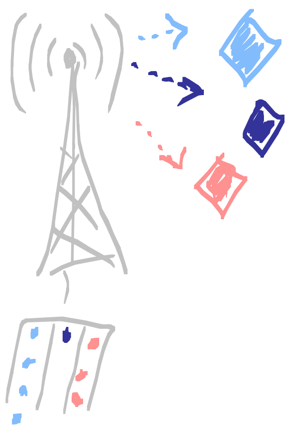
$$L(q) = \frac{1}{1+\alpha} \sum_n q_n^{1+\alpha} \longrightarrow \text{Max Weight family}$$

$$L(q) = \frac{1}{1+\alpha} \sum_n \frac{q_n^{1+\alpha}}{\lambda_n^{\alpha}} \longrightarrow \text{TCP family}$$

(How can we advance beyond these?)

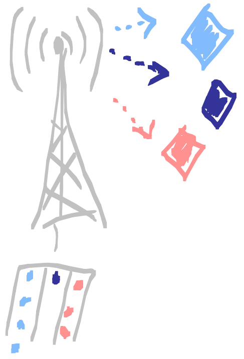
A model for wireless downlink

novel feature: a random environment



A wireless base station is transmitting to N devices. New work arrives for device n at rate λ_n, and is queued. The "state of nature" M evolves according to a finite-state irreducible Markov chain, with eqm distribution $\pi^m = \mathbb{P}(M=m)$. In state m, the base station knows m and picks an action $\rho \in \mathcal{S}^m \subset \mathbb{R}_+^N$; ρ_n is the service rate for queue n with this choice.

A model for wireless downlink



To define Λ:

A static service policy is a collection of $\sigma^m \in \langle \mathcal{S}^m \rangle$, specifying a convex combination of actions for each m.

$$\Lambda = \left\{ \lambda \in \mathbb{R}_+^N : \quad \lambda \le \sum_m \pi^m \sigma^m \text{ for some collection of } \sigma^m \in \langle \mathcal{S}^m \rangle \right\}$$

The drift model

keeps track of $s_p^m(t)$ = amount of time in $[0,t]$ that state of nature $= m$ and action $p \in \mathcal{S}^m$ is chosen.

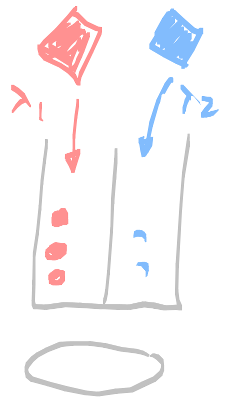
The Lyapunov function

$L(q) = \frac{1}{2} \sum q_n^2$ works exactly as before.

The MaxWeight policy: to maximize the decrease in $L(q(t))$, when in state m, pick $\sigma(t) \in \underset{p \in \mathcal{S}^m}{\operatorname{argmax}} \sum p_n q_n(t)$.

A model with utilities

Novel feature: arrival rates are to be chosen optimally

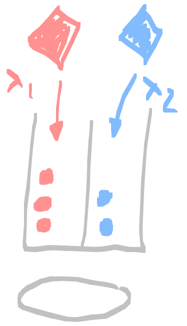


It is often the case that data rates aren't given exogeneously — rather, they are chosen adaptively. A sensible way to choose them is e.g. to

$$\text{maximize } \sum_n \frac{\lambda_n^{1-\alpha}}{1-\alpha} \quad \text{over } \lambda \in \mathbb{R}_+^N$$
$$\text{such that } \lambda \leq \sigma \text{ for some } \sigma \in \langle S \rangle.$$

[Eryilmaz + Srikant 2005]

A model with utilities



Heuristic: Write out the Lagrangian,

$$\mathcal{L}(\lambda, \sigma; q) = \sum_n \frac{\lambda_n^{1-\alpha}}{1-\alpha} - \sum_n q_n (\lambda_n - \sigma_n).$$

The complementary slackness condition is $\lambda_n = q_n^{-1/\alpha}$.

This suggests: let user n choose his/her data rate $\lambda_n(t) = q_n(t)^{-1/\alpha}$.

IDEA: There is a dual variable for the constraint "$\lambda_n \le \sigma_n$". Interpret it as queue size.

Lyapunov function:

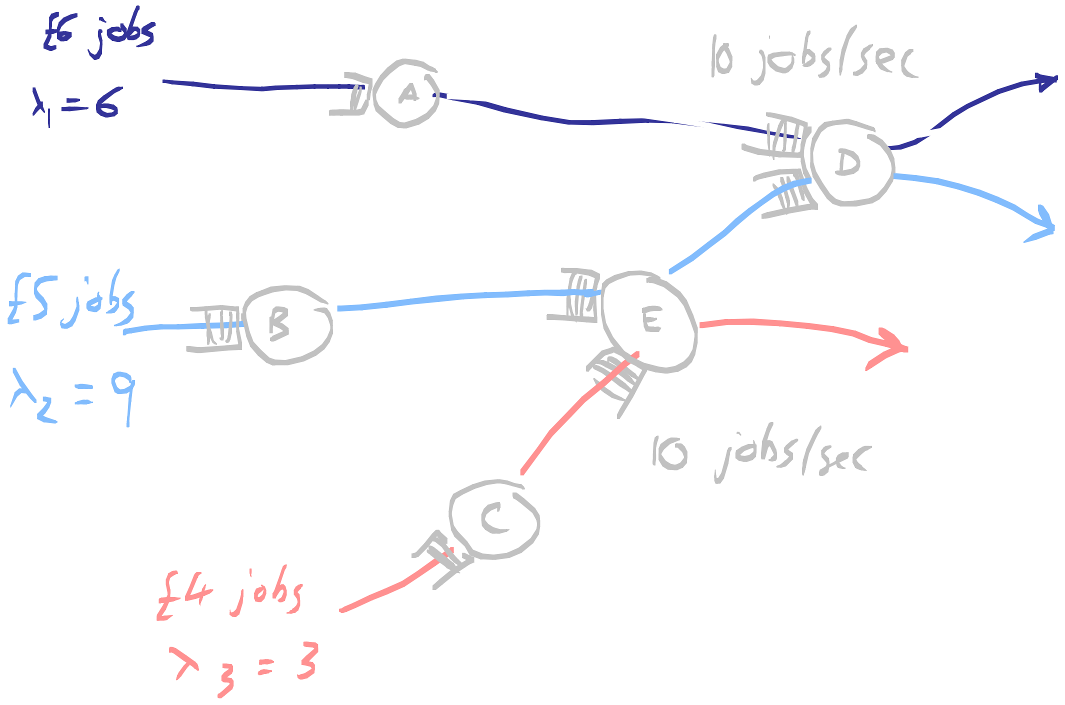
Suppose the utility-maximizing rates are λ^*.

Let $q_n^* = (\lambda_n^*)^{-\alpha}$. Then a Lyapunov function is

$$L(q) = \sum_n (q_n - q_n^*)^2.$$

A model for a web data centre

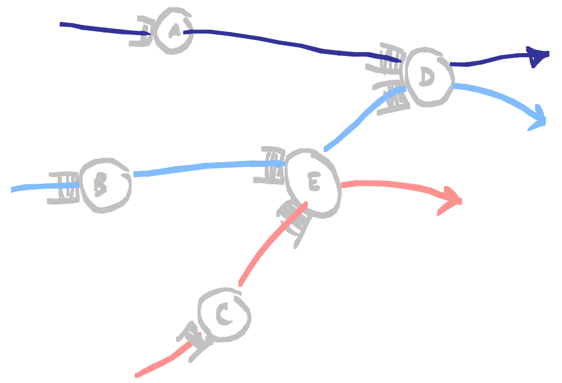
Novel feature: multihop, the "backpressure" algorithm



In a data centre, requests may have to be processed by several different nodes.

When the data centre is overloaded, requests have to be discarded. Is there a simple adaptive scheme that learns it should discard 5/sec of the £5 jobs?

A model for a web data centre



Guess the Lyapunov function

$$L(q) = \frac{1}{1+\alpha} \sum_n w_n q_n^{1+\alpha}$$

for arbitrary constants $\alpha > 0$, $w_n > 0$.

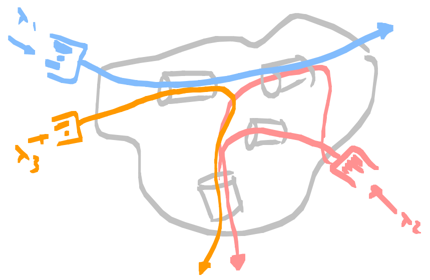
The scheduling rule that makes $\frac{d}{dt} L(q(t))$ as large negative as possible is:

$$\text{pick } \sigma(t) = \underset{\rho \in S}{\arg\max} \sum_n \rho_n \Big(w_n q_n(t)^\alpha - \underbrace{w_d q_d(t)^\alpha}_{\text{queue downstream of } n\text{, if there is one.}} \Big)$$

This is called "backpressure".

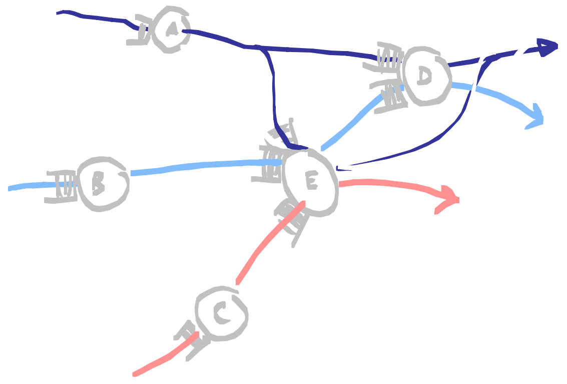
It says: if the queue downstream is too big, don't send work to it.

Models with route choice



In the bandwidth-sharing model, it is simple to add route choice: it simply changes the capacity constraint $Ap \le C$. The Lyapunov function is the same as before.

[Kang, Kelly, Lee, Williams 2009]



In the multihop queueing network model, it is simple to add route choice; the Lyapunov function is the same as before, and the nodes with route choice just use backpressure.

Summary

A switched network consists of a collection of queues, with constraints on which queues may be served concurrently. Their evolution can be described heuristically by a drift model

$$\frac{d}{dt} q_n(t) = \lambda_n - \sigma_n(t) \qquad (\text{when } q_n(t) > 0)$$

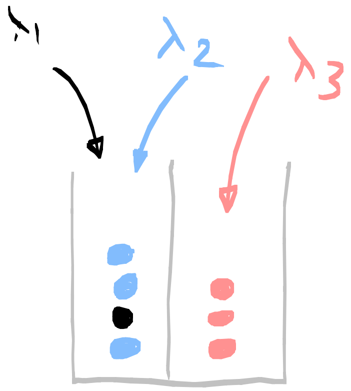
and the service constraint is:

$\sigma(t) \in \langle S \rangle$ for some finite set $S \subset \mathbb{R}_+^N$.

The scheduling algorithm restricts which $\sigma(t) \in \langle S \rangle$ is chosen. We described the MaxWeight family and the TCP family. They can both be thought of as "steepest descent of the Lyapunov function".

The stability region Λ is the set of arrival rates λ such that the drift model can be stabilized. The MaxWeight and TCP algorithms stabilize the drift model, whenever $\lambda \in \Lambda$.

Open Problem I



$$S = \left\{ (1, 0, 1), \; (0, 2, 0) \right\}$$

Suppose it's not the case that each traffic class has its own queue — ie suppose different classes can share a queue.

Suppose also that queues are strictly FIFO.
e.g. if we pick service action $(1,0,1)$ now then the offered service to class 1 is wasted.

This is called <u>head-of-line blocking</u>.

What is an appropriate drift model? Stability region? Lyapunov function? Scheduling algorithm?

Switched Networks

A switched network consists of a collection of queues, with constraints on which queues may be served concurrently. Their evolution can be described heuristically by a drift model

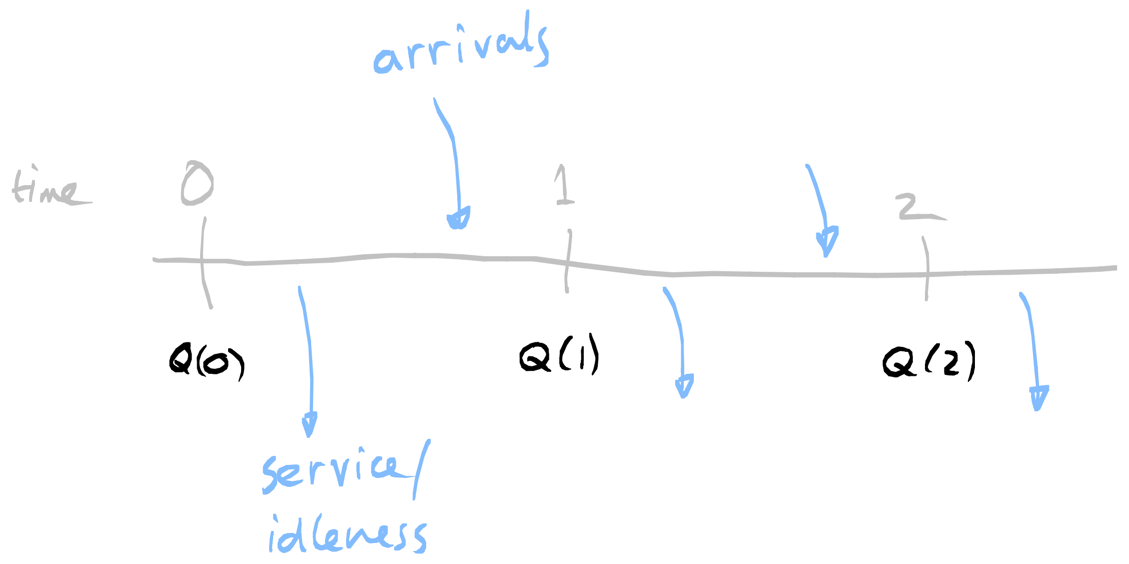
$$\frac{d}{dt} q_n(t) = \lambda_n - \sigma_n(t) \qquad (\text{when } q_n(t) > 0)$$

and the service constraint is:

$$\sigma(t) \in \langle S \rangle \quad \text{for some finite set } S \subset \mathbb{R}_+^N.$$

The scheduling algorithm restricts which $\sigma(t) \in \langle S \rangle$ is chosen. We described the MaxWeight family and the TCP family.

Queueing Dynamics



$Q(t)$ = queue size at time t, $\quad Q(0)=0$

$A(t)$ = arrivals in the interval $(0,t)$, $\quad A(0)=0$,

$S_\pi(t)$ = # times that action $\pi \in \mathcal{S}$ was chosen in $(0,t)$. $\quad S_\pi(0)=0$

$Z(t)$ = idleness incurred in $(0,t)$, $\quad Z(0)=0$.

Queue Dynamics

The switched network (Q, A, Z, S) obeys

Queue size = init. size + work in − workout

$$Q(t) = Q(0) + A(t) - \sum_\pi \pi S_\pi(t) + Z(t)$$

Idleness increases when there isn't enough to serve

$$Z_n(t+1) = Z_n(t) + \Big[\sum_\pi \pi_n \big(S_\pi(t+1) - S_\pi(t) \big) - Q_n(t) \Big]^+$$

Use max-weight scheduling

$$S_\pi(t+1) = S_\pi(t) \quad \text{if} \quad \pi \cdot Q(t) < \max_{\rho \in S} \rho \cdot Q(t)$$

Always do something

$$\sum_\pi S_\pi(t) = t$$

Let the <u>fluid model equations</u> for abs. cts. functions (q, a, z, s) be (+)

$$q(t) = q(0) + a(t) - \sum_\pi \pi s_\pi(t) + z(t)$$

$$\dot{z}_n(t) = 0 \quad \text{if} \quad q_n(t) > 0$$

$$\dot{s}_\pi(t) = 0 \quad \text{if} \quad \pi \cdot q(t) < \max_{\rho \in S} \rho \cdot q(t)$$

(+)

$$\sum_\pi s_\pi(t) = t$$

$$a(t) = \lambda t$$

each $z_n(\cdot)$ and $s_\pi(\cdot)$ is non-decreasing

The Fluid Scaling and Fluid Limit

Let $q^r(t) = Q(rt)/r$ $\quad z^r(t) = Z(rt)/r$

$a^r(t) = A(rt)/r$ $\quad s^r_\pi(t) = S_\pi(rt)/r$ $\quad$ for $r \in \mathbb{N}$, $t \in \mathbb{R}_+$,

after extending the domain of Q etc. from $\mathbb{N}$ to $\mathbb{R}^+$ by linear interpolation.

let $x^r = (q^r, a^r, s^r, z^r)$ over some fixed period $[0,T]$.

Theorem

If there exists $\lambda \in \mathbb{R}_+^N$ and $\varepsilon_r \to 0$ such that

$$\sup_{t \in [0,T]} \left| a^r(t) - \lambda t \right| < \varepsilon_r \text{ for all } r$$

then

$\inf_{y \in FMS} \| x^r - y \|$

$$d(x^r, FMS) \longrightarrow 0 \text{ uniformly in } x^r$$

where FMS is the set of functions (q, a, s, z) which satisfy the fluid model equations, with $q(0) = 0$

FMS stands for "fluid model solutions"

Theorem If there exists $\lambda \in \mathbb{R}_+^N$ and $\varepsilon_r \to 0$ such that

$$\sup_{t\in[0,T]} \left| a^r(t) - \lambda t \right| < \varepsilon_r \quad \text{for all } r \qquad (*)$$

then

$$d\left(x^r, \text{FMS}\right) \longrightarrow 0 \quad \text{uniformly in } x^r = (q^r, a^r, s^r, z^r),$$

where FMS is the set of functions (q,a,s,z) which satisfy the fluid model equations, with $q(0)=0$

Corollary Suppose the arrival process to queue n is a Bernoulli process of rate λ_n. Pick e.g. $\varepsilon_r = r^{-1/6}$. Then

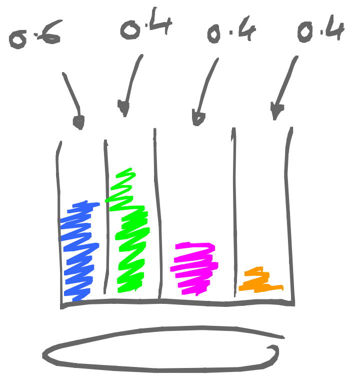
$$\mathbb{P}\left(a^r(\cdot) \text{ satisfies } (*) \right) \longrightarrow 1.$$

Therefore, for any $\delta > 0$,

$$\mathbb{P}\left(d(x^r, \text{FMS}) < \delta \right) \longrightarrow 1.$$

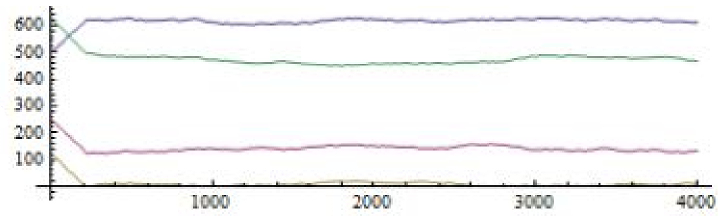
If we can prove something useful about the set FMS of fluid model solutions, we can deduce properties of $Q(\cdot)$.

Simulation

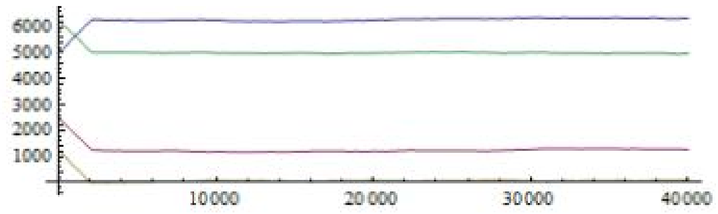


$$S = \{(1,0,0,0), (0,1,1,1)\}.$$

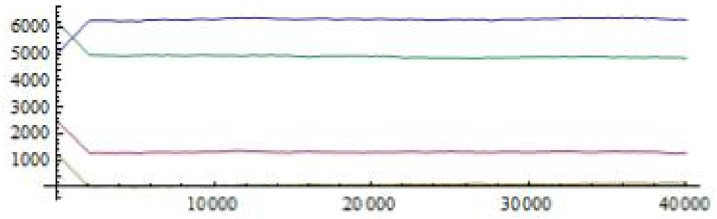
Bernoulli arrivals



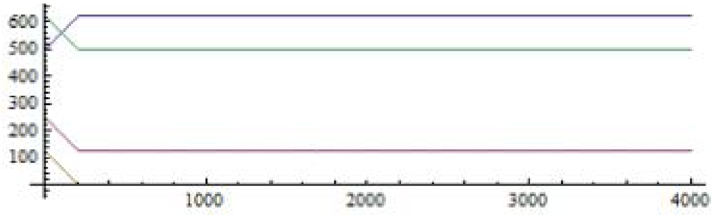
Bernoulli arrivals, longer timescale



Poisson arrivals



constant-rate arrivals



Proof: Let E_r be the event that $\{ a^r(\cdot) \text{ satisfies } (*) \}$.
$\mathbb{P}(E_r) \to 1$.

For all randomnesses $\omega \in E_r$, $a^r(\cdot)(\omega)$ satisfies $(*)$,
hence $d(x^r(\cdot)(\omega), FMS) \to 0$.

This convergence is uniform in x^r, ie

$$\sup_{\omega \in E_r} d(x^r(\cdot)(\omega), FMS) \longrightarrow 0.$$

ie $\forall \delta \; \exists r_0$ s.t. for $r \geq r_0$, $\sup_{\omega \in E_r} d(x^r(\cdot)(\omega), FMS) < \delta$.

$\Rightarrow \forall \delta \; \exists r_0$ s.t. for $r \geq r_0$, $\mathbb{P}\big(d(x^r, FMS) < \delta \big) \geq \mathbb{P}(E_r)$.

$\Rightarrow \forall \delta$, $\mathbb{P}\big(d(x^r, FMS) < \delta \big) \to 1$.

Proof of Fluid Limit

[Technique is due to Bramson, 1998]

Lemma

FLP stands for "fluid limit points"

Let $E_r = \{$ all possible fluid-scaled queue dynamics x^r on $[0,T]$, such that $\sup_{t\in[0,T]} |a^r(t) - \lambda t| < \varepsilon_r$ and $q^r(0) \le K$. $\}$

Let $FLP = \{$ continuous functions x on $[0,T]$ such that there is a subsequence $x^{r'} \in E_{r'}$ for which $x^{r'} \to x$ $\}$.

Then $\sup_{x\in E_r} d(x, FLP) \longrightarrow 0$.

"Every fluid-scaled version x^r is close to some limit point, for large r"

Lemma

If $x \in FLP$ then $x \in FMS$.

"If x is the limit of a subsequence of the x^r, then x satisfies the fluid model equations."

Example: deriving the MaxWeight eqn

CLAIM: If $x \in FLP$ then x satisfies the following equation:

$$\dot{s}_\pi(t) = 0 \quad \text{if } \pi \cdot q(t) < \max_{\rho \in S} \rho \cdot q(t) \qquad (\text{if } t \text{ is a regular point})$$

PROOF: Since $x \in FLP$, x is the limit of some subsequence, $x^r \to x$.
Let $x^r = (q^r, a^r, s^r, z^r)$, $x = (q, a, s, z)$.
Assume t is a regular point, and suppose $\pi \cdot q(t) < \max_{\rho \in S} \rho \cdot q(t)$.

Since $q(\cdot)$ is absolutely continuous, there is some interval $[t, t+\delta]$ and error $\eta > 0$ such that $\pi \cdot q(s) < \max_{\rho \in S} \rho \cdot q(s) - \eta$ for $s \in [t, t+\delta]$
We know $q^r \to q$, hence $\pi \cdot \frac{Q(rs)}{r} < \max_{\rho \in S} \rho \cdot \frac{Q(rs)}{r}$ for $s \in [t, t+\delta]$.
Therefore, π is never chosen (in the original discrete system).
Therefore $s_\pi(t+\delta) = s_\pi(t)$.
Therefore $\dot{s}_\pi(t) = 0$.

The general approach is:

* make up some plausible fluid model equations
* prove that all fluid limit points (FLPs) satisfy your equations
* deduce that the fluid-scaled system lies close to some fluid model solution (FMS)

* prove that all fluid model solutions have some particular characteristic
* deduce that the fluid-scaled system has (nearly) that characteristic
* rewrite in terms of the original (unscaled) system.

The general approach is:

could be anything, not just eqns for derivatives

* make up some plausible fluid model equations
* prove that all fluid limit points (FLPs) satisfy your equations
* deduce that the fluid-scaled system lies close to some fluid model solution (FMS)
* prove that all fluid model solutions have some particular characteristic

tradeoff of effort

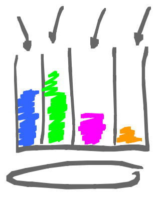
If there is not a unique FMS, maybe you need to obtain an extra fluid model eqn, or maybe the fluid-scaled system really is random.

Questions about fluid models

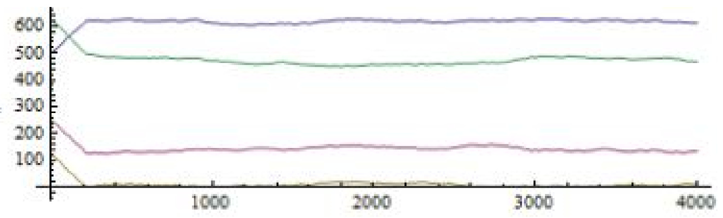
Q1. Is the service rate $\sigma(t) = \sum_\pi \pi \dot{s}_\pi(t)$ uniquely determined by $q(t)$?

Q2. Is there a unique solution for $q(\cdot)$?

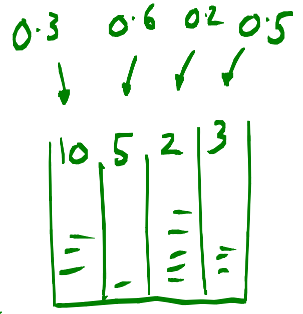
Q3. Are the fluid model solutions for MaxWeight always piecewise-linear, with finitely many pieces?



$S = \{(1,0,0,0), (0,1,1,1)\}$.



A1.



$\mathcal{S} = \{(1,0,0,0),$
$(0,1,1,1)\}$:

$(1,0,0,0)\cdot Q = (0,1,1,1)\cdot Q$.

So both actions have the same weight.

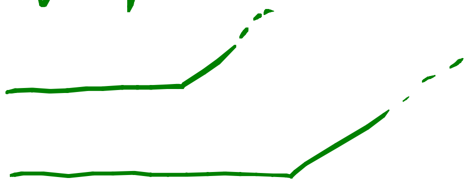
Our EMEq says just "$\dot{s}_\pi(t) = 0$ if $\pi \cdot q < \max_p p\cdot q$"

ie "$\sigma(t) \in \arg\max_{p \in \langle S \rangle} p\cdot q$".

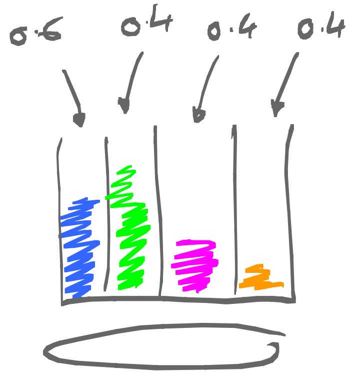
Therefore any linear combination $(1-a, a, a, a)$ is max-weight.

On the other hand, if we set $a=0$, then q_0 drops while the others increase, so we're straight away forced to switch to $a=1$. So maybe there is a unique choice of a.

A2. Consider e.g. $\frac{d}{dt} q(t) = \begin{cases} 0 & \text{if } q(t)=0 \\ 1 & \text{if } q(t)>0 \end{cases}$. Here, $\frac{d}{dt} q(t)$ is uniquely determined by $q(t)$ — but there are many solutions.



The limits of fluid limits

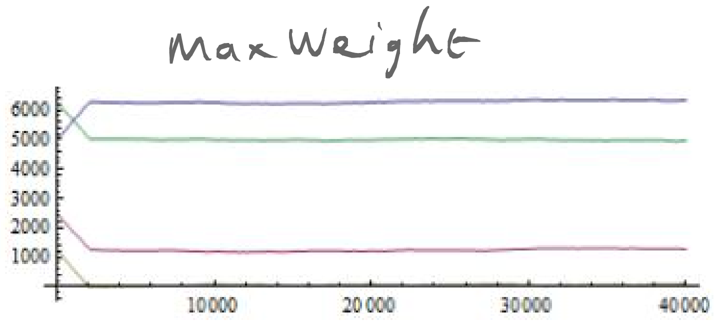


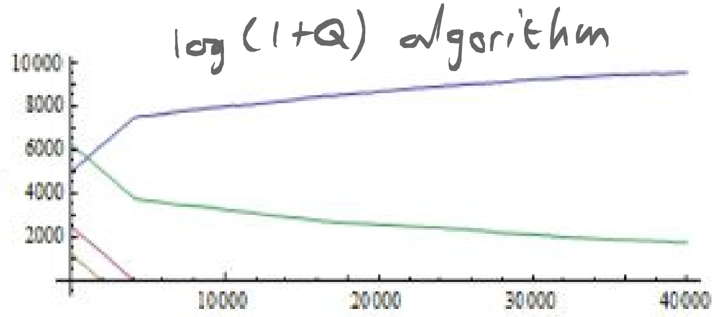
$S = \{(1,0,0,0),\ (0,1,1,1)\}.$

suppose we use the algorithm:

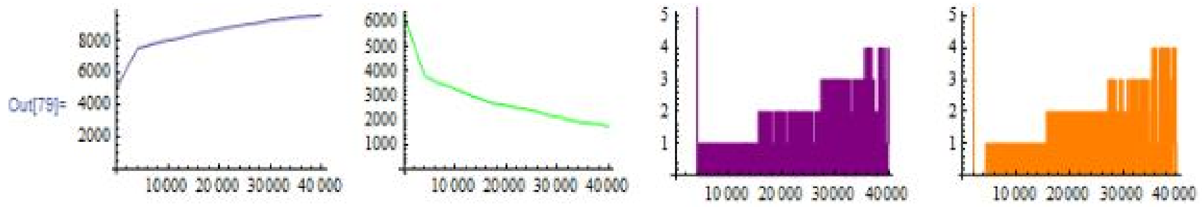
if $\log(1+Q_0) < \log(1+Q_1) + \log(1+Q_2) + \log(1+Q_3)$,
then use $(0,1,1,1)$ else use $(1,0,0,0)$.

What is an appropriate fluid model equation?





rescaled queue sizes for log(1+Q) algorithm



Guess :

"if $\log(1+q_0) > \log(1+q_1) + \log(1+q_2) + \log(1+q_3)$

then $\dot{s}_{(0111)} = 0$".

Recall the proof for MaxWeight.

Since $q(\cdot)$ is absolutely continuous, there is some interval $[t, t+\delta]$
and error $\eta > 0$ such that $\pi \cdot q(s) < \max_{\rho \in S} \rho \cdot q(s) - \eta$ for $s \in [t, t+\delta]$
We know $q^r \to q$, hence $\pi \cdot \frac{Q(rs)}{r} < \max_{\rho \in S} \rho \cdot \frac{Q(rs)}{r}$ for $s \in [t, t+\delta]$. doesn't work for $\log(1+Q)$ alg.
Therefore, π is not chosen throughout this interval.

Suppose e.g. $Q = (100, 10, 10, 10)$ and $r = 10$.

$\rightarrow q^r = (10, 1, 1, 1)$.

$$\log(1+q_0^r) > \log(1+q_1^r) + \log(1+q_2^r) + \log(1+q_3^r)$$

$$\log 11 > \log 8.$$

$$\log(1+Q_0) \qquad \log(1+Q_1) + \log(1+Q_2) + \log(1+Q_3)$$

$$\log 101 < \log(11^3).$$

The alg tries to balance: $\log(1+Q_0) \approx \log(1+Q_1) + \log(1+Q_2) + \log(1+Q_3)$.

If $Q_0 \approx r q_0$, maybe the rest are $r^{\frac{1}{3}}, r^{\frac{1}{3}}, r^{\frac{1}{3}}$ or $r^{\frac{1}{2}}, r^{\frac{1}{4}}, r^{\frac{1}{4}}$, or ....

Summary

Most modern theory about switched networks uses fluid models as a starting point.

Fluid-scaled versions of the system dynamics converge to fluid limit points ;
we can prove that these limit points obey certain fluid model equations.

We can characterize the behaviour of fluid model solutions
i.e. solutions to the fluid model equations.
This tells us something about the original system dynamics.

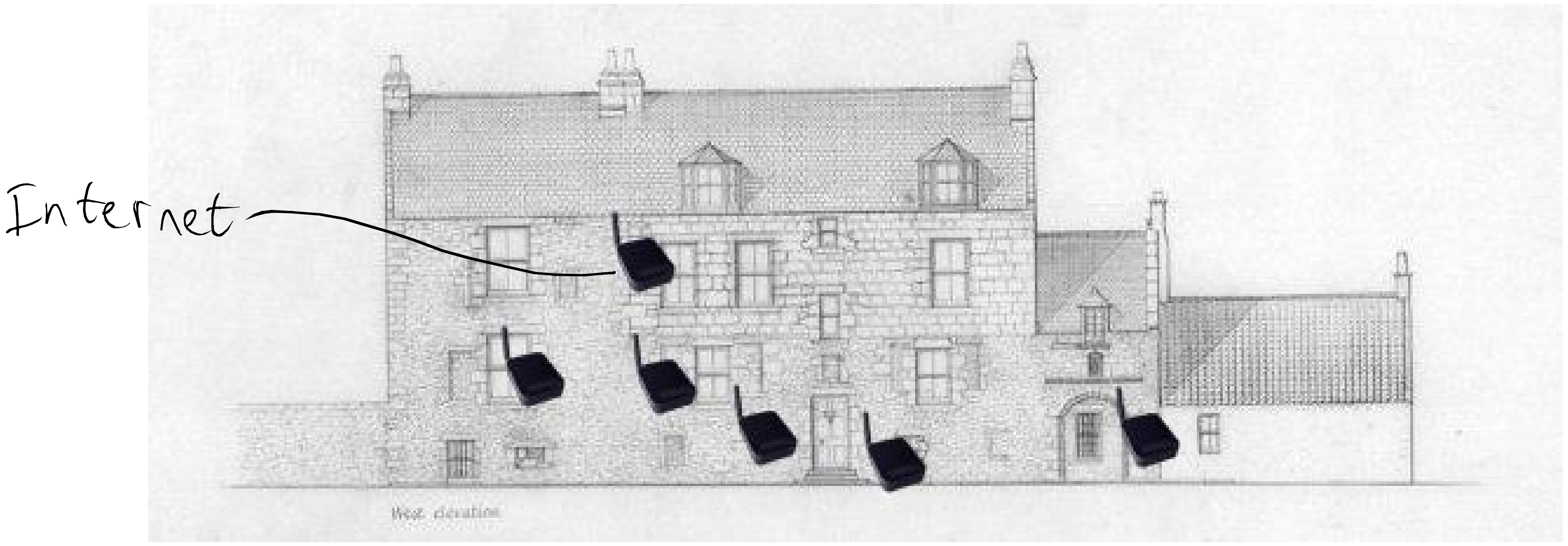
We only know sensible fluid models for simple algorithms like MaxWeight and the TCP family.

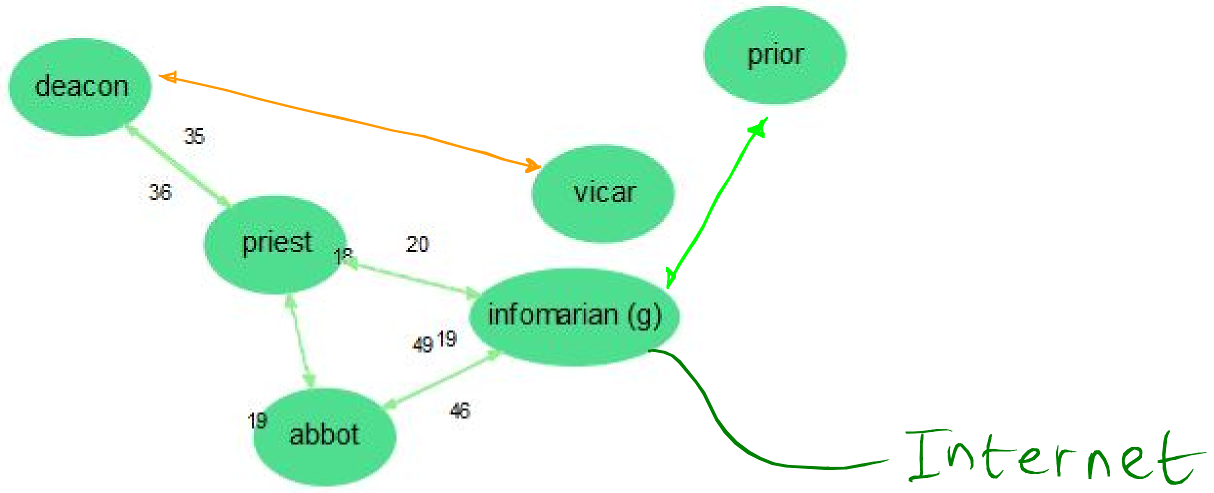
Performance analysis of Switched Networks

How to use fluid models and Lyapunov functions to understand underload, critical load, and overload.

And why it all comes down to two optimization problems.

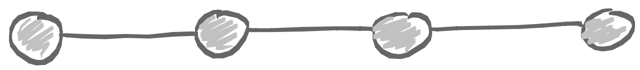
An example wireless network



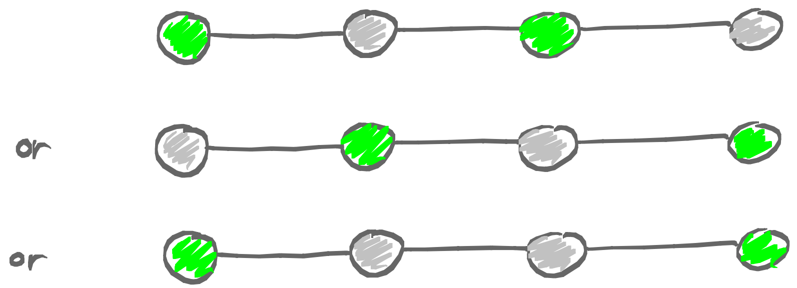


A simpler example wireless network

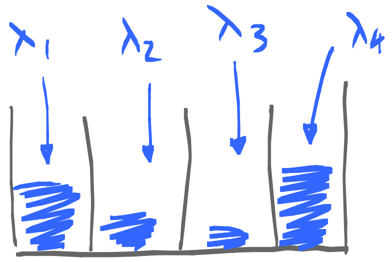
[Ji, Athanasopoulo, Srikant 2009]



4 wireless nodes, some of which interfere, operating in slotted time



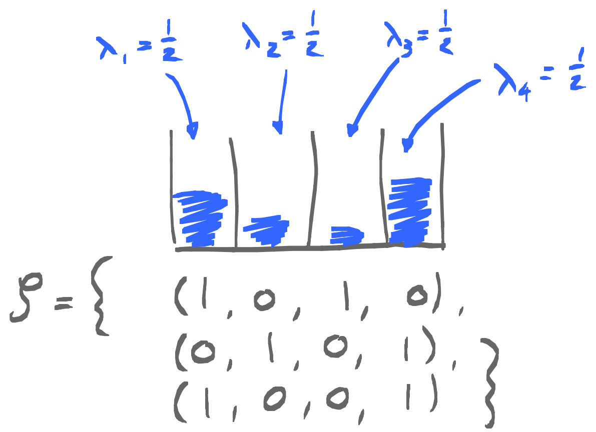
In each timeslot, one of three possible transmission patterns is chosen. (Assume a centralized scheduler makes the choice.)



$$\rho = \{ (1,0,1,0), (0,1,0,1), (1,0,0,1) \}$$

There is a constant-rate flow of packets to each node. Packets are queued, then transmitted.

What scheduling algorithm should I run?



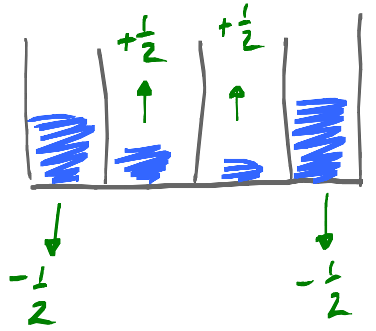
$\lambda_1 = \frac{1}{2}$
$\lambda_2 = \frac{1}{2}$
$\lambda_3 = \frac{1}{2}$
$\lambda_4 = \frac{1}{2}$

$$\mathcal{S} = \{ (1,0,1,0), (0,1,0,1), (1,0,0,1) \}$$

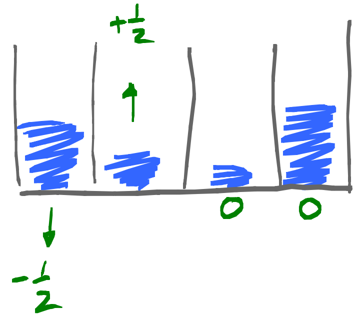
The fluid model says: $\dot{q}_r(t) = [\lambda_r - \sigma_r(t)]^+$ for some $\sigma(t) \in \langle \mathcal{S} \rangle$.

Here are some possible choices for $\sigma(t)$, and the consequent $\dot{q}(t)$:

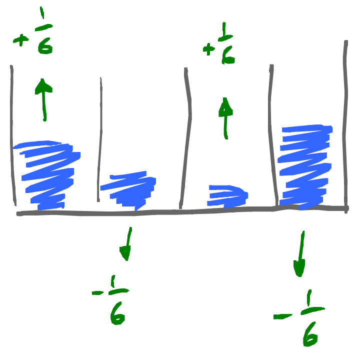
$\sigma = (1,0,0,1)$



$\sigma = \frac{1}{2}(1,0,0,1) + \frac{1}{2}(1,0,1,0)$

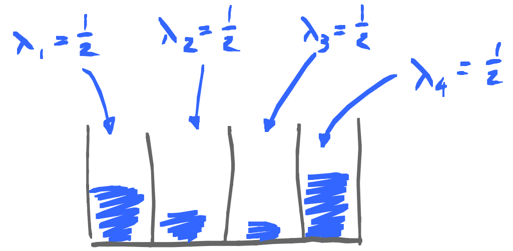


$\sigma = \frac{1}{3}(1,0,1,0) + \frac{2}{3}(0,1,0,1)$



Which of these is best? Misleading hint: think of MaxWeight.

Critical Workloads



$\lambda_1 = \frac{1}{2}$
$\lambda_2 = \frac{1}{2}$
$\lambda_3 = \frac{1}{2}$
$\lambda_4 = \frac{1}{2}$

$$\mathcal{S} = \left\{ \begin{array}{l} (1, 0, 1, 0), \\ (0, 1, 0, 1), \\ (1, 0, 0, 1) \end{array} \right\}$$

Observe that for every possible service rate $\sigma(t) \in \langle \mathcal{S} \rangle$,

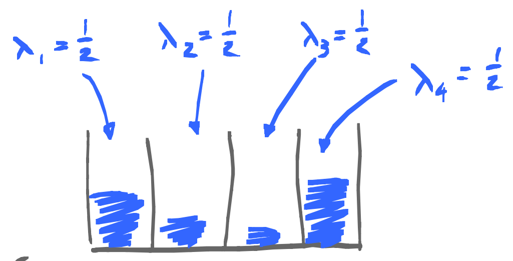
$$\left.\begin{array}{l} \dot{q}_1(t) + \dot{q}_2(t) \geq 0 \\ \dot{q}_3(t) + \dot{q}_4(t) \geq 0 \end{array}\right\} \text{with equality if the queues aren't empty}$$

$$\dot{q}_2(t) + \dot{q}_3(t) \geq 0 \qquad \text{with strict inequality if } (1,0,0,1) \text{ is ever chosen.}$$

It's a waste — an irreversible action — to pick $(1\ 0\ 0\ 1)$.

Though note that if the arrival rates were different, e.g. $(\frac{2}{3}\ \frac{1}{3}\ \frac{1}{3}\ \frac{2}{3})$, then the last inequality wouldn't hold, and it might be useful to pick $(1\ 0\ 0\ 1)$.

Critical Workloads



$\lambda_1 = \frac{1}{2}$
$\lambda_2 = \frac{1}{2}$
$\lambda_3 = \frac{1}{2}$
$\lambda_4 = \frac{1}{2}$

$$S = \left\{ \begin{array}{l} (1,0,1,0), \\ (0,1,0,1), \\ (1,0,0,1) \end{array} \right\}$$

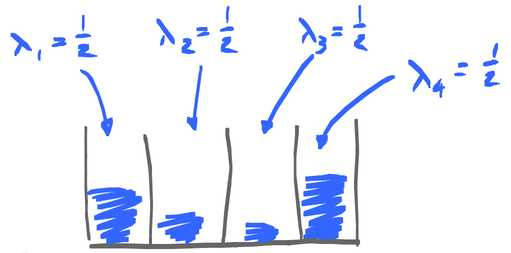
Observe that for every possible service rate $\sigma(t) \in \langle S \rangle$,

$$\left.\begin{array}{l} \dot q_1(t) + \dot q_2(t) \geq 0 \\ \dot q_3(t) + \dot q_4(t) \geq 0 \end{array}\right\} \text{with equality if the queues aren't empty}$$

$$\dot q_2(t) + \dot q_3(t) \geq 0 \quad \text{with strict inequality if } (1,0,0,1) \text{ is ever chosen.}$$

The scheduling algorithm can shift work between queues by choosing $\sigma(t)$ appropriately.
But there are certain linear combinations of queue sizes that are constrained by S.
These are known as <u>virtual resources</u> or <u>workloads</u>.
They depend on λ as well as S.

Critical Workloads



$\lambda_1 = \frac{1}{2}$
$\lambda_2 = \frac{1}{2}$
$\lambda_3 = \frac{1}{2}$
$\lambda_4 = \frac{1}{2}$

$$S = \left\{ \begin{array}{l} (1,0,1,0), \\ (0,1,0,1), \\ (1,0,0,1) \end{array} \right\}$$

Observe that for every possible service rate $\sigma(t) \in \langle S \rangle$,

$$\left.\begin{array}{l} \dot{q}_1(t) + \dot{q}_2(t) \geqslant 0 \\ \dot{q}_3(t) + \dot{q}_4(t) \geqslant 0 \end{array}\right\} \text{with equality if the queues aren't empty}$$

$$\dot{q}_2(t) + \dot{q}_3(t) \geqslant 0 \quad \text{with strict inequality if } (1,0,0,1) \text{ is ever chosen.}$$

"Lord grant me the serenity to accept the things I cannot change,
the courage to change the things I can,
and the wisdom to know the difference."

The static planning problem & dual

Consider the problem:

Find a mix of service actions $\sigma \in \langle S \rangle$ to keep the system stable

SPLAN(λ): minimize $\sum_{\pi \in S} a_\pi$ over $a \in \mathbb{R}_+^{|S|}$

such that $\lambda \leq \sum_{\pi \in S} a_\pi \pi$.

If SPLAN(λ) < 1: underload
If SPLAN(λ) $= 1$: critical load
If SPLAN(λ) > 1: overload.

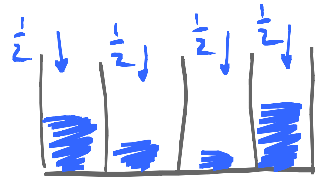
The dual is

SDUAL(λ): maximize $\xi \cdot \lambda$ over $\xi \in \mathbb{R}_+^N$

such that $\xi \cdot \pi \leq 1$ for all $\pi \in S$.

The dual problem tells us virtual resources.

[Laws 1992, Kelly+Laws 1993, Harrison 2000]



$\mathcal{S} = \{ (1,0,1,0), (0,1,0,1), (1,0,0,1) \}$

SDUAL(λ):

maximize $\xi \cdot \lambda$ over $\xi \in \mathbb{R}_+^N$

such that $\xi \cdot \pi \leq 1$ for all $\pi \in \mathcal{S}$.

$\Xi = \{ (1,1,0,0), (0,1,1,0), (0,0,1,1) \}$.

The feasible region does not depend on λ.
It is a convex polytope.
Let Ξ be the maximal extreme points.

$\dot{q}_1 + \dot{q}_2 \geq 0$

$\dot{q}_2 + \dot{q}_3 \geq 0$

$\dot{q}_3 + \dot{q}_4 \geq 0$

Then $\dot{q}(t) = [\lambda - \sigma(t)]^+ \geq \lambda - \sigma(t)$

$\Rightarrow \quad \dot{q}(t) \cdot \xi \geq \lambda \cdot \xi - 1 \qquad$ for all $\xi \in \Xi$.

The teleology of switched networks.

The scheduling algorithms we have considered
(MaxWeight family and TCP family)
can be viewed as "steepest descent of Lyapunov function"

The switched network constraint, $\sigma(t) \in \mathcal{S}$,
entails constraints on the workloads.

The teleology of switched networks.

The scheduling algorithms we have considered (MaxWeight family and TCP family) can be viewed as "steepest descent of Lyapunov function"

The switched network constraint, $\sigma(t) \in \mathcal{S}$, entails constraints on the workloads.

This prompts us to consider the optimization problem

$\mathrm{TELE}(\lambda, q(0), t)$: minimize $L(q')$ over $q' \in \mathbb{R}_+^N$
such that $q' \cdot \xi \geq q(0) \cdot \xi + (\lambda \cdot \xi - 1)t$ for all $\xi \in \Xi$.

This problem is deeply linked to performance analysis.

Critical Load i.e. $\max_{\xi \in \Xi} \xi \cdot \lambda = 1$.

The teleology problem, at $t=\infty$, is

TELE$(\lambda, q(0), \infty)$: minimize $L(q')$ over $q' \in \mathbb{R}_+^N$
such that $q' \cdot \xi \geqslant q(0) \cdot \xi$ for all $\xi \in \Xi$ such that $\xi \cdot \lambda = 1$

All fluid model solutions (nearly, eventually) solve this problem.
Therefore the original stochastic system also (nearly) solves it.
This perhaps leads to a reflected Brownian motion limit process.

Critical Load i.e. $\max_{\xi \in \Xi} \xi \cdot \lambda = 1$.

The teleology problem, at $t=\infty$, is

TELE$(\lambda, q(0), \infty)$: minimize $L(q')$ over $q' \in \mathbb{R}_+^N$
such that $q' \cdot \xi \geq q(0) \cdot \xi$ for all $\xi \in \Xi$ such that $\xi \cdot \lambda = 1$

All fluid model solutions (nearly, eventually) solve this problem.

THEOREM. This problem has a unique solution, call it $\Delta W(q(0))$.
Let $\text{INV} = \{ q : q \text{ solves TELE}(\lambda, q, \infty) \}$.
All fluid model solutions satisfy the following:
q is a fixed point $\Leftrightarrow$ $q \in \text{INV}$, and $d(q(t), \text{INV}) \to 0$.

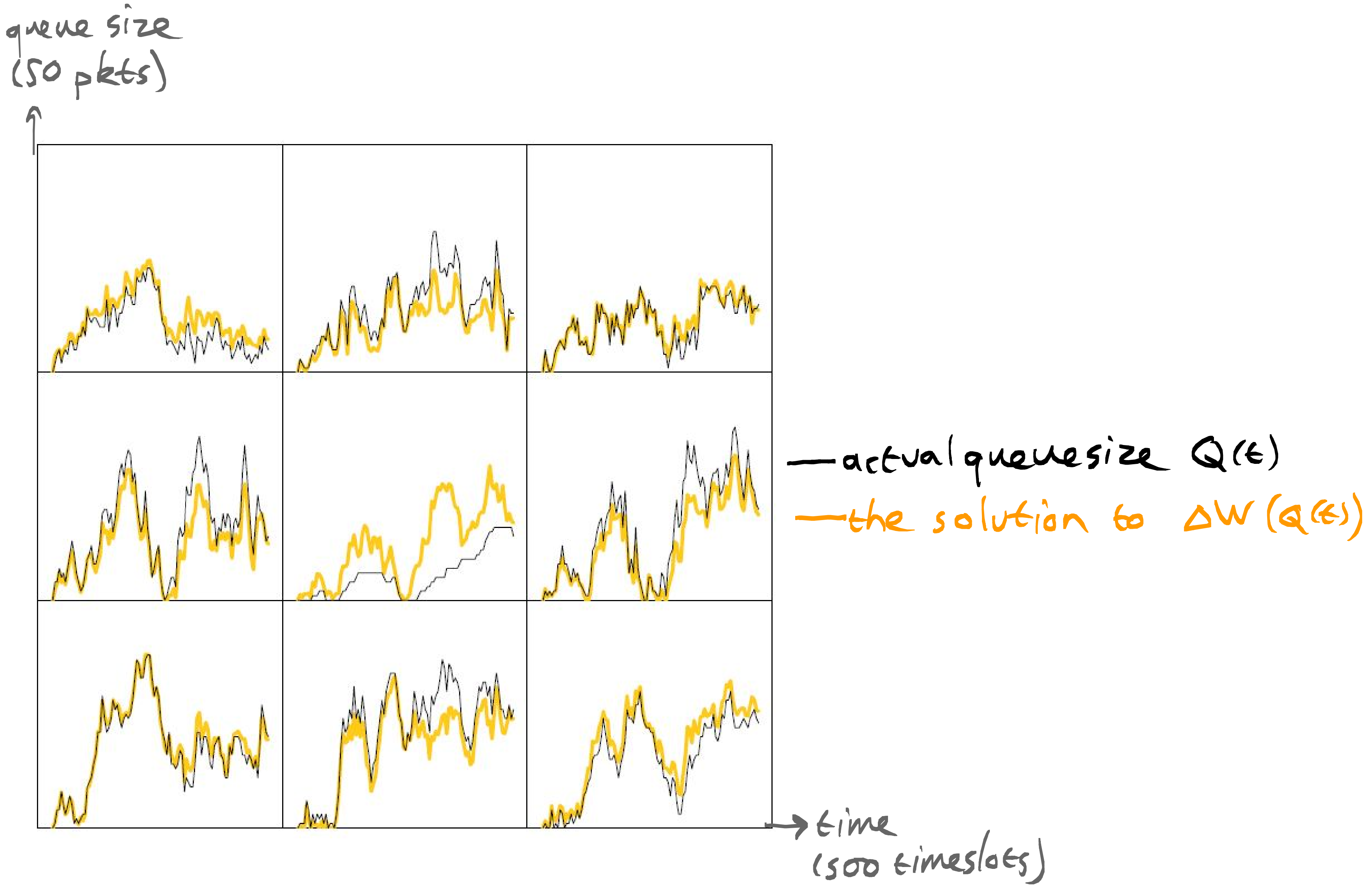
Therefore the original stochastic system also (nearly) solves it.

THEOREM. $\mathbb{P}\left(\sup_{t \in [0, r^2]} \left| Q(t) - \Delta W(Q(t)) \right| < \delta r \vee \delta \sup_{t \in [0,r^2]} |Q(t)| \right) \to 1$

This perhaps leads to a reflected Brownian motion limit process.

Guess: the bigger INV, the smaller the eqm. distribution of queue size.

Illustration:
An input-queued switch with $N=9$ queues,
$\mathcal{S} = \{3\times3 \text{ permutation matrices}\}$



WHEN EXACTLY ONE VIRTUAL RESOURCE IS CRITICAL

Stolyar, 2004 : RBM for MaxWeight, slotted time model,
In this setting, any alg. in the MaxWeight family is optimal.

This is a bad result, because it misses some differences which are visible in simulations. [McKeown + Keslassy 2000]

Dai + Lin, 2005, 2008 : same, for a continuous-time model (a stochastic processing network)

WHEN THERE ARE MULTIPLE BOTTLENECKS

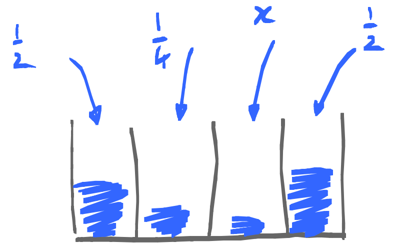
Kelly + Williams, 2005 Bandwidth-sharing model, fluid limit result.
Kang, Kelly, Lee, Williams, 2009 and RBM when $\alpha = 1$.

I + Athanasopoulou + Srikant 2009. MaxWeight is not optimal for this 4-queue example.

Shah + W. Mult. state space collapse for MaxWeight, slotted time

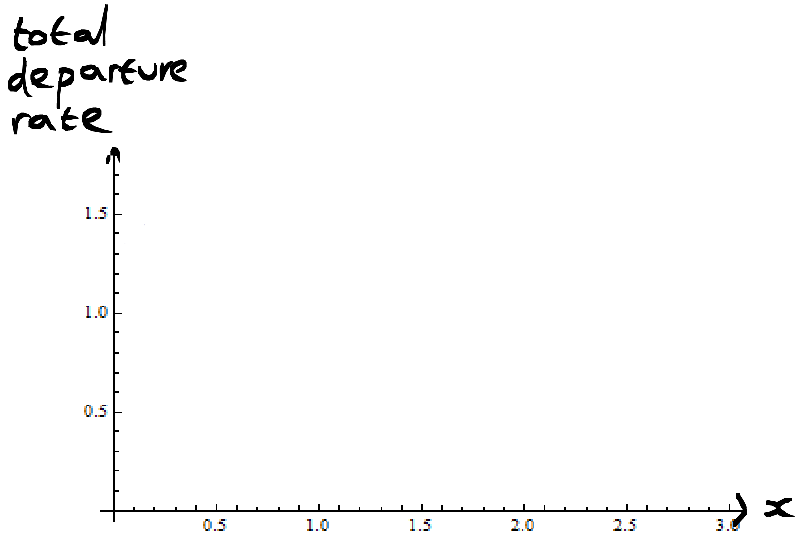
Overload i.e $\max_{\xi \in \Xi} \lambda \cdot \xi > 1$

In overload, it is inevitable that some workloads will grow. The scheduling algorithm can nonetheless choose in which queues this workload is stored. (or, if there is reneging, which customers to put off.)



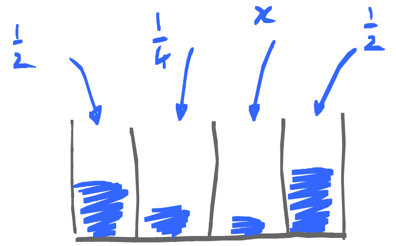
$$\rho = \{ (1,0,1,0), (0,1,0,1), (1,0,0,1) \}$$

$$\Xi = \{ (1,1,0,0), (0,1,1,0), (0,0,1,1) \}.$$



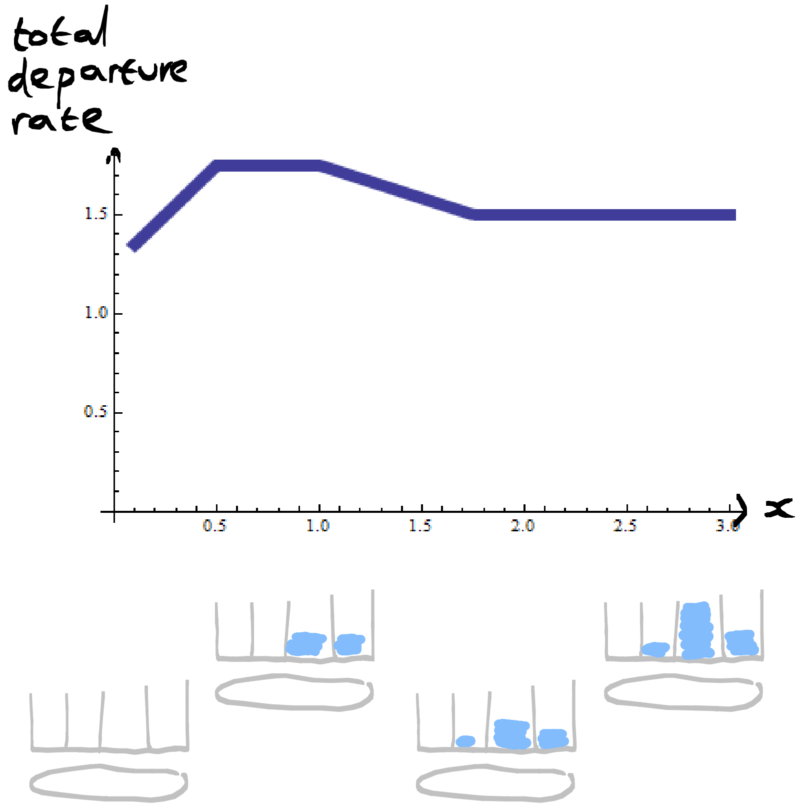
Overload i.e $\max_{\xi \in \Xi} \lambda \cdot \xi > 1$

In overload, it is inevitable that some workloads will grow.
The scheduling algorithm can nonetheless choose in which queues this workload is stored. (or, if there is reneging, which customers to put off.)



$$\rho = \{ (1,0,1,0), (0,1,0,1), (1,0,0,1) \}$$

$$\Xi = \{ (1,1,0,0), (0,1,1,0), (0,0,1,1) \}.$$



Overload i.e $\max_{\xi \in \Xi} \lambda \cdot \xi > 1$

The workload constraints grow linearly in t. However, the rescaled teleology problem $\frac{1}{t}\,\mathrm{TELE}(\lambda, q(0), t)$ is interesting as $t \to \infty$:

minimize $L(q')$ over $q' \in \mathbb{R}_+^N$
such that $q' \cdot \xi \geq \lambda \cdot \xi - 1$ for all $\xi \in \Xi$ such that $\xi \cdot \lambda > 1$
and $q' \leq \lambda$.

Queue sizes grow linearly, and the growth rates are given by this problem.

THEOREM. If $q(0) = 0$ then the unique fluid model solution is $q(t) = t q'$, where q' is the unique solution to the above. Otherwise, $\frac{q(t)}{t} \to q'$.

Overload i.e $\max_{\xi \in \Xi} \lambda \cdot \xi > 1$

The workload constraints grow linearly in t. However, the rescaled teleology problem $\frac{1}{t}\,\mathrm{TELE}(\lambda, q(0), t)$ is interesting as $t \to \infty$:

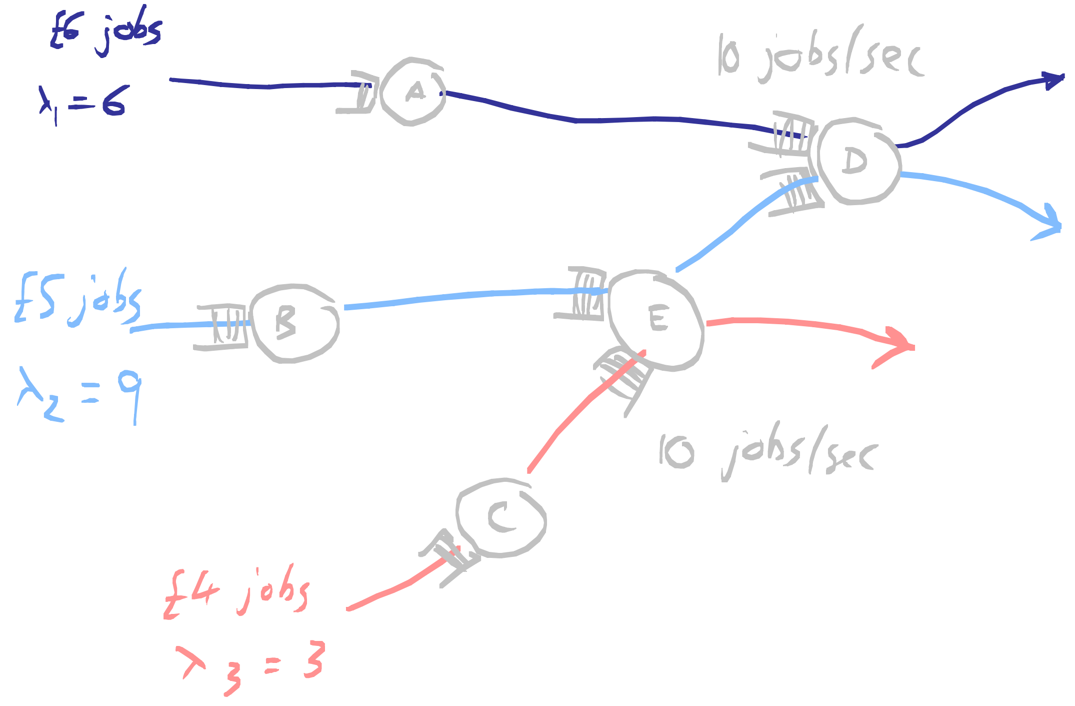
minimize $L(q')$ over $q' \in \mathbb{R}_+^N$
such that $q' \cdot \xi \geq \lambda \cdot \xi - 1$ for all $\xi \in \Xi$ such that $\xi \cdot \lambda > 1$
and $q' \leq \lambda$.

Queue sizes grow linearly, and the growth rates are given by this problem.

Egorova + Borst + Zwart 2007: bandwidth-sharing model, general file size dist. existence, and uniqueness for special networks

Shah + W.: bandwidth-sharing with exp. file sizes and $\alpha \geq 1$, and MaxWeight. uniqueness.

Application: admission control for data centres



When the data centre is overloaded, requests have to be discarded. Is there a simple adaptive scheme that learns it should discard 5/sec of the £5 jobs?

Answer: run MaxWeight with $L(q) = \sum_n q_n^{1+\alpha} \times$ cost of jobs in queue n; Let α be small. Then $L(q)$ is ≈ the rate at which revenue is lost, $L(q) \approx \sum_n q_n \times$ cost of jobs in queue n, and MaxWeight minimizes this over all feasible service rates.

Underload i.e. $\max_{\xi \in \Xi} \lambda \cdot \xi < 1$.

We already know that, in underload, the fluid model is stable.

$\frac{d}{dt} L(q(t)) < 0$ unless $q(t) = 0$, and $\exists H > 0$ s.t. $q(t) = 0$ for $t \geq H |q(0)|$.

Furthermore, $\frac{Q(rT)}{r}$ satisfies a large deviations principle [Subramanian].

The rate function for $L\left(\frac{Q(rT)}{r}\right)$ is based on the scaled TELE problem. [Venkataramanan + Lin, 2007, 2009]

$$\frac{1}{r} \log \mathbb{P}\left(L\left(Q(rT)\right) \geq rx \right) \approx - \inf_{\mu : \max_{\xi \in \Xi} \xi \cdot \mu > 1} \frac{\ell(\mu)}{RTELE(\mu)}$$

local rate function for arrival process

overload opt. problem.

The most likely path to overflow is:
"Pick some overload arrival rate μ, then run it long enough to reach $L(\cdot) = rx$". The overload analysis tells us how quickly the queues grow.

Summary

Mon switched network models cover a range
of problems to do with tradeoffs in resource allocation.

The MaxWeight and TCP-like scheduling algorithms
Mon "try" to minimize a Lyapunov function,
Fri subject to workload constraints
They are appealing because they are adaptive/online,
and (sometimes) lend themselves to distributed implementation.

This lets us characterize fluid model solutions,
Wed from which we can make inferences about stochastic systems.

There seem to be close links between
Fri overload $\xrightarrow{\text{most likely path}}$ underload $\xleftrightarrow{\text{moderate deviations}}$ critical load

Q. Other scheduling algorithms & objectives?

↳ What is optimal, for the 4-queue example?
↳ what is the cost of e.g. giving some queues priority?
↳ what is an algorithm that is simpler than MaxWeight but (nearly) as good?

Q. Other model features?

↳ Stickiness, i.e. cost of switching from one action to another?
↳ Continuous-time version of MaxWeight?
↳ Head-of-line blocking, and similar "measure-valued" issues?

Q. Deep queueing theory links.

↳ In what sense are overload, critical load & underload all describing the same phenomena?