STOCHASTIC NETWORKS
PART III
LENT 2011

FRANK KELLY
STATISTICAL LABORATORY, UNIVERSITY OF CAMBRIDGE
F.P.KELLY@STATSLAB.CAM.AC.UK
TYPESET BY ELENA YUDOVINA
STATISTICAL LABORATORY, UNIVERSITY OF CAMBRIDGE
E.YUDOVINA@STATSLAB.CAM.AC.UK

1. Introduction

This course is about the way microscopic laws governing behaviour of individual elements of the system can give rise to macroscopic behaviour in the system. One example of this is the physics of gases: the air in this room is made up of molecules, and you can think of it in terms of molecular movement; however, you can also think about it in terms of Boyle's Law (relating pressure and volume of a gas). In this course we will mostly consider constructed systems, where if we understand the connection between the microscopic and the macroscopic behaviour, we might be able to change the microscopic laws.

Course website: `http://www.statslab.cam.ac.uk/~frank/STOCHNET/`

Prerequisites:

- Basic optimization: at least at the level of Lagrange multipliers. (See Sections 1, 2.1 of Richard Weber's notes at `http://www.statslab.cam.ac.uk/~rrw1/mor/s.pdf`.)
- Markov Chains: at least the discrete-time theory, although continuous-time will be helpful. (See the book James Norris, *Markov Chains*, CUP 1998, `http://www.statslab.cam.ac.uk/~james/Markov/`, sections 1.7 and 2.4 especially; and/or Chapter 1 of Bruce Hajek's notes `http://www.ifp.illinois.edu/~hajek/Papers/networkanalysis.html`.)

Topics we will cover:

Queueing networks:

We will first look at a single queue. We will see how to model it as a Markov chain,

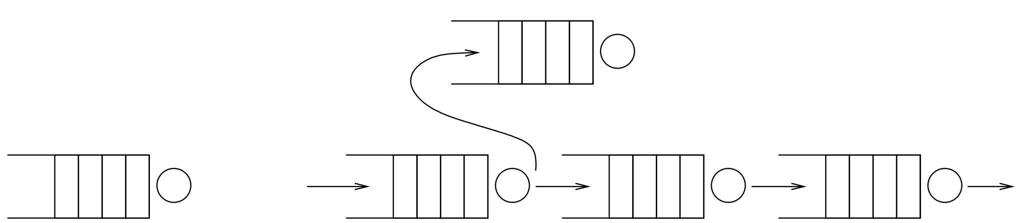


FIGURE 1. A single queue, and a network of several queues.

and derive information on the distribution of the queue size. We will then pass on

Date: Lent 2011.

to a network of queues. Starting from a simplified description of each queue, we will get system behaviour.

Fundamental questions: how can we define traffic intensity? Can we identify Poisson flows in the system?

Loss networks:

The links between the nodes have several "circuits", and when a "call" is placed

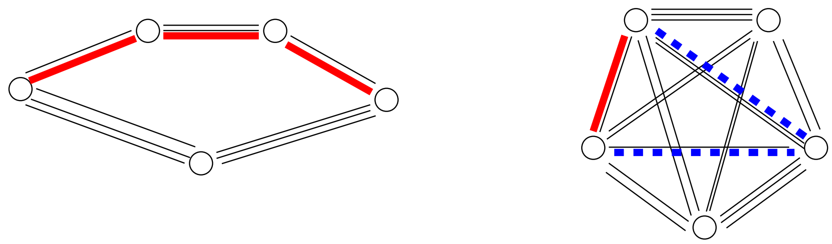


FIGURE 2. A loss network. A loss network with a complete graph.

between one node and another, it needs to simultaneously get a circuit on each of the links – otherwise, the call is lost.

Queueing networks	Loss networks
sequential use of resources congestion $\Longrightarrow$ delay	simultaneous resource possession congestion $\Longrightarrow$ loss

If the switchboards are computerised, we can afford to have dynamic routing policy, e.g. the following: in the complete graph network, if a call arrives and there is a circuit available on the direct link between the two nodes, then the call is scheduled along the direct link. Otherwise, we attempt to redirect the call via another node (chosen at random), on a two-circuit path. (These are the solid red and the dotted blue options in Figure 2.)

What is the loss probability in such a circuit, as a function of the arrival rates? Note that this is impossible! The network is an irreducible finite-state Markov chain,

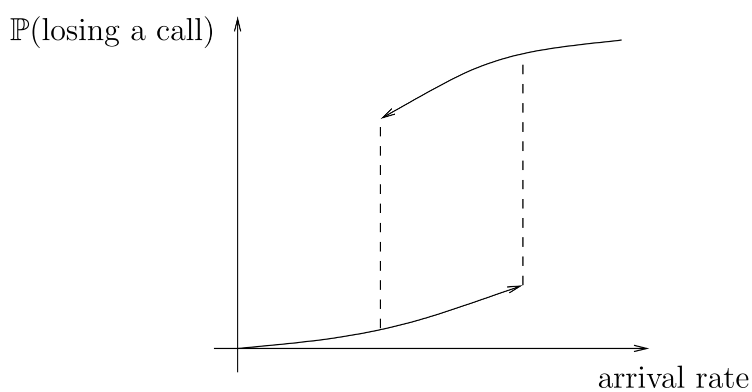


FIGURE 3. Experimental probability of losing a call: hysteresis

so it must have a *unique* stationary distribution, hence *unique* probability of losing a call at a given arrival rate.

On the other hand, if the utilisation (i.e. proportion of occupied circuits) is low, calls are likely to be scheduled along direct links; if the utilisation is high, they are more likely to be scheduled along indirect routes, causing positive feedback.

How can both of these insights be true? They concern two different scaling regimes:

(1) If we leave the arrival rate constant and simulate the network for a long enough time (e.g., the lifetime of the universe), then the average proportion of lost calls will indeed converge to the stationary probability coming from the unique stationary distribution.
(2) If, however, we fix a time period $[0, T]$, fix the per-link arrival rate of calls, and let the number of links tend to infinity, then the time to go from one branch of the graph above to the other will tend to ∞, i.e. system will "freeze" in one of the two states (low-utilisation or high-utilisation).

If you're familiar with the Ising model, the same phenomenon occurs there. The Ising model considers particles located at lattice points with states ± 1, which are allowed to switch states, and the particle is more likely to switch to a state where it agrees with the majority of its neighbours. Clearly, the unique stationary distribution of any finite-sized system is symmetric with mean 0; on the other hand, if we look over a finite time horizon at ever-larger squares, they will "freeze" either in the state where almost all particles are $+1$ or almost all particles are -1.

The fundamental questions for the loss network model are the blocking probability of the system, and the stability (or instability) of it ("congestion collapse").

Multiple access schemes:

The fundamental issue here is contention resolution. Suppose that we have multiple

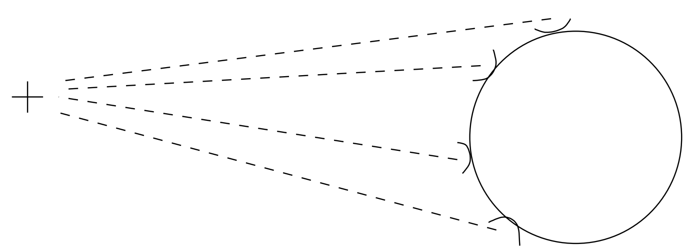


Figure 4. Multiple base stations contacting a satellite

base stations, which cannot talk to each other directly (because the Earth is in the way) and must do so through a satellite. That is, if one of them needs to pass a message, it sends it to the satellite, which then broadcasts the message to all of them (the address is part of the message, so it gets decoded by the correct station at the end). The issue is that if two stations try to transmit a message simultaneously, it gets killed by the interference. How can we avoid this issue?

(1) We could divide time into slots, and assign every fourth slot to each of the stations. However, this only works if each of the stations has at most a quarter of the traffic (and only if we know there are four of them!)
(2) We could implement a token ring: set up an order between the stations, and have the last thing that a station transmits be a "token" which means that it is done transmitting. Then the next station is allowed to start transmitting (or it may simply pass on the token). However, if there is a large number of stations, then simply passing on the token around all of them will take a very long time!
(3) The ALOHA protocol: listen to the channel; if nobody else is transmitting, just start your own transmission. Since it takes some time for messages to reach the satellite and be broadcast back, it's possible that there will be collisions

(if two stations decide to start transmitting at sufficiently close times). If this happens, stop transmitting and wait for a *random* amount of time before trying to retransmit.

(4) The Ethernet protocol: if you are retransmitting the message for the kth time, the expected waiting time above will have mean 2^k.

The last two protocols should work well if there are many stations but low probability of collision.

The fundamental issue here is the speed-of-light delay in the network, which is becoming a problem on smaller and smaller scales as computational power increases. An algorithm of this sort is used to allocate channels to mobile phone calls, and lately even for communication between chips.

Broadband networks: Suppose we have a communication link of total capacity C, which is being shared by lots of users, each of which wants some rate as a function of time. We are interested in controlling the probability that traffic gets lost (i.e. that the sum of the demanded rates exceeds capacity). How can we do this?

We will consider the problem in the "large deviations" limiting regime.

Internet modelling: What happens when I want to see a web page? I send a request to the server, who then starts sending the page to me as a stream of packets through the large network of the internet. The server sends one packet at first; I send an acknowledgement packet (ACK) back; when the server receives the ACK, it sends me 2 packets, and we repeat the process with each of those. When the server doesn't receive an ACK for one of the packets (i.e., when a packet is dropped), it will decrease the rate at which the packets are sent to me. (The reason packets get dropped is that in the network that is the Internet they have to go through finite-size buffers, which overflow.) This is known as Transmission Control Protocol, or TCP, and is implemented only at the endpoints (me and the server) – the insides of the network are essentially brainless.

We would like to know how control loops like this interact in the network. Will the network be stable? (What does that mean?) How will the network end up sharing the resources between the users?

Note that in this system the response to congestion is to decrease the rate of service. In this sense, it behaves like a processor-sharing queue – congestion means that it will take longer for the job to finish. Will the network in fact look like a single queue of some kind?

Transport networks; electricity networks: We will say something about these too, but in the mean time let's get started.

1.1. **Markov Chains.** See James Norris, *Markov Chains*. Cambridge University Press, 1998.

Let S be a countable state space.

Definition. $(X(t), t \in \mathbb{R})$ is a Markov chain (sometimes called Markov process)[1] with state space S and transition rates $q(j,k)$ if, for $j,k \in S$, we have

$$\mathbb{P}[X(t+\delta t) = k | X(t) = j] = q(j,k)\delta t + o(\delta t).$$

We may also write X_t for $X(t)$.

In this course, we will assume that all chains are conservative, irreducible, and non-explosive. *Conservative* means that the process cannot leave S – that is, $\sum_{j\in S} \mathbb{P}[X(t) = j] = 1$. *Irreducible* means that it is possible to get from any state to any other, possibly by a (finite) path through the other states. *Explosive* means that a chain makes infinitely many transitions in a finite time period – e.g. if the time to go $1 \to 2$ were $1/2$, the time to go $2 \to 3$ were $1/4$, and generally the time to go $i \to i+1$ were 2^{-i}, the chain would make infinitely many transitions in one time unit. We are assuming that this does not happen.

We will now derive an alternative description of a continuous-time Markov chain. Let $(X(t), t \in \mathbb{R})$ be a Markov chain (process) with transition rates $q(j,k)$. The Markovian property means that conditional on the present, the future and the past are independent. Let $q(j) = \sum_{k\in S} q(j,k)$, where we set $q(j,j) = 0$ for all $j \in S$.

Fact. Starting in the state j, the process $X(t)$ stays in j for an exponentially distributed time with mean $1/q(j)$. The probability that the next state is k is given by $p(j,k) = \frac{q(j,k)}{q(j)}$. The matrix $(p(j,k))$ is called the transition matrix for the embedded jump chain.

Definition. $\pi = (\pi(j))$ is an equilibrium distribution for X if:

- $\pi(j) > 0 \ \forall j \in S$
- $\sum_{j\in S} \pi(j) = 1$
- $\pi(j) \sum_{k\in S} q(j,k) = \sum_{k \in S} \pi(k) q(k,j)$.

π is a limiting, ergodic, and stationary distribution:

Limiting: $\forall j \in S$, $\mathbb{P}(X(t) = j) \to \pi(j)$ as $t \to \infty$

Ergodic: $\forall j \in S$, $\frac{1}{T}\int_0^T I[X(t) = j]dt \to \pi(j)$ as $T \to \infty$. Here, I is the indicator function, and the integral is the amount of time, between 0 and T, that the process spends in state j.

Stationary: If $\mathbb{P}(X(0) = j) = \pi(j)$ for all $j \in S$, then $\mathbb{P}(X(t) = j) = \pi(j)$ for all $j \in S$ and all $t \geq 0$.

Proposition 1.1. *Suppose $(X(t)), t \in \mathbb{R}$ is a stationary Markov process with transition rates $q(j,k)$ and equilibrium distribution π; i.e. $\mathbb{P}(X(t) = j) = \pi(j)$ for all t. Then $(X(-t), t \in \mathbb{R})$ is a stationary Markov process with the same equilibrium distribution π and transition rates $q'(j,k) = \frac{\pi(k)}{\pi(j)} q(k,j)$. This is called the* reversed process.

Proof. That $(X(-t))$ is a stationary Markov process is clear (note that the alternate characterisation of the Markov property – conditional on the present, the future is independent of the past – is symmetric in time). The statement is about the transition rates.

[1] An object for which $X_n \in S$ for $n \in \mathbb{Z}$ and S countable is called a Markov *chain*. An object for which $X(t) \in S$ for $t \in \mathbb{R}$ and S uncountable is called a Markov *process*. An object for which $t \in \mathbb{R}$ but S is countable goes by either name, and we will be using them interchangeably.

Let $Y(t) = X(-t)$. Then

$$\mathbb{P}(Y(t+h) = k|Y(t) = j) = \frac{\mathbb{P}(Y(t+h) = k, Y(t) = j)}{\mathbb{P}(Y(t) = j)} = \frac{\mathbb{P}(X(-t-h) = k, X(-t) = j)}{\mathbb{P}(X(-t) = j)} = \frac{\pi(k)(q(k,j)h + o(h))}{\pi(j)}.$$

Therefore, $\lim_{h\to 0} \frac{1}{h}\mathbb{P}(Y(t+h) = k|Y(t) = j) = \frac{\pi(k)}{\pi(j)}q(k,j)$. □

Definition. If the process satisfies

(1) $$\pi(j)q(j,k) = \pi(k)q(k,j)\ \forall j,k \in S$$

we say that the process is *reversible*. Equation (1) is called *detailed balance*.

Note that detailed balance implies the invariance condition

(2) $$\pi(j)\sum_{k\in S} q(j,k) = \sum_{k\in S} \pi(k)q(k,j)\ \forall j \in S,$$

which is sometimes known as *full balance*. When true, detailed balance (1) is much easier to check than full balance (2).

Example 1.1 (Erlang's formula)**.** Calls arrive as a Poisson process of rate λ. A call lasts an exponentially distributed amount of time with parameter μ; these times are independent of each other and of the arrival times. There are C lines; if all C lines are busy, arriving calls are lost.

Let $X(t) = j$ be the number of busy lines; then the transition rates are given by

$$q(j,j+1) = \lambda,\ j = 0,1,\dots,C-1 \qquad q(j,j-1) = j\mu,\ j = 1,2,\dots,C$$

(note for the transition rate $j \to j-1$ that we are looking for the minimum of j independent exponentials).

Detailed balance:

$$\pi(j-1)q(j-1,j) = \pi(j)q(j,j-1) \implies \pi(j) = \frac{\lambda}{\mu j}\pi(j-1) = \left(\frac{\lambda}{\mu}\right)^j \frac{1}{j!}\pi(0).$$

Since also $\sum_{j=0}^C \pi(j) = 1$, we see $\pi(0) = \left(\sum_{j=0}^C (\frac{\lambda}{\mu})^j \frac{1}{j!}\right)^{-1}$.

Thus, the equilibrium probability that all lines are busy is

$$\pi(C) = \mathcal{E}(\frac{\lambda}{\mu}, C) \quad \text{where } \mathcal{E}(\nu, C) = \frac{\nu^C/C!}{\sum_{j=0}^C \nu^j/j!}.$$

This is known as Erlang's formula.

By the PASTA property ("Poisson arrivals see time averages"), this is also the proportion of calls lost. (One way to see the PASTA property is that, if you look back from a point of arrival, the interarrival times look like a sequence of independent exponentials, just as if you had picked an arbitrary fixed point and looked back from there.)

Example 1.2 (Telephone exchange in a finite village)**.** Suppose now that we have C lines and $M < \infty$ subscribers. Letting $X(t) = j$ be the number of busy lines, we now have the transition rates

$$q(j,j+1) = \lambda(M-j),\ j = 0,\dots,C-1 \qquad q(j,j-1) = \mu j,\ j = 1,2,\dots,C$$

(Note that if $M \to \infty$ with λM fixed, we recover the previous model.)

The equilibrium distribution $\pi_M(j)$ satisfies (check this!)

$$\pi_M(j-1)q(j-1,j) = \pi_M(j)q(j,j-1) \implies \pi_M(j) = \pi_M(0)\binom{M}{j}\left(\frac{\lambda}{\mu}\right)^j, \ j = 0,1,\dots,C.$$

Now consider the process at the times when a call is initiated:

$$\mathbb{P}(j \text{ lines are busy and a call is initiated in } (t,t+h)) = \pi_M(j)(\lambda(M-j)h+o(h)), \ j = 0,1,\dots,C.$$

Therefore, letting $h \to 0$, conditional on a call being initiated, the probability of j lines being busy is

$$\frac{\pi_M(j)\lambda(M-j)}{\sum \pi_M(i)\lambda(M-i)} \sim \pi_M(j)(M-j) \sim \binom{M}{j}(M-j)\left(\frac{\lambda}{\mu}\right)^j \sim \binom{M-1}{j}\left(\frac{\lambda}{\mu}\right)^j.$$

(Here, we have omitted the normalisation constants, and left only the dependence on j.) Normalising, we get $\pi_{M-1}(j), \ j = 0,1,\dots,C$. That is

$$\mathbb{P}(\text{an initiated call is lost}) = \pi_{M-1}(C) < \pi_M(C).$$

See also Question 3 on Example Sheet 1.

2. Queueing networks

2.1. $M/M/1$ **Queue.** An $M/M/1$ queue has a Poisson (**M**emoryless) process of arrivals, exponential (**M**emoryless) service times, and **1** server at the front of the queue. Letting $X(t) = j$ be the number of customers in the queue, the transition rates for X are

$$q(j,j+1) = \lambda, \ j = 0,1,\dots \qquad q(j,j-1) = \mu, \ j = 1,2,\dots$$

The stationary distribution of this process is $\pi(j) = (1-\rho)\rho^j$ where $\rho = \lambda/\mu$ is the *traffic intensity.*

Exercise 1. Check the detailed balance equations.

The reversed process has transition probabilities $q'(j,k) = \frac{\pi(k)}{\pi(j)}q(k,j) = q(j,k)$ for all $j,k \in \mathbb{Z}_+$.

Let $(X(t), t \in \mathbb{R})$ be a stationary $M/M/1$ queue, $\mathbb{P}(X(t) = j) = \pi(j)$ for all $j \in S$, $t \in \mathbb{R}$. Then reversibility means that

$$X(t) \stackrel{\mathcal{D}}{=} (X(-t), t \in \mathbb{R}),$$

where $\stackrel{\mathcal{D}}{=}$ means *equal in distribution.* That is, there is no statistic that can distinguish the two processes.

Now consider the sample path: The arrival process A counts the jumps up of the chain $X(t)$; the departure process D counts the jumps down of the chain $X(t)$. Note that A is a Poisson process of rate λ by the construction of the queue. On the other hand, A is defined on $(X(t), t \in \mathbb{R})$ exactly as D is defined on $(X(-t), t \in \mathbb{R})$ – the departures of the original process are the arrivals of the reversed process. Since $(X(t), t \in \mathbb{R})$ and $(X(-t), t \in \mathbb{R})$ are distributionally equivalent, we conclude that D is also a Poisson process of rate λ.

Theorem 2.1 (Burke, Reich)**.** *In equilibrium, the departure process from an $M/M/1$ queue is a Poisson process.*

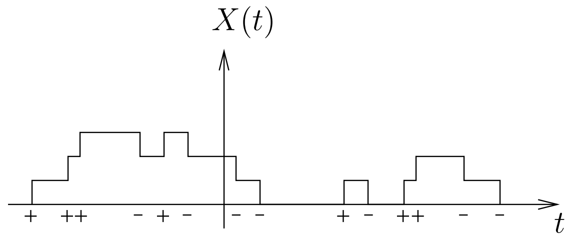


FIGURE 5. Sample path of the $M/M/1$ queue. $+$ marks the points of A, $-$ marks the points of D.

Exercise 2. What will happen in an $M/M/2$ queue? (The 2 means that there are two potential servers at the front; equivalently, $q(j, j-1) = 2\mu$ if $j \geq 2$, but $q(1,0) = \mu$ because only one of the servers is working.)

Remark. We must be in equilibrium for this to hold! For example, if we condition on the queue size being positive, then the time until the next departure is also exponential, but now of rate μ rather than λ. Another way of saying this is that A and D are both Poisson processes, but they are by no means independent. Below, we analyze their dependence structure.

For events A, B we write $A \perp\!\!\!\perp B$ to mean that A and B are independent. Now, for a fixed time $t_0 \in \mathbb{R}$ we have

$$(X(t), t \leq t_0) \perp\!\!\!\perp A \cap (t_0, \infty)$$

(future arrivals are independent of the past process). Applying this to the reversed process, we get

$$(X(t), t \geq t_1) \perp\!\!\!\perp D \cap (-\infty, t_1).$$

In particular, the number of people in the queue now is independent of the departure process up until now. (But clearly it isn't independent of the future departure process!)

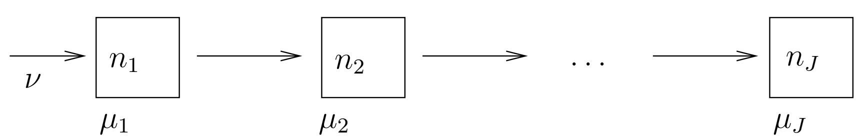


FIGURE 6. A series of $M/M/1$ queues

2.2. **A series of $M/M/1$ queues.** In queue j, the service times are independent exponentials of rate μ_j; assume $\nu < \mu_j$ for all j. Note that by the above section, in stationary regime each of the queues has a Poisson arrival process, so this really is a series of $M/M/1$ queues.

Let $\mathbf{n} = (n_1, \ldots, n_J)$ be a vector of the number of customers. The marginals for this system are $\pi_j(n_j) = (1-\rho_j)\rho_j^{n_j}$, where $\rho_j = \nu/\mu_j$. What is the joint distribution of all J queue lengths?

For a fixed time t_0, consider the following quantities:

(1) $n_1(t_0)$
(2) Departures from queue 1 prior to t_0
(3) Service times of customers in queues $2, 3, \ldots, J$
(4) $(n_2(t_0), \ldots, n_J(t_0))$.

We have (1) $\perp\!\!\!\perp$ (2) by the above discussion; and (3) $\perp\!\!\!\perp$ (1,2) by construction. Therefore, 1, 2, and 3 are mutually independent, and in particular (1) $\perp\!\!\!\perp$ (2, 3). On the other hand, (4) is a function of (2, 3), so we conclude that (1) $\perp\!\!\!\perp$ (4).

Similarly, $\forall j$ we have $n_j(t_0) \perp\!\!\!\perp (n_{j+1}(t_0), \ldots, n_J(t_0))$. Therefore,

$$\pi((n_1, \ldots, n_J)) = \prod_{j=1}^{J} \pi_j(n_j).$$

You can use this technique to show a number of other results; e.g., that the sojourn times of a customer in the different queues is independent. However, this technique is fragile; in particular, it will not stand up to any sort of feedback between queues. We will now develop a technique that doesn't gives such fine-detail results, but can tolerate feedback.

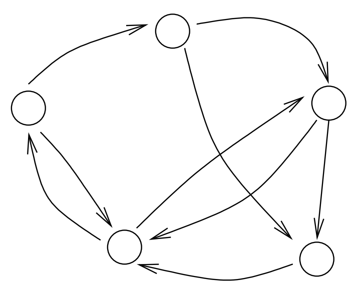


FIGURE 7. Closed migration process

2.3. **Closed migration process.** Let $\mathbf{n} = (n_1, \ldots, n_J)$ be a Markov process with state space $S = \{\mathbf{n} \in \mathbb{Z}_+^J : \sum_{j=1}^J n_j = N\}$. We refer to j as *colonies.*

Define the operator T_{jk} as

$$T_{jk}(n_1, \ldots, n_J) = \begin{cases} (n_1, \ldots, n_{j-1}, n_j - 1, n_{j+1}, \ldots, n_{k-1}, n_k + 1, n_{k+1}, \ldots, n_J), & j < k \\ (n_1, \ldots, n_{k-1}, n_k + 1, n_{k+1}, \ldots, n_{j-1}, n_j - 1, n_{j+1}, \ldots, n_J), & j > k \end{cases}$$

That is, T_{jk} transfers one member from colony j to colony k.

We take the transition rates for the process to be

$$q(n, T_{jk}(n)) = \lambda_{jk}\phi_j(n_j), \quad \phi_j(0) = 0.$$

Suppose that $\mathbf{n}$ is irreducible in S (i.e. the graph along which transitions occur is strongly connected – there is a path from any vertex to any other vertex). In this case, we call $\mathbf{n}$ a *closed migration process.*

Example 2.1. If $\phi_j(n) = \min(n, s)$, then colony j behaves as an s-server queue with exponential service times with parameter $\lambda_j = \sum_k \lambda_{jk}$. On leaving j, the individual goes to k with probability λ_{jk}/λ_j.

If $\phi_j(n) = n$ for all j, we get what is called a *linear migration process*; this is what would result from independent movement of the N individuals.

If $N = 1$, we have a random walk with equilibrium distribution (α_j), where the α_j satisfy

$$(3) \qquad \alpha_j > 0 \quad \sum_j \alpha_j = 1 \quad \alpha_j \sum_k \lambda_{jk} = \sum_k \alpha_k \lambda_{kj}, \qquad j = 1, 2, \ldots, J.$$

We refer to these equations as the *traffic equations*; when $N \neq 1$, they do not give equilibrium distributions of much of anything, but they do define a unique set of quantities α_j.

Remark. You will notice that we haven't actually specified the α_j, because we could always multiply λ_{kj} by a constant depending somehow on k and j. This doesn't make too much of a difference.

Theorem 2.2. *The equilibrium distribution for a closed migration process is given by*

$$\pi(\mathbf{n}) = B_N \prod_{j=1}^{J} \frac{\alpha_j^{n_j}}{\prod_{r=1}^{n_j} \phi_j(r)}, \qquad \mathbf{n} \in S.$$

Here, B_N is a normalisation constant, and α_j are the solutions to the traffic equations (3).

Although this expression looks somewhat complicated, it's really as simple as it could be: the distribution is a certain product over the individual colonies.

Proof. We will prove this by brute force, i.e. by checking the full balance equations

$$\pi(\mathbf{n}) \sum_j \sum_k q(\mathbf{n}, T_{jk}\mathbf{n}) \stackrel{?}{=} \sum_j \sum_k \pi(T_{jk}\mathbf{n}, \mathbf{n}) q(T_{jk}\mathbf{n}, \mathbf{n}).$$

These will be solved if we can solve the *partial balance* equations

$$\pi(\mathbf{n}) \sum_k q(\mathbf{n}, T_{jk}\mathbf{n}) \stackrel{?}{=} \sum_k \pi(T_{jk}\mathbf{n}) q(T_{jk}\mathbf{n}, \mathbf{n}) \quad \forall j$$

Note that these equations are concerned with an individual leaving or entering a single colony j. We now recall

$$q(\mathbf{n}, T_{jk}\mathbf{n}) = \lambda_{jk}\phi_j(n_j), \quad \pi(T_{jk}\mathbf{n}) = \pi(\mathbf{n}) \frac{\phi_j(n_j)}{\alpha_j} \frac{\alpha_k}{\phi_k(n_k+1)}, \quad q(T_{jk}\mathbf{n}, \mathbf{n}) = \lambda_{kj}\phi_k(n_k+1)$$

(note that $T_{jk}(\mathbf{n})$ has one more customer in colony k than $\mathbf{n}$ does, hence the appearance of $n_k + 1$ in the arguments). After plugging in and cancelling terms, we see that the partial balance equations are equivalent to

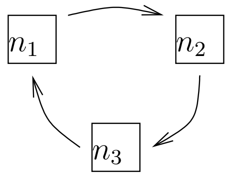
$$\sum_k \lambda_{jk} \stackrel{?}{=} \frac{1}{\alpha_j} \sum_k \alpha_k \lambda_{kj},$$

which is true by the definition of the α_j. □

Remark. The full balance equations state that the total probability flux into and out of any state is the same. The detailed balance equations state that the total probability flux between any pair of states is the same. Partial balance says that, for a fixed state, there is a subset of the states for which the total probability flux into and out of the group is equal.

Example 2.2. Suppose that we have K lines and a single operator. Calls arrive into the system as a Poisson process of rate ν. If at that time there are K calls already in the system, then the incoming call is lost. Otherwise, if the incoming call finds a free line, it must wait for the operator, who services calls in a first-come-first-serve fashion and takes an exponential amount of time with each call. Once a call has been put through, it will last another exponential amount of time.

We model this system as a closed migration process as in Figure 8. The transition rates



n_1 lines free
n_2 lines waiting for service
n_3 connected calls

FIGURE 8. Closed migration process for the telephone exchange

are

$$\begin{aligned} \lambda_{12} &= \nu \quad \phi_1(n_1) = I[n_1 > 0] \\ \lambda_{23} &= \lambda \quad \phi_2(n_2) = I[n_2 > 0] \\ \lambda_{31} &= \mu \quad \phi_3(n_3) = n_3 \end{aligned}$$

We can easily solve the traffic equations:

$$\alpha_1 : \alpha_2 : \alpha_3 = \frac{1}{\nu} : \frac{1}{\lambda} : \frac{1}{\mu}$$

since we have a random walk on the three vertices, and these are the average amounts of time it spends in each of the vertices. Therefore, by the theorem,

$$\pi(n_1, n_2, n_3) \sim \frac{1}{\nu^{n_1}} \frac{1}{\lambda^{n_2}} \frac{1}{\mu^{n_3}} \frac{1}{n_3!}.$$

In particular, the proportion of the incoming calls that are lost is

$$\mathbb{P}(n_1 = 0) = \sum_{n_2+n_3=K} \pi(0, n_2, n_3).$$

2.4. **Open migration process.** We now add the possibility of customers entering and exiting the system. Let

$$T_{j\cdot}\mathbf{n} = (n_1, \ldots, n_{j-1}, n_j - 1, n_{j+1}, \ldots, n_J) \qquad T_{\cdot k}\mathbf{n} = (n_1, \ldots, n_{k-1}, n_k + 1, n_{k+1}, \ldots, n_J)$$

That is, $T_{j\cdot}$ means that a single member of colony j left the system, and $T_{\cdot k}$ means that a single member entered colony k from the outside world. We will take as the transition rates for our system

$$q(\mathbf{n}, T_{jk}\mathbf{n}) = \lambda_{jk}\phi_j(n_j) \quad q(\mathbf{n}, T_{j\cdot}\mathbf{n}) = \mu_j\phi_j(n_j) \quad q(\mathbf{n}, T_{\cdot k}\mathbf{n}) = \nu_k.$$

Note that we are treating $\cdot$ as just another destination in the second equation, and assuming that immigration into colony k is Poisson (i.e., $\cdot$ behaves like a colony with an infinite number of customers, so that the rate of leaving it is unaffected by the number that have already left).

If the resulting process $\mathbf{n}$ is irreducible in $\mathbb{Z}_+^J$, we call $\mathbf{n}$ an *open migration process.*

Let $(\alpha_1, \ldots, \alpha_J)$ satisfy the *traffic equations*

$$\alpha_j(\mu_j + \sum_k \lambda_{jk}) = \nu_j + \sum_k \alpha_k\lambda_{kj} \quad \forall j. \tag{4}$$

Exercise 3. There exists a unique, positive solution to the traffic equations. (This is problem 1 on example sheet 2.)

Since we now have an infinite state-space, we are not guaranteed a normalisable invariant distribution. Let

$$b_j^{-1} = \sum_{n=0}^{\infty} \frac{\alpha_j^n}{\prod_{r=1}^n \phi_j(r)}, \quad b_j := 0 \text{ if the sum is infinite} \quad (j = 1, 2, \dots, J)$$

Theorem 2.3. *If $b_1, \dots, b_J > 0$ then $\mathbf{n}$ has equilibrium distribution*

$$\pi(\mathbf{n}) = \prod_{j=1}^{J} \pi_j(n_j), \quad \pi_j(n_j) = b_j \frac{\alpha_j^{n_j}}{\prod_{r=1}^{n_j} \phi_j(r)}.$$

Remark. This generalises the product form distribution we got for a series of $\cdot/M/1$ queues – note that the series of $\cdot/M/1$ queues was an open migration process.

Proof. Once again, we need to check the full balance equations

$$\pi(\mathbf{n}) \left(\sum_j \sum_k q(\mathbf{n}, T_{jk}\mathbf{n}) + \sum_j q(\mathbf{n}, T_{j\cdot}\mathbf{n}) + \sum_k q(\mathbf{n}, T_{\cdot k}\mathbf{n}) \right) \stackrel{?}{=}$$
$$\sum_j \sum_k \pi(T_{jk}\mathbf{n}, \mathbf{n}) q(T_{jk}\mathbf{n}, \mathbf{n}) + \sum_j \pi(T_{j\cdot}\mathbf{n}) q(T_{j\cdot}\mathbf{n}, \mathbf{n}) + \sum_k \pi(T_{\cdot k}\mathbf{n}) q(T_{\cdot k}\mathbf{n}, \mathbf{n}),$$

which will be satisfied if we can solve the partial balance equations

$$\pi(\mathbf{n}) \left(\sum_k q(\mathbf{n}, T_{jk}\mathbf{n}) + q(\mathbf{n}, T_{j\cdot}\mathbf{n}) \right) \stackrel{?}{=} \sum_k \pi(T_{jk}\mathbf{n}, \mathbf{n}) q(T_{jk}\mathbf{n}, \mathbf{n}) + \pi(T_{j\cdot}\mathbf{n}) q(T_{j\cdot}\mathbf{n}, \mathbf{n}) \tag{5}$$

and

$$\pi(\mathbf{n}) \sum_k q(\mathbf{n}, T_{\cdot k}\mathbf{n}) \stackrel{?}{=} \sum_k \pi(T_{\cdot k}\mathbf{n}) q(T_{\cdot k}\mathbf{n}, \mathbf{n}). \tag{6}$$

(These look a lot like the earlier partial balance equations with an added colony called $\cdot$.)

Exercise 4. Check Equations (5) by substitution.

To see that Equations (6) are satisfied, we substitute

$$q(\mathbf{n}, T_{\cdot k}\mathbf{n}) = \nu_k \quad \pi(T_{\cdot k}\mathbf{n}) = \pi(\mathbf{n}) \frac{\alpha_k}{\phi_k(n_k + 1)} \quad q(T_{\cdot k}\mathbf{n}, \mathbf{n}) = \mu_k \phi_k(n_k + 1)$$

to get after cancellation

$$\sum_k \nu_k \stackrel{?}{=} \sum_k \alpha_k \mu_k.$$

Note that this is not directly one of the traffic equations (4), but it will be what we get if we sum the equations (4) over all j (and cancel terms). □

Remark.
- The form of π establishes that at a fixed time t the random variables $n_1(t), n_2(t), \dots, n_J(t)$ are independent. They clearly do not have to be independent as processes! (For example, in a series of queues we know that if n_1 decreases by 1 at time t, then n_2 must increase by 1 at the same time.)

- The distribution $\pi_j(n_j)$ is the same as if the arrivals at colony j were Poisson processes of rate $\alpha_j\lambda_j$, and departures happened at rate $\lambda_j\phi_j(n_j)$ (where $\lambda_j = \mu_j + \sum_k \lambda_{jk}$ is the total rate at which departures actually happen from colony j when there's one individual there). However, in general the arrivals are not a Poisson process. For example, consider the system in Figure 9 below. If ν is quite small but λ_{21}/λ_2 is quite large, the typical arrival process into queue 1 will look like the right-hand picture: rare bursts of arrivals of geometric size.

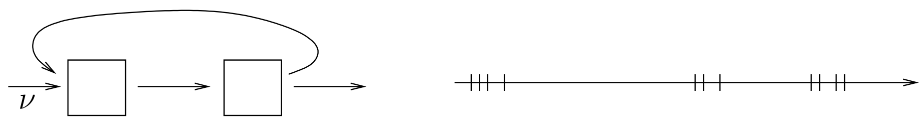


FIGURE 9. Simple open migration network. The typical arrival process into queue 1 is illustrated on the right.

Theorem 2.4. *If* $(\mathbf{n}(t), t \in \mathbb{R})$ *is a stationary open migration process, then so is the reversed process* $(\mathbf{n}(-t), t \in \mathbb{R})$.

Proof. This requires checking that the transition probabilities have the required form. We have

$$q'(\mathbf{n}, T_{jk}\mathbf{n}) = \frac{\pi(T_{jk}\mathbf{n})}{\pi(\mathbf{n})} q(T_{jk}\mathbf{n}, \mathbf{n}) = \frac{\phi_j(n_j)}{\alpha_j}\frac{\alpha_k}{\phi_k(n_k+1)}\lambda_{kj}\phi_k(n_k+1) = \lambda'_{jk}\phi_j(n_j), \quad \lambda'_{jk} = \frac{\alpha_k}{\alpha_j}\lambda_{kj}.$$

Similarly (check!), $q'(\mathbf{n}, T_{j\cdot}\mathbf{n}) = \mu'_j\phi_j(n_j)$ and $q'(\mathbf{n}, T_{\cdot k}\mathbf{n}) = \nu'_k$, where $\mu'_j = \nu_j/\alpha_j$ and $\nu'_k = \alpha_k\mu_k$. ☐

Corollary 2.5. *The exit process from colony* k*, i.e. the stream of individuals leaving the system from colony* k*, is a Poisson process of rate* $\alpha_k\mu_k$.

Remark. We are not saying anything about internal streams of individuals – at least some of these are almost certainly not Poisson, as per the example in Figure 9.

Example 2.3. Consider the following network. The arrows indicate positive transition rates (i.e. positive values of λ, ν, μ). On the left, the processes that we know are Poisson have

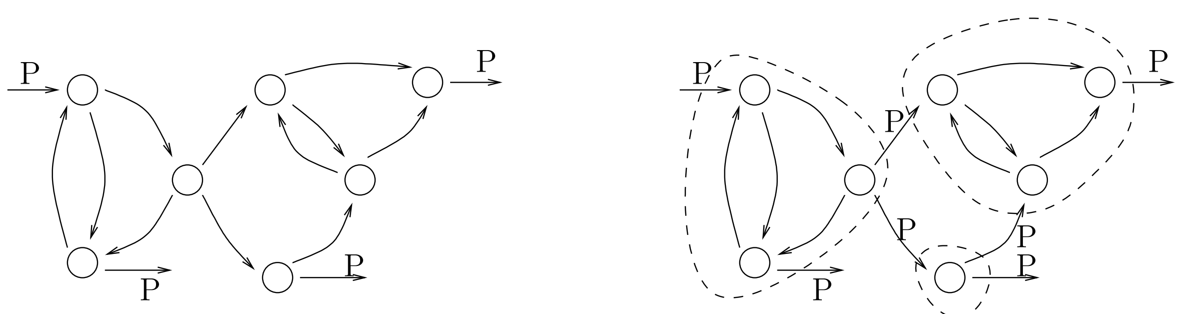


FIGURE 10. Example open migration network. On the right, the equivalence classes of communicating colonies.

been marked P. However, if we partition the colonies into communicating classes (two colonies communicate if there is a path between them in both directions), then the exit process from a communicating class must be Poisson as well (we can think of each communicating class as an open migration process in its own right). Since a probabilistic splitting of a Poisson process is also Poisson, all the arrows marked P on the right are also Poisson. The other arrows need not be Poisson processes.

Remark. All of our discussion so far holds with $J = \infty$ if we add the extra condition $\prod_{i=1}^{\infty} b_i > 0$. We will now discuss an example with an infinite state-space.

Example 2.4 (Family size process)**.** This process was first studied by David Kendall, who was interested in family names in Yorkshire. It has since been applied to genetic models of mutations, or to trap counts (counts of the number of each species of moth caught in a trap).

Let n_j be the number of families of size j, and let the transition rates be

$q(\mathbf{n}, T_{j,j+1}\mathbf{n}) = j\lambda n_j,$	$j = 1, 2, \dots$	λ is the intensity of producing offspring
$q(\mathbf{n}, T_{j,j-1}\mathbf{n}) = j\mu n_j,$	$j = 2, \dots$	μ is the rate of individuals dying
$q(\mathbf{n}, T_{\cdot 1}\mathbf{n}) = \nu$		immigration
$q(\mathbf{n}, T_{1\cdot}\mathbf{n}) = \mu n_1$		if a one-member family has a death, it disappears

In the below, we use $\phi_j(n_j) = n_j$.

The traffic equations have a solution (check!)

$$\alpha_j = \frac{\nu}{\lambda j}\left(\frac{\lambda}{\mu}\right)^j$$

with

$$b_j^{-1} = \sum_{n=0}^{\infty} \frac{\alpha_j^n}{n!} = e^{\alpha_j}, \qquad \prod_{j=1}^{\infty} b_i = e^{-\sum \alpha_i} > 0 \text{ if } \lambda < \mu.$$

(This is the expected result: we need the birth rate to be smaller than the death rate for the system to have a stationary distribution.) In this case,

$$\pi(n) = \prod_{j=1}^{\infty} e^{-\alpha_j} \frac{\alpha_j^{n_j}}{n_j!}, \quad n_j \sim \text{Poisson}(\alpha_j).$$

The number of families, $N = \sum_j n_j$, is distributed as a Poisson variable with parameter $\sum_j \alpha_j = -\frac{\nu}{\lambda}\log(1 - \lambda/\mu)$ (check!).

A difficult exercise: let $M = \sum_j jn_j$ be the number of individuals; them $\mathbb{E}[N|M] = \sum_{i=1}^{M} \rho/(\rho + i - 1)$, where $\rho = \nu/\lambda$. (It is not hard to see that the count of families given the number of individuals should be a measure of the relative sizes of immigration / mutation and birth rate / replication rate.)

2.5. **Generalizations.**

Example 2.5 (Communication network / optimal allocation)**.** Consider a network of cities with unidirectional communication links between them. If we model the links as queues (as in the right-hand diagram), we get something that isn't quite an open migration network (in an open migration network, individuals can travel in cycles, whereas packets would not do that).

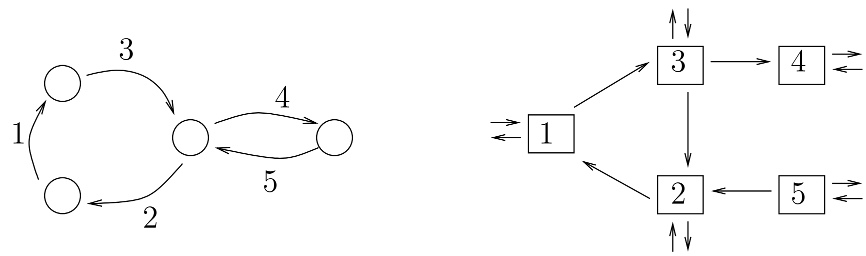


FIGURE 11. A communication network. On the right, the same network with links modelled as queues.

Let a_j be the average arrival rate of messages at queue j, and n_j be the number of messages in queue j. Supose that we have a product-form distribution with

$$\mathbb{P}(n_j = n) = (1 - a_j/\phi_j) \left(\frac{a_j}{\phi_j}\right)^n,$$

where ϕ_j is the service rate at queue j. (We can build an extended model of the system in which we can prove that this happens, but we won't be doing that.)

Suppose now that we are allowed to choose $\phi_1, \dots, \phi_J$ subject only to the budget constraint $\sum_{j=1}^J \phi_j = F$. What choice will minimize the mean number of packets in the system? (As we will see later, in the section on Little's law, this also minimizes the mean delay.)

That is, we want to

$$\min \mathbb{E}(\sum n_j) = \sum_j \frac{a_j}{\phi_j - a_j}$$

$$\text{subject to } \sum_{j=1}^J \phi_j = F,$$

$$\phi_j \geq 0 \ \forall j.$$

We are minimizing a convex function over a convex set, so we can use the Lagrangian:

$$\mathcal{L} = \sum_j \frac{a_j}{\phi_j - a_j} + y(\sum_{j=1}^J \phi_j - F),$$

and (check!)

$$\frac{\partial \mathcal{L}}{\partial \phi_j} = 0 \implies \phi_j = a_j + \sqrt{a_j/y}.$$

Choosing y to satisfy the budget constraint,

$$1/\sqrt{y} = \frac{F - \sum a_k}{\sum \sqrt{a_k}} \implies \phi_j = a_j + \frac{\sqrt{a_j}}{\sum \sqrt{a_k}}(F - \sum a_k).$$

This is known as *Kleinrock's square root allocation rule* (because excess capacity is allocated proportionally to the square root of the arrival rates). This was the scheme used to allocate capacity in the early network ARPANET. (Now most packets in the Internet are not in queues, but rather just travelling.)

Example 2.6 (Processor sharing)**.** Consider a model in which N individuals, or jobs, oscillate between being served by a server and being idle, where the server has total service rate

μ that it shares out equally between the n_0 jobs that are active. You can check (by verifying

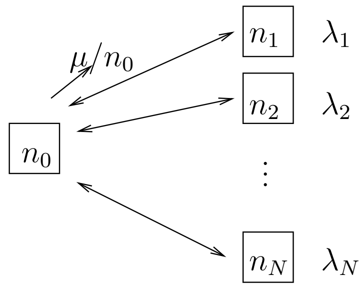


FIGURE 12. Processor-sharing system. n_0 jobs are currently being served. $n_j = 0$ or 1 for $j \neq 0$.

detailed balance) that the stationary distribution for this system is

$$\pi(n_0, \ldots, n_N) \sim n_0! \mu^{-n_0} \prod_{j=1}^{N} \lambda_j^{-n_j}.$$

The interesting fact about this example is that while in order to check detailed balance we need to assume quite specific service and arrival disciplines (Poisson arrivals and independent exponential service times), the stationary distribution is actually the same if the processes are quite arbitrary with the same mean. This is true for certain scheduling disciplines – for example, it is true for the processor sharing queue, but not for the first-come-first-served discipline. We will not be looking into this in this course.

2.6. **Little's Law.** Consider a stochastic process $(X_t, t \geq 0)$ on which is defined $n(t) = n(X(t))$, the *number of customers in the system* at time t. Suppose $(X_t, t \geq 0)$ has a *regeneration point* E_0 with the property that $X(0) \in E_0$ is a regeneration point, and when a system returns to E_0 the system probabilistically restarts. Suppose also that the regeneration point is such that $n(E_0) = 0$. (For example, in an open migration process a regeneration point may be "entire system is empty".) Let T_1 be the time of firts return to E_0, and suppose $\mathbb{E}T_1 < \infty$. Then we can define the mean number of customers in the system

$$\lim_{T \to \infty} \frac{1}{T} \int_0^T n(s) ds \left(\stackrel{w.p.1}{=} \lim_{T \to \infty} \frac{1}{T} \int_0^T \mathbb{E}n(s) ds \right) = \frac{\mathbb{E} \int_0^{T_1} n(s) ds}{\mathbb{E}T_1} =: L$$

(that is, the average number of customers in the system can be computed from a single regeneration period).

Also, let $W_n, n = 1, 2, \ldots$ be the amount of time spent by the nth individual in the system. (Note that if, in an open migration system, we are interested in $n(t)$ being the number of customers in a subset of the states, then it is possible for a customer to leave the "system" and reenter it later; we add up all the time he spends in the system.) Then we can define the mean time in the system as

$$\lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} W_n \left(\stackrel{w.p.1}{=} \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}W_n \right) = \frac{\mathbb{E} \sum_{n=1}^{N} W_n}{\mathbb{E}N} =: W,$$

where N is the number of customers who arrive during the first regeneration cycle $[0, T_1]$.

Finally, we can define the mean arrival rate

$$\lim_{T\to\infty} \frac{1}{T}(\text{number of arrivals in } [0,T]) \left(\overset{w.p.1}{=} \lim_{T\to\infty} \frac{1}{T}\mathbb{E}(\text{number of arrivals in } [0,T]) \right) = \frac{\mathbb{E}N}{\mathbb{E}T_1} =: \lambda.$$

Note that we do not assume that the arrivals are Poisson of rate λ, just that the long-run average arrival rate is λ.

Remark. We will not prove any of these statements. They follow from renewal theory (as developed, e.g., in Part II Applied Probability course).

Theorem 2.6 (Little's Law)**.** *For "quite general" queue or system, $L = \lambda W$.*

Proof. Note

$$L = \frac{\mathbb{E}\int_0^{T_1} n(s)ds}{\mathbb{E}T_1} = \frac{\mathbb{E}\int_0^{T_1} n(s)ds}{\mathbb{E}N}\frac{\mathbb{E}N}{\mathbb{E}T_1}$$

whereas

$$\lambda W = \frac{\mathbb{E}N}{\mathbb{E}T_1}\frac{\mathbb{E}\sum_{i=1}^N W_i}{\mathbb{E}N}.$$

Thus, it suffices to show $\int_0^{T_1} n(s)ds = \sum_{i=1}^N W_i$. See Figure 13:

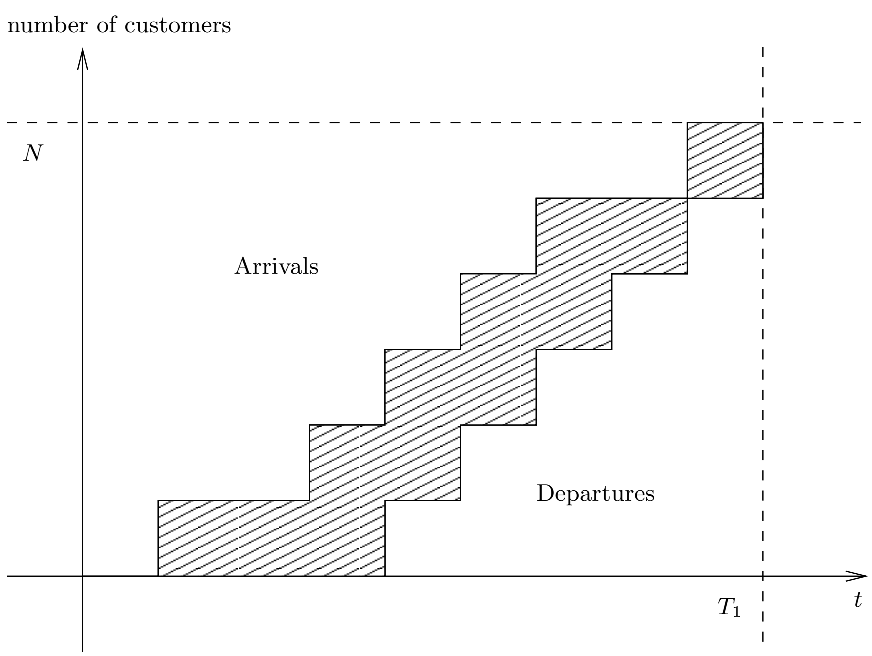


FIGURE 13. Shaded area is $\int_0^{T_1} n(s)ds$, but also $\sum_{i=1}^N W_i$.

It is clear that the shaded area is equal to $\int_0^{T_1} n(s)ds$, and that if the service discipline is First-In First-Out then it is also equal to $\sum_{i=1}^N W_i$. However, it's also clear that $\sum_{i=1}^N W_i$ doesn't actually depend on the service discipline, finishing the proof. $\square$

Remark. Little's law holds whenever we can make sense of the quantities L, λ, and W (the proof of the relationship between them is going to be essentially the same; the question is simply whether it makes sense to talk of the average arrival rate, number of customers in the system, or waiting time of a customer.)

2.7. **Linear migration processes.** We will talk about time-dependent linear migration processes on a general state space. Let $\mathcal{X}$ be a set, and $\mathcal{F}$ a σ-algebra of measurable subsets of $\mathcal{X}$.

Definition. A set of points in $\mathcal{X}$ is distributed as a Poisson process with mean measure $M(E)$ for $E \in \mathcal{F}$ if for disjoint sets $E_1, \ldots, E_k \in \mathcal{F}$ the numbers of points in E_i are independent Poisson random variables with mean $M(E_i)$.

Example 2.7. If $\mathcal{X} = \mathbb{R}$, $\mathcal{F}$ is the Borel σ-algebra generated by open intervals, and M is the Lebesgue measure, then (check!) the distances between successive points are independent exponential random variables.

Suppose individuals arrive at $\mathcal{X}$ in a Poisson stream of rate ν, and then move independently through $\mathcal{X}$ according to some stochastic process before possibly leaving $\mathcal{X}$.

Example 2.8. Consider $\mathcal{X}$ to be a semi-infinite motorway. Cars enter the motorway as points of a Poisson process, and then move independently to the right at a fixed speed; the speeds of the different cars are iid random variables. At $t = 0$ the system is empty; at time t, we can ask about the number of cars in a subset of the motorway.

For $E \in \mathcal{F}$, let

$$P(t, E) = \mathbb{P}(\text{individual is in } E \text{ at time } t \text{ after her arrival into the system}).$$

Suppose that $\mathcal{X}$ is empty (of individuals, that is) at time $t = 0$.

Theorem 2.7. *At time t, individuals are distributed over $\mathcal{X}$ according to a Poisson process with mean measure*

$$M(t, E) = \nu \int_0^t P(u, E) du, \quad E \in \mathcal{F}.$$

Remark. In working out the family name example, we saw that the stationary distribution for a linear migration process with a countable state space has this form. We are now extending this result.

Proof. Let $E_1, \ldots, E_k \in \mathcal{F}$ be disjoint, and let $n_j(t)$ be the number of individuals in E_j at time t. Write $\mathbf{z}^{\mathbf{n}(t)}$ for $z_1^{n_1(t)} \ldots z_k^{n_k(t)}$. We will work out the joint probability generating function of $n_j(t)$, $\mathbb{E}\mathbf{z}^{\mathbf{n}(t)}$, and will then see that it looks like the product of probability generating functions of Poisson random variables.

Let m be the number of arrivals into the system in $(0, t)$. Conditional on m, the arrival times $\tau_1, \ldots, \tau_m$ are independent and uniform in $(0, t)$ (unordered, of course). Let A_{ri} be the event that the individual who arrived at time τ_r is in E_i at time t; then $\mathbb{P}(A_{ri}) = P(t - \tau_r, E_i)$.

Now,

$$\begin{aligned}
\mathbb{E}[\mathbf{z}^{\mathbf{n}(t)} | m, \tau_1, \ldots, \tau_m] &= \mathbb{E}\left[\prod_{r=1}^m \prod_{i=1}^k z_i^{I[A_{ri}]} | m, \tau_1, \ldots, \tau_m\right] \\
&= \prod_{r=1}^m \mathbb{E}[\prod_{i=1}^k z_i^{I[A_{ri}]} | \tau_r] \text{ by independence of } \tau_r \\
&= \prod_{r=1}^m \left(1 - \sum_{i=1}^k (1 - z_i) P(t - \tau_r, E_i)\right)
\end{aligned}$$

Taking the average over τ_r, we get

$$\mathbb{E}[\mathbf{z}^{\mathbf{n}(t)}|m] = \prod_{r=1}^{m} \left(1 - \sum_{i=1}^{k}(1-z_i)\frac{1}{t}\int_0^t P(t-\tau, E_i)d\tau\right) = \left(1 - \sum_{i=1}^{k}(1-z_i)\frac{1}{t}\int_0^t P(t-\tau, E_i)d\tau\right)^m.$$

To take the average over m, note that m is a Poisson random variable with mean νt, and for a Poisson random variable X with mean λ we have $\mathbb{E}z^X = e^{-(1-z)\lambda}$. Therefore,

$$\mathbb{E}[\mathbf{z}^{\mathbf{n}t}] = \exp(-\nu t\sum_{i=1}^{k}(1-z_i)\frac{1}{t}\int_0^t P(u, E_i)du) = \prod_{i=1}^{k}\exp(-(1-z_i)M(t, E_i)).$$

This shows that the joint probability generating function of $n_1(t), \ldots, n_k(t)$ is the product of probability generating functions of Poisson random variables with means $M(t, E_i)$; therefore, $n_1(t), \ldots, n_k(t)$ are independent Poisson with the right means, as required. $\square$

3. Loss networks

Example 3.1 (Some examples)**.**

- Telephone exchange on a network: there are circuits between some of the cities (not in a complete graph), and an arriving call needs a free circuit on each link of the route it will take through the network. Otherwise, the call will be lost.
- Voice over IP: there are no physical circuits here, but there is still a notion of capacity, and if a new arriving call would put some of the intermediate links above their bandwidth, it should not be accepted.
- Communication over an optical fibre ring: again, calls use capacity rather than physical circuits here. We can think about calls of different type needing different amounts of capacity; a video call is equivalent to about 32 voice calls.

The simplest model of a loss network is a single link with C circuits. The arrivals are Poisson of rate ν, the holding times are exponential with mean 1, and blocked calls are lost. In this case, as we know from Erlang's formula,

$$\pi(n) \sim \frac{\nu^n}{n!}, \quad \mathbb{P}(\text{an arriving call is lost}) = \pi(C) = \nu^C/C!\left(\sum_{i=1}^{C}\nu^i/i!\right)^{-1} = \mathcal{E}(\nu, C).$$

3.1. **Network model.**

Let the set of links be $1, 2, \ldots, J$, and let C_j be the number of circuits on link j. A *route* r is a subset of $\{1, 2, \ldots, J\}$, and the set of all routes is called R (this is a subset of the power set of links). Let A be the *link-route incidence matrix*,

$$A_{jr} = \begin{cases} 1, & j \in r \\ 0, & j \notin r \end{cases}$$

(Later, we will allow more general coefficients for A_{jr}.)

Call requesting route r arrive as a Poisson process of rate ν_r. They are accepted if there is at least one free circuit on each link $j \in r$; otherwise they are lost. Accepted calls last an exponentially distributed time with mean 1; arrival times and holding times are independent.

Let n_r be the number of calls in progress on route r. The number of circuits busy on link j is given by $\sum_r A_{jr}n_r$. Let $\mathbf{n} = (n_r, r \in R)$, and let $\mathbf{C} = (C_1, \ldots, C_J)$.

Then $\mathbf{n}$ is a Markov process with state space $\{\mathbf{n} \in \mathbb{Z}_+^R : A\mathbf{n} \leq \mathbf{C}\}$ (the inequality is to be read componentwise). This process is called a *loss network with fixed routing*.

3.2. **Approximation procedure.** We will later compute the exact equilibrium distribution for $\mathbf{n}$. However, it is not very useful in practice because the normalising constant in it is difficult to estimate; moreover, the state space of $\mathbf{n}$ is very large, and we are generally interested only in some aggregate states. Instead, we will compute below an approximate model.

Let B_j be the probability of blocking on link j (i.e. the probability that, when a call needing link j arrives, it finds that all circuits on the link are busy.) If the blocking probabilities were independent from link to link (which in a real network they are not!), then the traffic offered to link j would be Poisson with rate

$$\sum_r A_{jr}\nu_r \prod_{i\in r-\{j\}} (1-B_i).$$

(The traffic along each route is thinned at all the other links independently, giving a Poisson process of rate $\nu_r \prod_{i\in r-\{j\}}(1-B_i)$; and then these independent Poisson processes are added up for each route passing through j.) By Erlang's formula, we would then expect

$$B_j = \mathcal{E}\left(\sum_r A_{jr}\nu_r \prod_{i\in r-\{j\}} (1-B_i), C_j\right). \tag{7}$$

A solution to these equations exists by the Brouwer fixed point theorem. Indeed, (7) defines a map $F : [0,1]^J \to [0,1]^J$ via

$$(B_j,\ j=1,\dots,J) \mapsto (\mathcal{E}\left(\sum_r A_{jr}\nu_r \prod_{i\in r-\{j\}} (1-B_i), C_j\right),\ j=1,\dots,J).$$

The map F is a continuous map from a compact, convex set to itself, and thus by Brouwer's fixed point theorem has at least one fixed point. (We will later see that this fixed point is unique.) This fixed point, i.e. solution to (7) is called the *Erlang fixed point.*

Example 3.2. Consider the following network with three cities with an exchange in the middle. We are given the arrival rates $\nu_{12}, \nu_{23}, \nu_{31}$ (here, $\nu_{12} = \nu_{\{1,2\}}$). The approximation

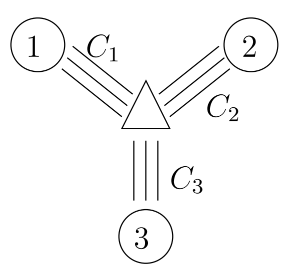


FIGURE 14. Telephone exchange for three cities

reads

$$\begin{aligned}
B_1 &= \mathcal{E}(\nu_{12}(1-B_2) + \nu_{31}(1-B_3), C_1)\\
B_2 &= \mathcal{E}(\nu_{12}(1-B_1) + \nu_{23}(1-B_3), C_2)\\
B_3 &= \mathcal{E}(\nu_{23}(1-B_2) + \nu_{31}(1-B_1), C_3)
\end{aligned}$$

The loss probabilities are then

$$L_{12} = 1 - (1 - B_1)(1 - B_2)$$

and similarly for L_{23}, L_{31}.

Remark. Solving the equations (7) exactly is almost certainly difficult. However, if we simply iterate the transformation F, we will almost certainly quickly converge to the fixed point (and if we use a damped iteration, i.e. taking a convex combination of the previous point and its image under F as the next point, we are guaranteed convergence). This is the method used in practice.

3.3. **Truncating reversible processes.** We now describe a way to compute the equilibrium distribution of the loss network above precisely.

Consider a Markov process with transition rates $(q(j, k),\ j, k \in S)$. Say that it is *truncated* to a set $A \subset S$ if $q(j, k)$ is changed to 0 for $j \in A$, $k \in S - A$, and the resulting process is irreducible in A.

Lemma 3.1. *If a reversible Markov process with state space S and equilibrium distribution $(\pi(j),\ j \in S)$ is truncated to $A \subset S$, then the resulting Markov process is reversible and has equilibrium distribution*

$$\pi(j) \left(\sum_{k \in A} \pi(k) \right)^{-1}, \ j \in A. \tag{8}$$

Proof.

$$\pi(j)q(j, k) = \pi(k)q(k, j)$$

by the reversibility of the original process, so (8) satisfies detailed balance. □

If the original process was not reversible, then (8) is the equilibrium distribution of the truncated process if and only if

$$\pi(j) \sum_{k \in A} q(j, k) = \sum_{k \in A} \pi(k)q(k, j).$$

You should check that this is what we referred to as "partial balance" earlier.

Now, consider a loss network with fixed routing for which $C_1 = \ldots = C_J = \infty$. This becomes a linear migration process with transition rates

$$q(\mathbf{n}, T_{\cdot r}\mathbf{n}) = \nu_r, \quad q(\mathbf{n}, T_{r\cdot}\mathbf{n}) = n_r$$

and equilibrium distribution

$$\pi(\mathbf{n}) = \prod_{r \in R} e^{-\nu_r} \frac{\nu_r^{n_r}}{n_r!}$$

(Equivalently, the individual routes become independent.) If we now truncate $\mathbf{n}$ to $S(\mathbf{C}) = \{\mathbf{n} : A\mathbf{n} \leq \mathbf{C}\}$, we will get the original loss network with fixed routing (and finite capacities). Therefore, the equilibrium distribution is

$$\pi'(\mathbf{n}) = G(\mathbf{C}) \prod_r \frac{\nu_r^{n_r}}{n_r!}, \ \mathbf{n} \in S(\mathbf{C}),$$

where

$$G(\mathbf{C}) = \left(\sum_{\mathbf{n}\in S(\mathbf{C})} \prod_r \frac{\nu_r^{n_r}}{n_r!} \right)^{-1}.$$

Unfortunately, computing this constant is NP-hard, but we will be able to use the form of the distribution to prove that the Erlang fixed point is a good approximation in a certain limiting regime.

Example 3.3 (Call repacking)**.** Consider the network in Figure 15. Suppose that calls can

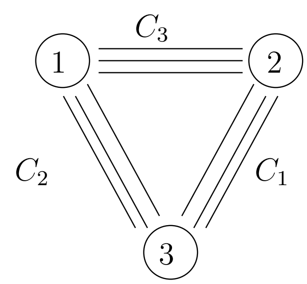


FIGURE 15. Telephone network with call repacking

be rerouted, even if in progress, if this will allow another call to be accepted. (We also, as always, assume that arrivals are Poisson and holding times are exponential.)

Let $n_{\alpha\beta}$ be the number of calls in progress between α and β. Then $\mathbf{n} = (n_{12}, n_{23}, n_{31})$ is a linear migration process with equilibrium distribution

$$\prod_r e^{-\nu_r} \frac{\nu_r^{n_r}}{n_r!}, \quad r = \{1,2\},\ \{2,3\},\ \{3,1\}$$

truncated to

$$A = \{\mathbf{n} : n_{12} + n_{23} \le C_3 + C_1,\ n_{23} + n_{31} \le C_1 + C_2,\ n_{31} + n_{12} \le C_2 + C_3\}$$

Indeed, it's clear that these restrictions are necessary. (They are *cut constraints* corresponding to the cut separating one of the vertices from the rest of the network). To see that they are sufficient, note that if all three inequalities of the form $n_{23} \le C_1$ hold, the state $\mathbf{n}$ is clearly feasible. If not, suppose WLOG $n_{23} > C_1$. Then we can route C_1 of the calls directly, and reroute the remaining $n_{23} - C_1$ calls via node 1. This will be possible provided $n_{23} - C_1 \le \min(C_3 - n_{12}, C_2 - n_{31})$, i.e. provided $n_{12} + n_{23} \le C_1 + C_3$ and $n_{23} + n_{31} \le C_1 + C_2$.

Thus, the equilibrium distribution is

$$\pi(\mathbf{n}) = G(A) \prod_r \frac{\nu_r^{n_r}}{n_r!}, \quad \mathbf{n} \in A,\ r = \{1,2\},\ \{2,3\},\ \{3,1\}.$$

Remark. This is equivalent to the following network:

If you're interested in graph theory, you should think about the set of graphs for which the above argument works, i.e. for which we only get single-vertex constraints here (perfect graphs, perhaps?) For example, suppose that we have a mobile cellular network, where the cells constitute a hexagonal tiling of the plane. There are multiple (C) frequencies on which communication can happen, but two adjacent cells cannot use the same frequency. It is clear that for every vertex we have a constraint on the three cells around it, $n_\alpha + n_\beta + n_\gamma \le C$; in fact, if we allow call repacking in this network, these are the only constraints.

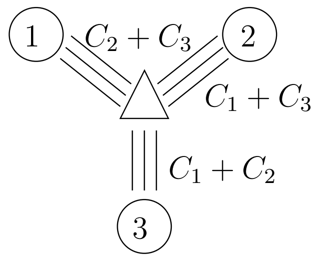


FIGURE 16. An equivalent network to the repacking model

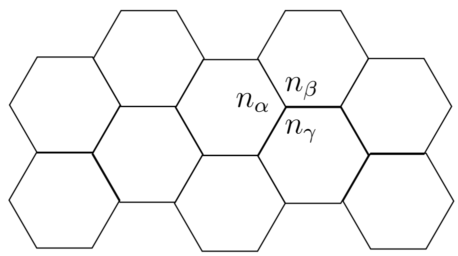


FIGURE 17. Mobile cellular network

We now compute the loss probabilities of a loss network precisely. From now on, we allow the link-route incidence matrix A to have entries in $\mathbb{Z}_+$, not just 0 or 1. The equilibrium distribution for the network is given by

$$\pi(\mathbf{n}) = G(\mathbf{C}) \prod_r \frac{\nu_r^{n_r}}{n_r!}, \quad \mathbf{n} \in S(\mathbf{C}) = \{\mathbf{n} : A\mathbf{n} \le \mathbf{C}\}$$

with $G(\mathbf{C}) = \left(\sum_{\mathbf{n} \in S(\mathbf{C})} \prod_r \frac{\nu_r^{n_r}}{n_r!} \right)^{-1}$, then the equilibrium probability that a call on route r is going to be accepted is

$$1 - L_r = \sum_{\mathbf{n} \in S(\mathbf{C} - A\mathbf{e}_r)} \pi(\mathbf{n}),$$

where $\mathbf{e}_r \in S(\mathbf{C})$ is the unit vector describing one call in progress on route r. Thus,

$$1 - L_r = \frac{G(\mathbf{C})}{G(\mathbf{C} - A\mathbf{e}_r)}.$$

This is not a useful result for complex (R big) or large (C big) networks, because computing the constants is NP-hard. However, we might hope for a limit result. We know that for large ν and C the Poisson distribution is going to be well approximated by a multivariate normal. Conditioning a multivariate normal on an inequality will have one of two effects: if the center is on the feasible side of the inequality, then the constraint has very little effect; if the center is on the infeasible side of the inequality, we will effectively restrict the distribution to the boundary of the constraint (because the tail of the normal distribution dies off very quickly), and the restriction of a multivariate normal to an affine subspace is again a multivariate normal distribution. We now begin to make the above more precise.

3.4. **Maximum probability.** We want to look for $\max \pi(\mathbf{n})$ over $\mathbf{n} \in \mathbb{Z}_+^R$ and $A\mathbf{n} \leq \mathbf{C}$. Write

$$\log \pi(\mathbf{n}) = \log \left(G(\mathbf{C}) \prod_r \frac{\nu_r^{n_r}}{n_r!} \right) = \log G(\mathbf{C}) + \sum_r \left(n_r \log \nu_r - \log(n_r!) \right).$$

By Stirling's approximation,

$$n! \approx \sqrt{2\pi} n^{n+1/2} e^{-n} \text{ as } n \to \infty,$$

so

$$\log(n!) \approx n \log n - n + O(\log n).$$

We will now replace the discrete variable $\mathbf{n}$ by a continuous variable $\mathbf{x}$, and also ignore the $O(\log n)$ terms to obtain the following PRIMAL problem:

$$\max \sum_r x_r \log \nu_r - x_r \log x_r + x_r$$
$$\text{subject to } A\mathbf{x} \leq \mathbf{C}$$
$$\mathbf{x} \geq 0.$$

Strong Lagrangian principle applies: the objective function is strictly concave, and the feasible region is a closed convex set. The Lagrangian is

$$\mathcal{L}(\mathbf{x}, \mathbf{z}; \mathbf{y}) = \sum_r x_r \log \nu_r - x_r \log x_r + x_r + \sum_j y_j (C_j - \sum_r A_{jr} x_r - z_j)$$

Here, $\mathbf{z} \geq 0$ are the slack variables and $\mathbf{y}$ are the Lagrange multipliers. We rewrite

$$\mathcal{L}(\mathbf{x}, \mathbf{z}; \mathbf{y}) = \sum_r x_r + \sum_r x_r (\log \nu_r - \log x_r - \sum_j y_j A_{jr}) + \sum_j y_j C_j - \sum_j y_j z_j$$

Since we want to maximize the Lagrangian over $\mathbf{z} \geq 0$ with a finite maximum, we require $\mathbf{y} \geq 0$; and at the optimal point we have $\mathbf{y} \cdot \mathbf{z} = 0$. Further, differentiating with respect to x_r gives

$$\log \nu_r - \log x_r - \sum_j y_j A_{jr} = 0,$$

or

$$\overline{x}_r = \nu_r e^{-\sum_j y_j A_{jr}}.$$

Then

$$\mathcal{L}(\overline{\mathbf{x}}, \overline{\mathbf{z}}; \mathbf{y}) = \sum_r \overline{x}_r + \sum_j y_j C_j = \sum_r \nu_r e^{-\sum_j y_j A_{jr}} + \sum_j y_j C_j.$$

This gives the Lagrangian DUAL problem as

$$\min \sum_r \nu_r e^{-\sum_j y_j A_{jr}} + \sum_j y_j C_j$$
$$\text{subject to } \mathbf{y} \geq 0.$$

To summarize, we have the Lagrangian PRIMAL

$$\max \sum_r x_r \log \nu_r - x_r \log x_r + x_r$$
$$\text{subject to } A\mathbf{x} \leq \mathbf{C} \tag{9}$$
$$\mathbf{x} \geq 0,$$

the solutions have $x_r = \nu_r e^{-\sum_j y_j A_{jr}}$, and the Lagrangian DUAL is

$$\min \sum_r \nu_r e^{-\sum_j y_j A_{jr}} + \sum_j y_j C_j \quad \text{subject to } \mathbf{y} \geq 0. \tag{10}$$

By the Strong Lagrangian Principle, $\mathbf{y} = \overline{\mathbf{y}}$ can be chosen so that the corresponding $\overline{\mathbf{x}}$ satisfies $\overline{\mathbf{x}} \geq 0$, $A\overline{\mathbf{x}} \leq \mathbf{C}$ (primal feasibility), $\overline{\mathbf{y}} \geq 0$ (dual feasibility), and $\overline{\mathbf{y}} \cdot \overline{\mathbf{z}} = 0$ (complementary slackness), where $\overline{\mathbf{z}} = \mathbf{C} - A\overline{\mathbf{x}}$.

Let $e^{-y_j} = 1 - B_j$. Equivalently, $(B_1, \ldots, B_J) \in [0,1)^J$ can be chosen so that

$$\begin{cases} \sum_r A_{jr} \nu_r \prod_i (1-B_i)^{A_{ir}} & = C_j \text{ if } B_j > 0 \\ & \leq C_j \text{ if } B_j = 0. \end{cases} \tag{11}$$

We call these *conditions on* $\mathbf{B}$. (They are the complementary slackness conditions, since $B_j > 0$ if and only if $y_j > 0$.) Then $\overline{x}_r = \nu_r \prod_i (1-B_i)^{A_{ir}}$ for $r \in R$.

The conditions on $\mathbf{B}$ have a fluid flow interpretation: $\overline{x}_r = \nu_r \prod_i (1-B_i)^{A_{ir}}$ looks like a thinning of the arrival stream ν_r by a factor $(1 - B_i)$ per each link i that the route r goes through (with multiplicity). Then $\sum A_{jr} \nu_r \prod_i (1-B_i)^{A_{ir}}$ is the aggregated flow on link j. Complementary slackness tells us that blocking only happens on links that are at full capacity.

Let's summarise what we have shown so far, adding a little bit:

Theorem 3.2. *There exists a unique optimal solution $\overline{\mathbf{x}} = (\overline{x}_r, r \in R)$ to the PRIMAL problem* (9)*. It can be expressed in the form*

$$\overline{x}_r = \nu_r \prod_j (1 - B_j)^{A_{jr}}, \ r \in R,$$

where $\mathbf{B} = (B_1, \ldots, B_J)$ is any solution to the conditions on $\mathbf{B}$ (11)*. There always exists a vector $\mathbf{B}$ satisfying these conditions; it is unique if A has rank J. There is a one-to-one correspondence between vectors satisfying conditions on B and optima of DUAL* (10)*, given by $1 - B_j = e^{-\overline{y}_j}$, $j = 1, \ldots, J$.*

Proof. Strict concavity of the PRIMAL objective gives uniqueness of $\overline{\mathbf{x}}$. The form of $\overline{\mathbf{x}}$ was found above.

To show the existence of $\mathbf{B}$ without relying on the strong Lagrangian principle, note that in the DUAL we are minimising a strictly convex, differentiable function over the positive orthant. Therefore, it achieves its minimum at some point $\overline{\mathbf{y}}$, and at $\overline{\mathbf{y}}$ the partial derivative $\partial/\partial y_j|_{\overline{\mathbf{y}}}$ is either 0 or, if $\overline{y}_j = 0$, then the partial derivative is nonnegative. Now, if we differentiate the objective of the DUAL, we get

$$-\sum_r A_{jr} \nu_r e^{-\sum_i y_i A_{ir}} + C_j,$$

so the minimum over $\mathbf{y} \geq 0$ will be at a point where

$$\begin{cases} \sum_r A_{jr} \nu_r e^{-\sum_i y_i A_{ir}} & = C_j \text{ if } y_j > 0 \\ & \leq C_j \text{ if } y_j = 0. \end{cases}$$

Note that these are precisely the conditions on $\mathbf{B}$, if we let $B_j = 1 - e^{-y_j}$; this gives the existence of a solution to the conditions on $\mathbf{B}$, and the one-to-one correspondence with the optima of the optimization problem DUAL.

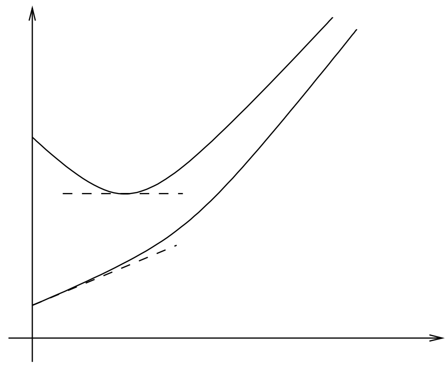


FIGURE 18. Minimum of a convex function over the nonnegative values occurs either where the derivative is zero, or at 0 with the derivative nonnegative.

Note that the DUAL objective function $\sum_r \nu_r e^{-\sum_j y_j A_{jr}}$ is strictly convex in the components of $\mathbf{y}A$. If A has rank J, the mapping $\mathbf{y} \mapsto \mathbf{y}A$ is one-to-one, and therefore the objective is strictly convex in the values of $\mathbf{y}$ and has a unique optimum. (If A is rank-deficient, we may not have strict convexity in the components of $\mathbf{y}$, and the optimizing $\mathbf{y}$ may not be unique.) □

Remark.

- A will have rank J if there is some single-link traffic on each link (i.e. $r = \{j\}$), since in that case A will contain a diagonal matrix as a submatrix. This is a natural assumption in many of the cases.
- Consider

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

corresponding to the system below.

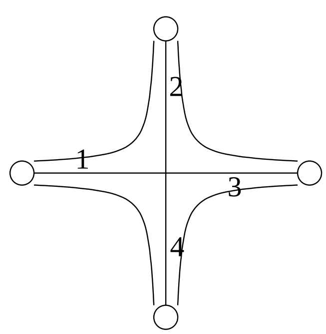


FIGURE 19. A network with a rank-deficient matrix

The routes are $R = \{12\}, \{23\}, \{34\}, \{41\}$. If $(B_1, \ldots, B_4)$ solves the conditions on $\mathbf{B}$, then so does

$$(1 - d(1 - B_1), 1 - d^{-1}(1 - B_2), 1 - d(1 - B_3), 1 - d^{-1}(1 - B_4)),$$

i.e. only $(1 - B_{odd})(1 - B_{even})$ is fixed. That is, non-uniqueness can in fact arise.

From now on we will assume that A has full rank J, and moreover that the unique solution to the conditions on $\mathbf{B}$ (11) satisfies

$$\sum_r A_{jr}\nu_r \prod_i (1-B_i)^{A_{ir}} < C_j \text{ if } B_j = 0,$$

i.e. the inequality is strict.

3.5. **A limiting regime.** Consider a sequence of networks as follows: replace $\boldsymbol{\nu} = (\nu_r, r \in R)$, $\mathbf{C} = (C_j, j = 1, \ldots, J)$ by $\boldsymbol{\nu}(N) = (\nu_r(N), r \in R)$ and $\mathbf{C}(N) = (C_j(N), j = 1, \ldots, J)$ for $N = 1, 2, \ldots$. We will consider the scaling regime in which

$$\frac{1}{N}\nu_r(N) \underset{n\to\infty}{\to} \nu_r \text{ for } r \in R$$
$$\frac{1}{N}C_j(N) \underset{n\to\infty}{\to} C_j \text{ for } j = 1, \ldots, J$$

(We take $\nu_r > 0$, $C_j > 0$ for all r and j.) We write $\mathbf{B}(N)$, $\overline{\mathbf{x}}(N)$, etc. for the Nth network.

Lemma 3.3. *As $N \to \infty$, $\mathbf{B}(N) \to \mathbf{B}$ and $\frac{1}{N}\overline{\mathbf{x}}(N) \to \mathbf{x}$.*

Proof. The sequence $\mathbf{B}(N)$, $N = 1, 2, \ldots$ takes values in the compact set $[0,1]^J$. Pick a convergent subsequence, say $\mathbf{B}(N_k)$, and let $\mathbf{B}' = \lim_{N_k\to\infty} \mathbf{B}(N_k)$. Then for $N = N_k$ large enough,

$$\begin{aligned}\sum_r A_{jr}\nu_r(N) \prod_i (1-B_i(N))^{A_{ir}} &= C_j(N) \text{ if } B'_j > 0\\ &\le C_j(N) \text{ if } B'_j = 0\end{aligned}$$

(Note that this is true with $B_j(N_k)$ on the right-hand side, but $B'_j > 0 \implies B_j(N_k) > 0$ for all N_k large enough and all $j = 1, \ldots, J$.)

Divide by N_k, and let $N_k \to \infty$, to obtain

$$\begin{aligned}\sum_r A_{jr}\nu_r \prod_i (1-B'_i)^{A_{ir}} &= C_j \text{ if } B'_j > 0\\ &\le C_j \text{ if } B'_j = 0.\end{aligned}$$

Note that this shows $B'_j \neq 1$ for all j (it can't be equal to 1 in the second line, and it can't be equal to 1 in the first line because $C_j > 0$ for all j, while the corresponding left-hand side would be equal to 0.

By the uniqueness of solutions to this set of equations, $\mathbf{B}' = \mathbf{B}$.

To finish the proof that $\mathbf{B}(N) \to \mathbf{B}$, we use a standard analysis argument. We have shown that any convergent sequence of $\mathbf{B}(N)$ converges to $\mathbf{B}$, and since we're in a compact set, any infinite subset of $\{\mathbf{B}(N)\}$ has a convergent subsequence. Consider an open neighbourhood O of $\mathbf{B}$; we will show that all but finitely many terms of $\mathbf{B}(N)$ lie in O. Indeed, the set $[0,1]^J \setminus O$ is still compact; if infinitely many terms $\mathbf{B}(N_k)$ were to lie in it, they would have a convergent subsequence; but that convergent subsequence would (by above) converge to $\mathbf{B}$, a contradiction.

Finally, since

$$\overline{x}_r(N) = \nu_r(N) \prod_i (1-B_i(N))^{A_{ir}},$$

we conclude that $\overline{x}_r(N)/N \to \overline{x}_r$. ☐

Let

$$p_N(\mathbf{n}(N)) = \prod_r \frac{\nu_r(N)^{n_r(N)}}{n_r(N)!},$$

(the unnormalised distribution of $\mathbf{n}(N)$), and also let

$$m_j(N) = C_j(N) - \sum_r A_{jr} n_r(N)$$

be the number of spare circuits on link j in the Nth network. Let

$$\mathcal{B} = \{j : B_j > 0\}, \quad A_{\mathcal{B}} = (A_{jr},\ j \in \mathcal{B}, r \in R).$$

The matrix $A_{\mathcal{B}}$ contains those rows of A that correspond to links with positive entries of $\mathcal{B}$ (i.e., intuitively, with positive blocking probability).

Our next task is to show that the distribution of $\mathbf{n}(N)$, appropriately scaled, converges to a normal distribution.

3.6. **Limit theorems.** Choose a sequence of states $\mathbf{n}(N) \in S(N)$, $N = 1, 2, \ldots$ and let $u_r(N) = N^{-1/2}(n_r(N) - \overline{x}_r(N))$ for $r \in R$. (This should be the appropriate scaling for proving a convergence to a normal distribution.)

Theorem 3.4. *The distribution of* $\mathbf{u}(N) = (u_r(N),\ r \in R)$ *converges to the distribution of the vector* $\mathbf{u} = (u_r,\ r \in R)$ *formed by conditioning independent normal random variables* $u_r \sim N(0, \overline{x}_r),\ r \in R$ *on* $A_{\mathcal{B}}\mathbf{u} = 0$*. Moments converge also, hence*

$$\frac{1}{N}\mathbb{E}[n_r(N)] \to \overline{x}_r,\ r \in R.$$

Sketch. By Stirling's formula,

$$p_N(\mathbf{n}(N)) = \prod_r (2\pi n_r(N))^{-1/2} \exp\left(\underbrace{\sum_r (n_r(N)\log\nu_r(N) - n_r(N)\log n_r(N) + n_r(N))}_{X} + O(N^{-1})\right)$$

Remark. Stirling's formula is

$$n! = (2\pi n)^{1/2}\exp(n\log n - n + \theta(n)),$$

where $\frac{1}{12n+1} < \theta(n) < \frac{1}{12n}$. (This follows by doing the Taylor expansion.) Our proof will not mention the form of this error term further, but you would need to use it to get tightness for the distributions of $\mathbf{u}(N)$. (The $O(N^{-1})$ term is uniform over all sequences $\mathbf{n}(N)$ such that $\sum_r u_r(N)^2 < \infty$.)

We now expand X:

$$X = -\sum_r n_r(N)\log\frac{n_r(N)}{e\nu_r(N)} = \underbrace{-\sum_r n_r(N)\log\frac{\overline{x}_r(N)}{\nu_r(N)}}_{\text{first term}}\underbrace{-\sum_r n_r(N)\log\frac{n_r(N)}{e\overline{x}_r(N)}}_{\text{second term}}$$

The first term, by definition of $\overline{\mathbf{x}}$ coming from an optimization problem, is

$$\sum_r n_r(N)\sum_j \overline{y}_j(N)A_{jr} = \sum_j \overline{y}_j(N)\sum_r A_{jr}n_r(N) = \sum_j \overline{y}_j(N)(C_j(N) - m_j(N)).$$

To deal with the second term, we note from the Taylor series expansion that

$$(1+a)(1-\log(1+a)) = 1 - \frac{1}{2}a^2 + o(a^2) \text{ as } a \to 0,$$

and therefore

$$\begin{aligned} -n_r(N)\log\frac{n_r(N)}{e\overline{x}_r(N)} &= (\overline{x}_r(N) + u_r(N)N^{1/2})\left(1 - \log(1 + \frac{u_r(N)N^{1/2}}{\overline{x}_r(N)})\right) \\ &= \overline{x}_r(N)\left(1 - \frac{1}{2}\left(\frac{u_r(N)N^{1/2}}{\overline{x}_r(N)}\right)^2 + o(N^{-1})\right) \\ &= \overline{x}_r(N) - \frac{u_r(N)^2}{2\overline{x}_r} + o(1) \end{aligned}$$

(we absorb the error from replacing $\overline{x}_r(N)/N$ by $\overline{x}_r$ into $o(1)$). Putting these together,

$$X = \sum_j \overline{y}_j(N)C_j(N) + \sum_r \overline{x}_r(N) - \sum_j \overline{y}_j(N)m_j(N) - \sum_r \frac{u_r(N)^2}{2\overline{x}_r} + o(1),$$

and thus

$$\underbrace{p_N(\mathbf{n}(N))\exp(-\sum_r \overline{x}_r(N) - \sum_j \overline{y}_j(N)C_j(N))}_{\text{term 1}} = \underbrace{\prod_j (1 - B_j(N))^{m_j(N)}}_{\text{term 2}} \underbrace{\prod_r (2\pi n_r(N))^{-1/2}}_{\text{term 3}} \underbrace{\exp\left(-\frac{u_r(N)^2}{2\overline{x}_r} + o(1)\right)}_{\text{term 4}}$$

Note that term 1 is a constant that does not depend on $\mathbf{n}$ (it does depend on N, of course). Since $p_N(\mathbf{n}(N))$ is an unnormalised probability distribution, the left-hand side is still an unnormalised probability distribution.

Term 3 comes from the fact that we are taking the discrete approximation to a normal distribution; this is the correct scaling for the number of points in this discrete approximation. Therefore, terms 3 and 4 together are the normal distribution that we would like to get.

Remark. One way to think about Term 3 is to observe that we would like the probability distribution of $\mathbf{u}(N)$ to be approximately normal, i.e. $e^{-u_r^2/2\overline{x}_r}$. Going from $p(\mathbf{n}(N))$ to $p(\mathbf{u}(N))$ will introduce a factor of $\sqrt{N}$, giving a term of the form $\sqrt{N/(2\pi n_r(N))}$. This converges to a constant, which joins the ranks of all the other normalising constants we are ignoring.

We will now show that term 2 has the effect of conditioning that normal distribution on the set $A_{\mathcal{B}}\mathbf{u} = 0$. For $j \in \mathcal{B}$, i.e. $B_j > 0$, we consider the quantity

$$\begin{aligned} m_j(N) = C_j(N) - \sum_r A_{jr}n_r(N) &= C_j(N) - \sum_r A_{jr}\left(\overline{x}_r(N) + u_r(N)N^{1/2}\right) \\ &= \underbrace{C_j(N) - \sum_r A_{jr}\overline{x}_r(N)}_{=0 \text{ for } N \text{ large enough}} - \left(\sum_r A_{jr}u_r(N)\right)N^{1/2}. \end{aligned}$$

Now,

$$\lim_{N\to\infty} \mathbb{P}(m_j(N) > \epsilon N^{1/2}) = 0$$

since the distribution $p_N(\mathbf{n}(N))$ is geometric with parameter $(1 - B_j) < 1$ in (unscaled) m_j. (We could take $\mathbb{P}(m_j(N) > f(N))$ for any $f(N) \to \infty$ as $N \to \infty$ here, and get the same result; but we only want it for $f(N) = \epsilon N^{1/2}$.) Consequently,

$$\lim_{N\to\infty} \mathbb{P}(|\sum_r A_{jr} u_r(N)| > \epsilon) = 0, \quad j \in \mathcal{B}$$

Therefore, the limiting $\mathbf{u}$ has $A_{\mathcal{B}}\mathbf{u} = 0$, as required.

Remark. To turn the end of the proof from handwaving into a solid proof, we need a tightness argument. That is, we have shown that $p_N(\mathbf{n}(N))$ has relative sizes that are appropriate to a normal distribution, but we can only assert this over compact sets, so we need to make sure that the probability mass doesn't leak to infinity. For this we will need the form of the error in Stirling's approximation.

□

Corollary 3.5. *For $r \in R$,*

$$L_r(N) \to L_r \equiv 1 - \prod_j (1 - B_j)^{A_{jr}}.$$

Here, $L_r(N)$ is the probability that a call on route r is lost (in network N).

Proof. By Little's law,

$$\underbrace{(1 - L_r(N))\nu_r(N)}_{\lambda} \cdot \underbrace{1}_{W} = \underbrace{\mathbb{E}[n_r(N)]}_{L}.$$

Therefore,

$$1 - L_r(N) = \frac{\mathbb{E}[n_r(N)]}{\nu_r(N)} = \frac{\mathbb{E}[n_r(N)]/N}{\nu_r(N)/N} \to \frac{\overline{x}_r}{\nu_r} = \prod_i (1 - B_i)^{A_{ir}}.$$

□

Example 3.4. Consider the system in Figure 20, with link-route incidence matrix

$$A = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}.$$

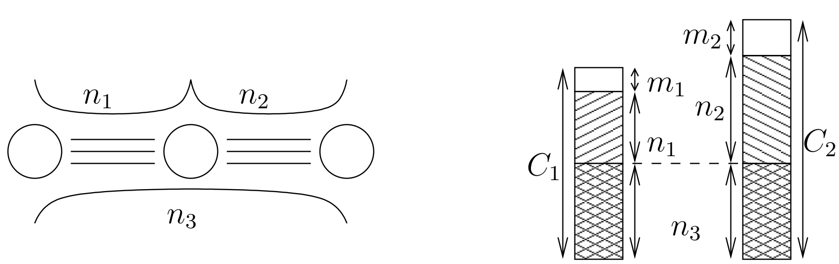


FIGURE 20. Shared communication link, and break-down of the number of circuits in use / free on each of the links

Suppose $\mathcal{B} = \{1, 2\}$, so $A_{\mathcal{B}} = A$. Then prior to conditioning on $A_{\mathcal{B}}\mathbf{u} = 0$ we have $u_r = \frac{n_r - \overline{x}_r}{\sqrt{N}} \to N(0, \overline{x}_r)$ independent normals. However, since $|\mathcal{B}| = 2$, the condition $A_{\mathcal{B}}\mathbf{u} = 0$ reduces the system to 1 dimension.

In the picture, the constraint $A_{\mathcal{B}}\mathbf{u} = 0$ means $m_1 = m_2 = 0$, and the single parameter is therefore n_3. (In reality, we will see that m_1 and m_2 are non-zero; in fact, they control the system. However, m_1 and m_2 are $O(1)$ while C_1 and C_2 are $O(N)$.)

3.7. **Erlang fixed point.** Consider the equations

$$E_j = \mathcal{E}\left((1 - E_j)^{-1} \sum_r A_{jr}\nu_r \prod_i (1 - E_i)^{A_{ir}}, C_j\right) \tag{12}$$

(this is the generalisation of the Erlang fixed point equations to matrices A that may not be 0-1). Our goal now is to show that there exists a unique solution to these equations, and that in the scaling regime that we're looking at it converges to the vector $\mathbf{B}$ coming from the maximal probability estimates.

Theorem 3.6. *There exists a unique solution $(E_1, \ldots, E_J) \in [0, 1]^J$ satisfying* (12).

Proof. We will prove this theorem by showing that we can rewrite the equations (12) as the stationary conditions for an optimization problem (with a unique optimum).

Define $U(y, C) : \mathbb{R}_+ \times \mathbb{Z}_+ \to \mathbb{R}_+$ by the implicit equation

$$U(-\log(1 - \mathcal{E}(\nu, C)), C) = \nu(1 - \mathcal{E}(\nu, C)).$$

That is, $U(y, C)$ is the *utilization* or mean number of circuits in use in the Erlang model, when the blocking probability is $E = 1 - e^{-y}$. (Note that choosing a value for the capacity C and for the blocking probability $1 - e^{-y}$ will automatically assign some value of ν!) Since both the utilization $\nu(1-\mathcal{E}(\nu, C))$ and the blocking probability $\mathcal{E}(\nu, C)$ are strictly increasing functions of ν, the function U is a strictly increasing function of y. Therefore, the function

$$\int_0^y U(z, C)dz$$

is a strictly convex function of y.

Consider now the optimization problem

$$\min \sum_r \nu_r e^{-\sum_j y_j A_{jr}} + \sum_j \int_0^{y_j} U(z, C_j)dz \tag{13}$$
$$\text{s.t. } \mathbf{y} \geq 0$$

Note that this problem looks a lot like (10), except that we have replaced the linear term in the objective by a strictly convex one. We call this problem the *revised DUAL.*

By strict convexity, revised DUAL (13) has a unique minimum. Differentiating, the stationary conditions for the unique minimum $\overline{\mathbf{y}}$ are

$$\sum_r A_{jr}\nu_r e^{-\sum_i \overline{y}_i A_{ir}} = U(\overline{y}_j, C_j), \quad j = 1, \ldots, J \tag{14}$$

Now suppose $\mathbf{E}$ solves the Erlang fixed point equations (12), and define y_j by $E_j = 1 - e^{-y_j}$ (i.e., $y_j = -\log(1 - E_j)$). We can rewrite (12) in terms of $\mathbf{y}$ as

$$\mathcal{E}(e^{y_j} \sum_r A_{jr}\nu_r e^{-\sum_i y_i A_{ir}}, C_j) = 1 - e^{-y_j}, \quad j = 1, \ldots, J$$

or, moving things from side to side and multiplying by ν,

$$\nu e^{-y_j} = \nu(1 - \mathcal{E}(e^{y_j} \sum_r A_{jr}\nu_r e^{-\sum_i y_i A_{ir}}, C_j)). \tag{15}$$

If we let

$$\nu = e^{y_j} \sum_r A_{jr}\nu_r e^{-\sum_i y_i A_{ir}},$$

then (15) will become precisely the statement of the stationary conditions (14). Since the solution to them is unique, we see that there is a unique solution to the Erlang fixed point equations.

Remark. Recall that in the definition of U we can prescribe one of y, a measure of blocking probability on the Erlang link, and ν, the arrival rate. Here we claim that if the arrival rate ν is as specified above, then the blocking probabilities, the associated y values, and the utilizations on the links will be such as to satisfy (14).

□

We will now show that the revised DUAL is asymptotically equal to the DUAL problem: we will show that $U(z, C_j) \approx C_j$.

Lemma 3.7.

$$U(y, C) = C - (e^y - 1)^{-1} + o(1)$$

as $C \to \infty$, uniformly over $y \in [a, b] \subset (0, \infty)$ (i.e. y on compact sets, bounded away from 0).

Proof. Consider an isolated Erlang link of capacity C being offered Poisson traffic at rate ν. Then

$$\pi_j = \frac{\nu^j}{j!}\left(\sum_{k=0}^{C} \frac{\nu^k}{k!}\right)^{-1}.$$

Let $\nu, C \to \infty$ with $C/\nu \to 1 - B$ for some $B > 0$. (That is, the capacity is smaller than the arrival rate by a constant factor, and the ratio gives B.) Then

$$\pi(C) = \frac{1}{1 + \frac{C}{\nu} + \frac{C(C-1)}{\nu^2} + \dots} \to B.$$

(Note that the denominator converges to $1/(1-(1-B))$.) Further, the expectation (with respect to π) of the number of free circuits is

$$\sum_{m=0}^{C} m\pi(C-m) = \pi(C)\left(0 + 1 \cdot \frac{C}{\nu} + 2 \cdot \frac{C(C-1)}{\nu^2} + \dots\right) \le$$

$$\pi(C) \sum_{m=0}^{\infty} m \frac{C^m}{\nu^m} = \pi(C) \frac{C/\nu}{(1 - C/\nu)^2} \to \frac{B(1-B)}{B^2} = \frac{1-B}{B}.$$

Indeed, we have monotone convergence: the elements of the series converge term-by-term upwards to the geometric bound. This guarantees that we have equality in the place of $\le$, that is,

$$\mathbb{E}_\pi[\text{number of free circuits}] \to \frac{1-B}{B} = \frac{e^{-y}}{1-e^{-y}} = (e^y - 1)^{-1}.$$

(We also see that the distribution of the number free circuits on the link converges to the geometric distribution, whose probability of being equal to 0 is the blocking probability B.) □

Corollary 3.8. *As $N \to \infty$, $E_j(N) \to B_j$. (As always, we are assuming that A has full rank J.) Here, $E_j(N)$ is the Erlang fixed point* (12)*, and B_j are the optimal dual variables of* (10)*, i.e. the parameters in the central limit theorem.*

Proof. The Erlang fixed point is the unique minimum of the revised DUAL objective function

$$\sum_r \nu_r(N) e^{-\sum_j y_j A_{jr}} + \sum_j \int_0^{y_j} U(z, C_j(N)) dz = N\left(\sum_r \nu_r e^{-\sum_j y_j A_{jr}} + \sum_j y_j C_j + o(1)\right).$$

That is, the revised DUAL objective, up to scaling, converges uniformly to the DUAL objective of (10). Since the revised DUAL objective is strictly convex and continuous, its unique minimum converges to the unique minimum of the DUAL, as required. □

3.8. **Miscellaneous discussion.**

3.8.1. *Diverse routing.* The Erlang fixed point also arises as a limit under *diverse routing*: consider a limit in which the network becomes more diverse, i.e.

$$\mathbb{P}(\text{two flows through a single link in the network share another link elsewhere}) \to 0.$$

For example, we could consider the star network with n, the number of leaves, tending to ∞.

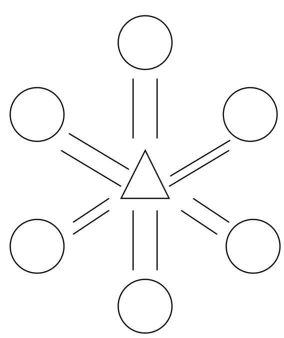


FIGURE 21. Star topology

(Recall that in the star network, calls are placed from one leaf to another, and so each call takes up a circuit on two links.) We will not scale the capacities on the links. We would expect that in this case the probability that two calls sharing one link actually share both of them tends to 0 as $n \to \infty$, although showing this precisely is quite difficult because the calls aren't independent.

This form of limit, when defined (which is the hardest part about it), leads to an Erlang fixed point approximation. This is unsurprising, since the assumption of diverse routing leads quite naturally to links becoming less dependent (and the Erlang fixed point approximation came from considering the links as independent of each other).

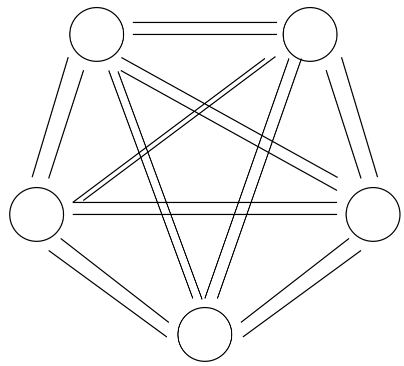


FIGURE 22. Complete graph topology

3.8.2. *Non-uniqueness.* Consider a network shaped as a complete graph, and entirely symmetric: that is, arrival rate between each pair of nodes is ν, and the number of circuits on each of the links is C. The number of nodes in the network is n.

We will not look at this network with fixed routing; instead, calls will be routed directly if possible, and otherwise we will try a randomly chosen 2-link alternative route. If that route happens to be blocked, the call is lost.

We will suppose that this model has an Erlang-like fixed point approximation, resulting from treating the blocking probabilities on the different links as independent. In that case, if B is the blocking probability on a link (the same on all of them, by symmetry), then

$$\mathbb{P}(\text{incoming call is accepted}) = \underbrace{(1-B)}_{\text{can route directly}} + \underbrace{B(1-B)^2}_{\text{can't route directly, but can via 2-link detour}}$$

and therefore the expected number of circuits per link that are busy is $\nu(1-B)+2\nu B(1-B)^2$. Thus, we look for a solution to

$$B = \mathcal{E}(\nu(1+2B(1-B)), C) \tag{16}$$

(indeed, if that is the arrival rate on the link, and the blocking probability is B, then the number of circuits that are busy will be as specified above).

Remark. An alternative way to derive this arrival rate is to just count the calls that come for link ij: we have an arrival rate of ν for the calls $i \leftrightarrow j$, plus for each other node k we have a traffic of $\nu B(1-B) \cdot \frac{1}{n-2}$ for calls $i \leftrightarrow k$ which get rerouted via j (and the same for calls $j \leftrightarrow k$ which get rerouted via i). Adding these $2(n-2)$ terms gives the result.

Suppose that $\nu, C \to \infty$ while keeping their ratio fixed. Then

$$\lim_{N\to\infty} \mathcal{E}(\nu N, CN) = (1 - C/\nu)^+ \quad \text{where } x^+ = \max(x,0) \text{ is the positive part of } x.$$

The fixed point equation (16) therefore simplifies to

$$B = \left[1 - C/\nu(1+2B(1-B))^{-1}\right]^+,$$

which can be solved directly. (The original equation expresses B as a ratio of two polynomials, but of quite high degree, so it's easier to solve it numerically.) The solution looks as follows:

Observe that in some regimes we have multiple solutions (in the case $C = \infty$, we have a cubic equation for B, which has multiple roots). The simulations of this system show

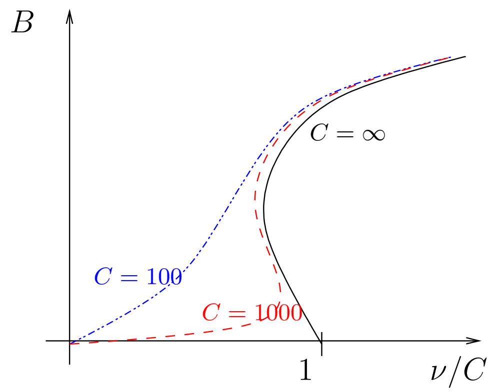


FIGURE 23. Blocking probabilities from the fixed point formula for different values of C

hysteresis: We would expect the system to follow the bottom solution to B until that has a

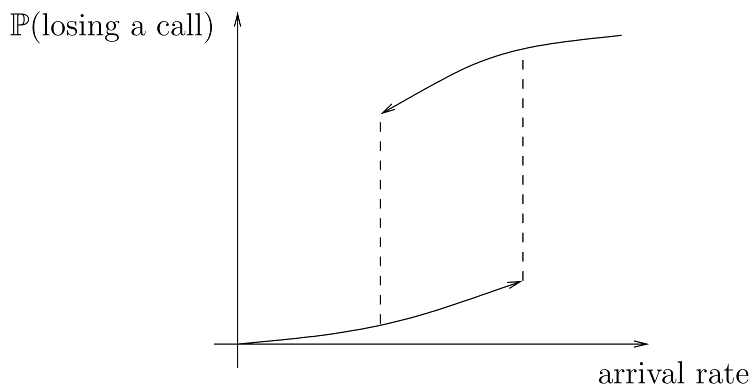


FIGURE 24. Empirical probabilities of losing a call in simulations

finite derivative, then jump to the upper branch, and then follow that while it has a finite derivative.

As we mentioned in the introductiory lectures of the course, while of course if we leave this finite-state Markov chain around for long enough, it will have a unique stationary distribution, the time over which it converges to it will grow larger as the network grows larger.

Another way to see that we should expect non-uniqueness is as follows. The fixed point equations (16) locate the stationary points of

$$\nu(e^{-y} + e^{-2y} \underbrace{(1 - 2/3e^{-y})}_{\text{non-convex!}} + \int_0^y U(z, C)dz)$$

(you will show this in Exercise 2.9). Since this is a non-convex function, changing the parameters slightly can change the location of the stationary points quite a lot:

Remark. If we allow more call repacking, the problem becomes *worse*, driving the system utilization down by as much as a factor of 1/2!

How can we control this given that we *do* want to allow rerouting of calls? (We want to allow call repacking because ν and C may not be matched, and also because some of the communication links might fail.)

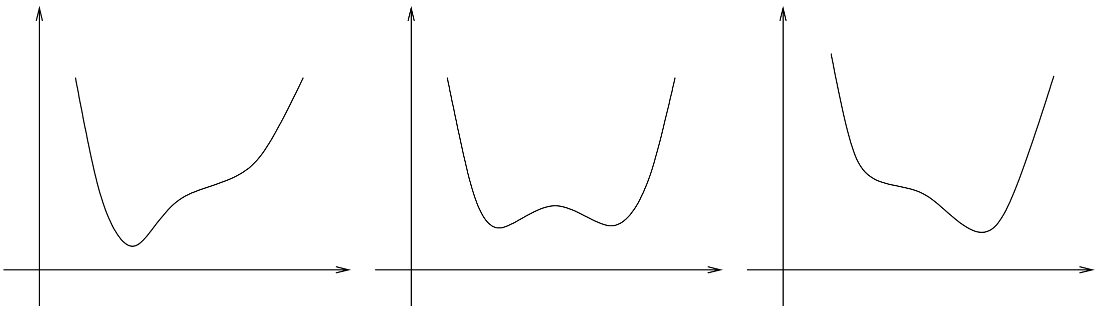


FIGURE 25. Minimum of a non-convex function can jump abruptly as the function varies smoothly.

Idea: "trunk reservation". A link will accept an alternatively routed call only if there are $\geq s$ circuits on the link, where $s \ll C$ is a small constant (typically, $s = 5$ for $C = 100, 1000$).

Then we use the following *sticky random scheme*: a call $i \to j$ is routed directly if there is a free circuit on the link between them. Otherwise, we try to reroute the call via a tandem node $k(i,j)$ (which is stored at i). If trunk reservation allows the rerouting, the call is accepted. If not, the call is lost, and the tandem node $k(i,j)$ is reset randomly. The intuition is that the system will find links with spare capacity in the network, but then once it uses up that spare capacity, will go and look for a different link.

4. Decentralized optimization

4.1. A game.

Consider the following game. You have a graph G, and you perform a random walk on G until you hit a subset $S \subset G$. When you hit $i \in S$ you receive a reward v_i, and the game ends. How much should you pay to play this game? (I.e., what is your expected reward?)

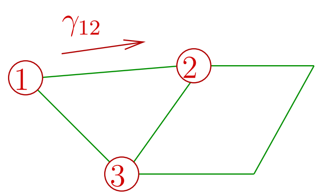


FIGURE 26. A game of a random walk on a graph with transition rates γ_{ij}

Clearly, the answer depends on the starting position, so let p_j be the expected reward starting from j. (If $j = i \in S$ then, of course, $p_j = v_i$.) Let $\gamma_{ij} = \gamma_{ji}$ be the transition rates for going $i \leftrightarrow j$ (so the random walk is reversible, with the uniform invariant distribution). Then:

$$p_j = \sum_i \frac{\gamma_{ji}}{\sum_k \gamma_{jk}} p_i \quad \text{for } j \in G \setminus S$$

We can rewrite this set of equations as follows:

$$0 = \sum_i \overbrace{\underbrace{\gamma_{ij}}_{1/R} \underbrace{(p_i - p_j)}_{\Delta V}}^{\text{current}}, \qquad j \in G \setminus S$$

$$p_j = v_j, \quad j \in S$$

Observe that these are Kirchhoff's equations for an electrical network G in which i and j are joined by a resistance of γ_{ij}^{-1}, and j is held at potential v_j.

Why is this game connected to current flows?

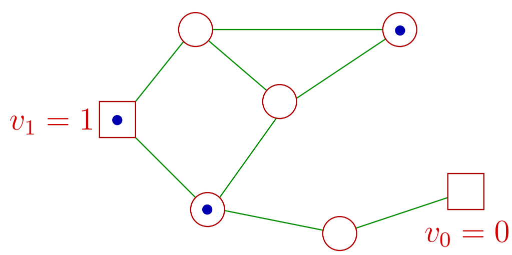


FIGURE 27. Button model

4.2. **Button model.** Suppose that on every node of the graph there is a button. Buttons on nodes j and k swap at rate γ_{jk}. Buttons are painted blue as they pass through node v_1 (which, if you're reading this in colour, has a blue dot at the centre), and are painted white as they pass through v_0.

This is a Markov process; the state is the set of buttons coloured blue.

For a given button, asking the question "am I blue?" amounts to asking whether the random walk that it has been performed has most recently been to v_1 or to v_0. Since the random walk is reversible, this is equivalent to asking whether from now forward it will visit v_1 or v_0 first. Thus, each button can be seen to be playing the game as above.

On the other hand, we can construct an equivalent model for this Markov process. Suppose that electrons perform a random walk with exclusion, i.e. attempts to jump to a node that is already occupied by an electron will be blocked. Suppose also that electrons are pushed in at v_1 (i.e., as soon as an electron leaves v_1, another one appears there) and pulled out at v_0 (i.e., as soon as an electron appears there, it is removed from the system). This is a rather simple model of electron movement in an electrical network.

Exercise 5. Convince yourself that these two models really are equivalent! (The Markov process has as its state the set of blue nodes in the first case, and the set of occupied nodes in the second. The swap of a blue and a white button corresponds to an electron moving from one state to another; the swap of a blue and a blue button doesn't change the state of the Markov process, and in the electron model corresponds to a jump being blocked.)

Let p_j be the (stationary) probability that the node j is occupied. Then

$$p_0 = v_0 = 0, \quad p_1 = v_1 = 1, \quad p_j = \sum_i \frac{\gamma_{ji}}{\sum_k \gamma_{jk}} p_i, \; j \neq 0, 1$$

Equivalently,

$$0 = \sum_i \gamma_{ij}(p_i - p_j), \qquad j \neq 0, 1$$

$$p_j = v_j, \quad j = 0, 1$$

which are precisely Kirchhoff's equations for an electrical network in which nodes i and j are joined by a resistance γ_{ij}^{-1} and nodes $0, 1$ are held at a voltage of v_0, v_1.

We can compute the *net* flow of electrons from site j to site k as

$$\gamma_{jk}\mathbb{P}(j \text{ occupied, } k \text{ empty}) - \gamma_{kj}\mathbb{P}(k \text{ occupied, } j \text{ empty})$$

Recall $\gamma_{jk} = \gamma_{kj}$. Also,

$$\begin{aligned}\mathbb{P}(j \text{ occupied, } k \text{ empty}) - \mathbb{P}(k \text{ occupied, } j \text{ empty}) = \\ (\mathbb{P}(j \text{ occupied, } k \text{ empty}) + \mathbb{P}(j \text{ occupied, } k \text{ occupied})) - \\ (\mathbb{P}(j \text{ occupied, } k \text{ occupied}) + \mathbb{P}(k \text{ occupied, } j \text{ empty})) \\ = p_j - p_k\end{aligned}$$

and therefore the net flow of electrons is

$$\gamma_{jk}(p_j - p_k) = \frac{p_j - p_k}{r_{jk}},$$

i.e. we have recovered Ohm's law that the current flow from j to k is proportional to the voltage difference (the constant of proportionality being γ_{jk}^{-1}).

This was a steady-state computation; we could produce a precise result that would tell us that in the long run the number of electrons that have moved from j to k will converge to this value, and even give a central limit theorem for the "shot noise" around the value.

4.3. **Extremal characterization.** We give here another angle of attack on the current flow in networks, which was developed in the late 19th century.

Let u_{jk} be the current flowing from j to k. Then the heat dissipation in the network is

$$\frac{1}{2}\sum_j \sum_k r_{jk} u_{jk}^2$$

(the 1/2 is there because we are double-counting each link). Suppose we want to minimize the heat dissipation subject to a given total current:

$$\min \frac{1}{2}\sum_j \sum_k r_{jk} u_{jk}^2$$

$$\text{over } u_{jk} = -u_{kj}, \quad j,k \in G$$

$$\text{subject to } \sum_k u_{jk} = \begin{cases} 0, & j \in J \\ -U, & j = 0 \\ +U, & j = 1 \end{cases}$$

The Lagrangian is

$$\mathcal{L}(\mathbf{u};\mathbf{p}) = \frac{1}{2}\sum_{j,k} r_{jk} u_{jk}^2 + \sum_j p_j \left(\sum_k u_{jk}\right) + p_0 U - p_1 U$$

where we abuse notation and call the Lagrange multipliers p_j. We haven't introduced a Lagrange multiplier for the second constraint; instead, we use it to eliminate u_{jk} with $j > k$ from the equations, leaving only u_{jk} with $j < k$. (In particular, the summation in the first term is over $j < k$; this corresponds to dividing the original objective function by 2.) Differentiating with respect to u_{jk},

$$\frac{\partial \mathcal{L}}{\partial u_{jk}} = r_{jk} u_{jk} + p_j - p_k,$$

so the solution is

$$u_{jk} = \frac{p_k - p_j}{r_{jk}}$$

That is, the Lagrange multipliers really are potentials in Kirchhoff's equations, and the currents u_{jk} obey Ohm's law with these potentials.

This is known as *Thomson's principle*: the flow pattern of current within a network of resistors is such as to minimize the heat dissipation for a given total current. (The Thomson in question is the physicist William Thomson, later Lord Kelvin.)

If we consider the dual of this optimiation problem, we will arrive at

$$\begin{gathered}\min \frac{1}{2}\sum_j \sum_k \gamma_{jk}(p_j - p_k)^2 \\ \text{over } p_j, \quad j \in G \\ \text{subject to } p_0 = 0,\ p_1 = 1\end{gathered}$$

This gives the optimality conditions

$$\frac{\partial \mathcal{L}}{\partial p_j} = \sum_k \gamma_{jk}(p_j - p_k) = 0, \quad j \neq 0, 1.$$

You may recognise the problem of minimising a quadratic form as the Dirichlet problem. (To get the classical Dirichlet problem on continuous spaces, imagine a network that is a square grid, and allow the grid to get finer and finer.)

The extremal characterization is extremely useful in answering the following question. Suppose that we remove an edge out of the network (i.e., assign an infinite resistance to it). What will this do to the equivalent resistance of the network? Intuitively, it is clear that it must increase, but how can we prove it?

In the extremal characterization, this is easy: the minimal heat dissipation can only increase if we enforce $u_{jk} = 0$; thus, the equivalent resistance must go up (or not go down, anyway). It is quite tricky to prove this result without using the extremal characterization.

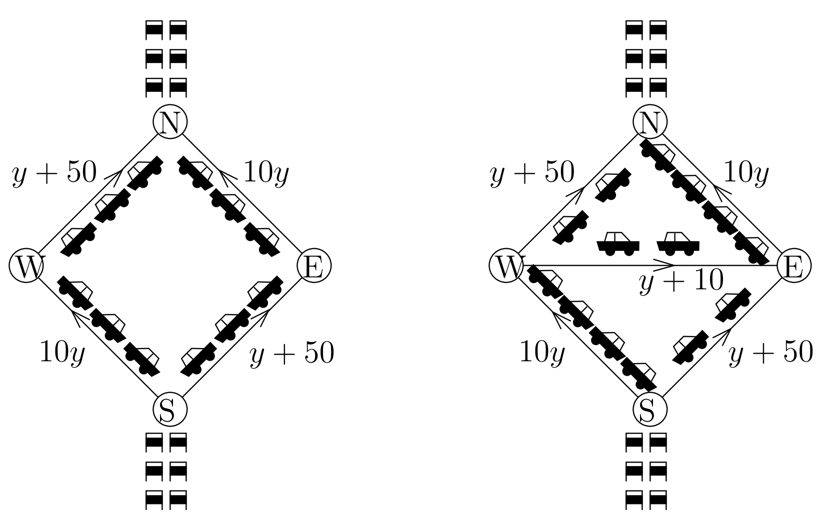


FIGURE 28. Braess's paradox. The addition of a link causes everyone's journey time to lengthen.

4.4. **Braess's paradox.** Consider Figure 28. We think of the cars in this road network. The functions next to the roads are the travel times, as functions of traffic. Note that all routes $S \to N$ have the same delay, so no drivers have an incentive to switch routes. Between the left- and the right-hand diagram we have added an extra road; and the delay for everyone in the system went from 83 to 92. Thus, if we block the central road in the right-hand network, everyone's delay will decrease! (Compare this with the electrical network, where removing a link *increases* the equivalent resistance.)

Let J be the set of directed links (we allow for the possibility of two one-way links going in opposite directions); $R \subset 2^J$ is the set of possible routes (a route is a subset of links); and let A be the link-route incidence matrix, so $A_{jr} = 1$ if $j \in r$ and $A_{jr} = 0$ otherwise.

Let x_r be the flow of route r, and $\mathbf{x} = (x_r, r \in R)$ the vector of flows. Then the flow on a link is

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

or equivalently, $\mathbf{y} = A\mathbf{x}$.

The delay that is incurred on a single link j is given by a function $D_j(y_j)$, which we assume to be continuously differentiable, and increasing. (We would also expect it to be convex.) E.g. see Figure 29.

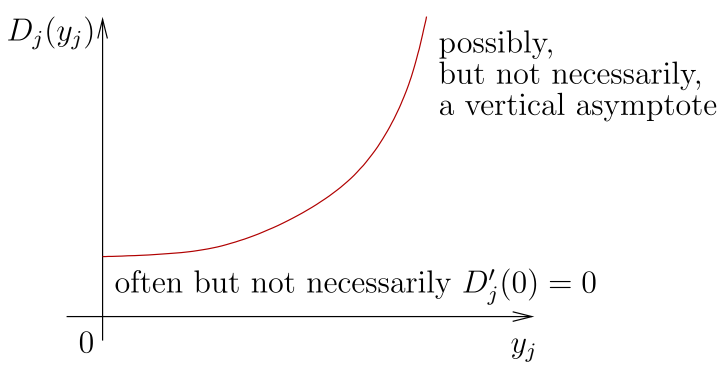


FIGURE 29. Possible delay as a function of traffic on a link

Finally, let S be the set of source-destination pairs, and let $H_{sr} = 1$ if the source-destination pair s is served by route r and $H_{sr} = 0$ otherwise. Let $H = (H_{sr}, s \in S, r \in R)$. Then for source-destination pair s the flow f_s is given by

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

or equivalently, $\mathbf{f} = H\mathbf{x}$.

Example 4.1. Consider the network in Figure 30.

We will take it to have the link-route incidence matrix

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

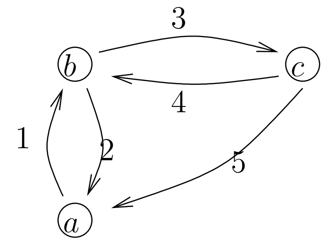


FIGURE 30. Example network

(Note that we are omitting the two-link route $b \to a$ along links 3 and 5; we could include it, but we don't have to.) The corresponding matrix H is

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

The conditions for a stable flow are of the form

$$x_r > 0 \implies \sum_{j \in J} D_j(y_j) A_{jr} = \min_{r' \in S(r)} \sum_{j \in J} D_j(y_j) A_{jr'}$$

where $S(r)$ is the set of routes serving the same source-destination pair as r.

Definition. A *Wardrop equilibrium* is a vector of flows along routes $\mathbf{x} = (x_r, r \in R)$ such that

$$x_r > 0 \implies \sum_{j \in J} D_j(y_j) A_{jr} = \min_{r' \in S(r)} \sum_{j \in J} D_j(y_j) A_{jr'},$$

where $\mathbf{y} = A\mathbf{x}$.

Theorem 4.1. *A Wardrop equilibrium exists.*

Proof. Consider the optimization problem

$$\begin{aligned} &\min \sum_{j \in J} \int_0^{y_j} D_j(u) du \\ &\text{over } \mathbf{x} \geq 0, \ \mathbf{y} \\ &\text{s.t. } H\mathbf{x} = \mathbf{f}, \ A\mathbf{x} = \mathbf{y}. \end{aligned}$$

The feasible region is convex, and the objective function is convex and differentiable (because D_j is an increasing, continuous function). Thus, an optimum exists and can be found by the Lagrangian techniques. The Lagrangian for the problem is

$$\mathcal{L}(\mathbf{x}, \mathbf{y}; \boldsymbol{\lambda}, \boldsymbol{\mu}) = \sum_{j \in J} \int_0^{y_j} D_j(u) du + \boldsymbol{\lambda} \cdot (\mathbf{f} - H\mathbf{x}) - \boldsymbol{\mu} \cdot (\mathbf{y} - A\mathbf{x})$$

To minimize, we differentiate:

$$\frac{\partial \mathcal{L}}{\partial y_j} = D_j(y_j) - \mu_j \quad \frac{\partial \mathcal{L}}{\partial x_r} = -\lambda_{s(r)} + \sum_j \mu_j A_{jr}.$$

We want to find a minimum over $y_j \in \mathbb{R}$ (note $y_j \geq 0$ will be forced by $A\mathbf{x} = \mathbf{y}$) and $x_r \geq 0$. Therefore, at the minimum we have $\mu_j = D_j(y_j)$ is the delay on link j, and also

$$\lambda_{s(r)} \begin{cases} = \sum_j \mu_j A_{jr}, & x_r > 0 \\ \leq \sum_j \mu_j A_{jr}, & x_r = 0 \end{cases}.$$

That is, we can interpret $\lambda_{s(r)}$ as the minimal delay available to the source-destination pair $s(r)$.

Thus, a solution to this optimization problem must give a Wardrop equilibrium. □

Remark. Is the Wardrop equilibrium unique? No: if D_j are all strictly increasing, then we get a unique optimal $\mathbf{y}$ (link flows), but there is no reason to expect uniqueness in $\mathbf{x}$. For example, in the network in Figure 31 we clearly can shift traffic between the solid and the dashed routes.

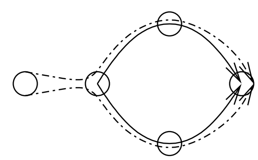


FIGURE 31. This network has many traffic patterns $\mathbf{x}$ giving the same $\mathbf{y}$

Note that if we remove a link (road) from the network, we must increase $\sum_j \int_0^{y_j} D_j(y_j)$, but this doesn't tell us whether the individual delays increase or decrease. This is because this is a somewhat unnatural objective function: the natural societal objective would be

$$\begin{aligned} &\min \sum_{j \in J} y_j D_j(y_j) \\ &\text{over } \mathbf{x} \geq 0,\ \mathbf{y} \\ &\text{s.t. } H\mathbf{x} = \mathbf{f},\ A\mathbf{x} = \mathbf{y}. \end{aligned}$$

Here, $\sum_j y_j D_j(y_j)$ is the rate at which total delay increases (the total delay of all the users in the system grows to infinity as more users pass through the system, so it is more natural to consider its rate of increase).

In this case, we have

$$\mathcal{L}(\mathbf{x}, \mathbf{y}; \boldsymbol{\lambda}, \boldsymbol{\mu}) = \sum_{j \in J} y_j D_j(y_j) + \boldsymbol{\lambda} \cdot (\mathbf{f} - H\mathbf{x}) - \boldsymbol{\mu} \cdot (\mathbf{y} - A\mathbf{x}),$$

$$\frac{\partial \mathcal{L}}{\partial y_j} = D_j(y_j) + y_j D_j'(y_j) - \mu_j \quad \frac{\partial \mathcal{L}}{\partial x_r} = -\lambda_{s(r)} + \sum_j \mu_j A_{jr}.$$

If we interpret $y_j D_j'(y_j)$ as the congestion toll that the users of link j pay, then μ_j is their total cost from both the delay and toll, and $\lambda_{s(r)}$ is the minimal cost available to source-destination pair $s(r)$. That is, by adding tolls on the links, we can encourage the users to a more socially optimal behaviour.

4.5. **Gallagher's algorithm.** Consider an open network of $\cdot/M/1$ queues (i.e., the service times are exponential, and the arrival process is whatever it happens to be). Let ϕ_j be the service rate at queue j, and let ν_r be the arrival rate of customers on route r.

The mean sojourn time in queue j is $\frac{1}{\phi_j-\lambda_j}$, where $\lambda_j = \sum_{r:j\in r}\nu_r$. (This is true whenever we have a product form stationary distribution.)

Suppose that a unit delay of a customer on route r costs w_r; then the mean cost per unit time is

$$W(\boldsymbol{\nu},\boldsymbol{\phi}) = \sum_r w_r \sum_{j:j\in r} \frac{\nu_r}{\phi_j - \sum_{r':j\in r'}\nu_{r'}}$$

Now,

$$\frac{\partial W}{\partial \nu_r} = \sum_{j\in r}\Big(\underbrace{\frac{w_r}{\phi_j-\lambda_j}}_{\substack{\text{direct cost to queue } j\\ \text{of inserting an extra customer on route } r}} + \underbrace{\sum_{r':j\in r'}\frac{\nu_{r'}w_{r'}}{(\phi_j-\lambda_j)^2}}_{\substack{\text{knock-on cost, or externality}\\ \text{of inserting an extra customer on any route } r\\ \text{passing through } j}} \Big)$$

Gallagher's algorithm involved measuring delays in the network and shifting traffic in the natural direction according to these derivatives.

Remark. Recall that in a loss network we derived the DUAL optimization problem

$$\min \underbrace{\sum_r \nu_r e^{-\sum_j y_j A_{jr}}}_{\text{carried traffic}} + \sum_j y_j C_j$$
$$\text{s.t. } \mathbf{y}\geq 0$$

That is, the optimization problem depends on the variable we are interested in, but with the wrong sign! The trunk reservation, sticky random scheme attempts to get the system objective function to line up more closely with the natural objective.

5. Random access networks

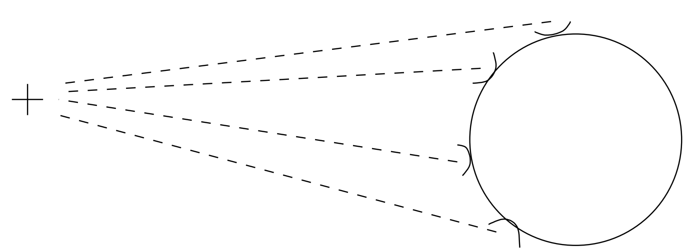


FIGURE 32. Multiple base stations contacting a satellite

Suppose time is slotted, and let $Z_t = 0$, 1, or $*$ according to whether 0, 1, or more than one transmission attempts are made duting the time slot $(t, t+1)$.

The packet arrivals are a Poisson process of rate ν, and every arrival is at a new station (i.e. all packets compete for the channel – we cannot queue the packets).

Let Y_t be the number of new packets arriving during $(t-1, t)$. All of these will first attempt transmission during $(t, t+1)$ (along with, possibly, retransmission attempts). Transmission attempts during $(t, t+1)$ are unsuccessful if there are more than one of them, i.e. if $Z_t = *$.

Remark. The fundamental issue being addressed here is the effect of speed-of-light delays on scheduling: what is happening in one part of the network cannot be known elsewhere in the network for some time.

Remark. The model above assumes that messages are of a single packet. If instead messages are of length M, then we can use the first packet of the message to reserve the channel for the rest of the message. Thus, if stable throughput of η is attainable for $M = 1$, then a stable throughput of $\frac{M}{\eta^{-1}+M-1}$ is achievable for bigger M. This is good news, provided $\eta > 0$.

5.1. **ALOHA model.** After an unsuccessful transmission attempt, the retransmission is delayed by a geometrically distributed random variable with mean f^{-1}, independently of everything else in sight. Then,

$$A_t = \text{number of attempts in } (t, t+1) = Y_t + \text{Binom}(N_t, f),$$

where N_t is the backlog of packets awaiting retransmission, which evolves as

$$N_{t+1} = N_t + Y_t - I[Z_t = 1].$$

Here,

$$Z_t = \begin{cases} 0, & A_t = 0 \\ 1, & A_t = 1 \\ *, & A_t > 1 \end{cases}$$

With a Poisson arrival process, Y_t is a Poisson random variable, so

$$\mathbb{P}(Z_t = 1 | N_t = n) = \underbrace{e^{-\nu} n f (1-f)^{n-1}}_{Y_t = 0,\text{ one retransmission}} + \underbrace{\nu e^{-\nu}(1-f)^n}_{Y_t = 1,\text{ no retransmissions}}$$

Let us compute the drift

$$\mathbb{E}[N_{t+1} - N_t | N_t = n] = \nu - \mathbb{P}(Z_t = 1 | N_t = n)$$

Thus, the drift is positive if

$$\nu > e^{-\nu}(nf + (1-f)\nu)(1-f)^{n-1} \to 0 \text{ as } n \to \infty.$$

This suggests (but doesn't prove) that N_t is a transient Markov chain (it will at some point get large, and then will drift off to infinity).

Remark. The drift condition does not prove that N_t is transient. Indeed, consider the Markov chain where from each state n there are two possible transitions: to $n+2$ with probability $1 - 1/n$, and to 0 with probability $1/n$. Then the drift is $1 - 2/n \to 1$, but $\mathbb{P}(\text{hit 0 at some point}) = 1$ so the chain is recurrent. In our process, the jumps are constrained; e.g., $\mathbb{E}[\text{jump}^2] < \infty$. This together with the drift condition can be turned into a proof that the Markov chain is transient, but we will pursue a different method.

Let

$$\begin{aligned} p(n) &= \mathbb{P}(\exists T < \infty : N_1, \ldots, N_T = n, Z_T = 0 \text{ or } 1 | N_1 = n) \\ &= \mathbb{P}(\text{channel unjams before backlog increases} | N_1 = n). \end{aligned}$$

To compute $p(n)$, note that if $Z_t = 0$ or 1 then $T = t$ and we "win"; if $Z_t = *$ and $N_{t+1} > n$ then no such T exists and we "lose"; and if $Z_t = *$ and $N_{t+1} = n$ then we "try again". Thus,

$$\begin{aligned} p(n) &= \frac{\mathbb{P}(\text{win})}{\mathbb{P}(\text{win}) + \mathbb{P}(\text{lose})} = \frac{\mathbb{P}(Z_t = 0 \text{ or } 1 | N_t = n)}{1 - \mathbb{P}(N_{t+1} = n, Z_t = * | N_t = n)} \\ &= \frac{e^{-\nu}(1+\nu)(1-f)^n + e^{-\nu} n f (1-f)^{n-1}}{1 - e^{-\nu}(1 - (1-f)^n - nf(1-f)^{n-1})} \\ &\sim \frac{nf(1-f)^{n-1}}{e^{\nu} - 1} \text{ as } n \to \infty \end{aligned}$$

In particular, $\sum_{n=0}^{\infty} p_n < \infty$.

Theorem 5.1 (First Borel-Cantelli Lemma)**.** *If (A_n) is a sequence of events such that $\sum_{n=1}^{\infty} \mathbb{P}(A_n) < \infty$ then the probability that infinitely many of the events occur is 0.*

Proof.

$$\mathbb{E}[\text{number of events occurring}] = \mathbb{E}[\sum_n I[A_n]] = \sum_n \mathbb{E}[I[A_n]] = \sum_n \mathbb{P}(A_n) < \infty.$$

Since the number of events occurring has finite expectation, it must be finite with probability 1. □

This together with the estimate on $\sum p(n)$ is almost good enough, but what are the events A_n?

Set $R(1) = 1$ and let $R(r+1) = \inf\{t > R(r) : N_t > N_{R(r)}\}$. These are the times when the process hits record values.

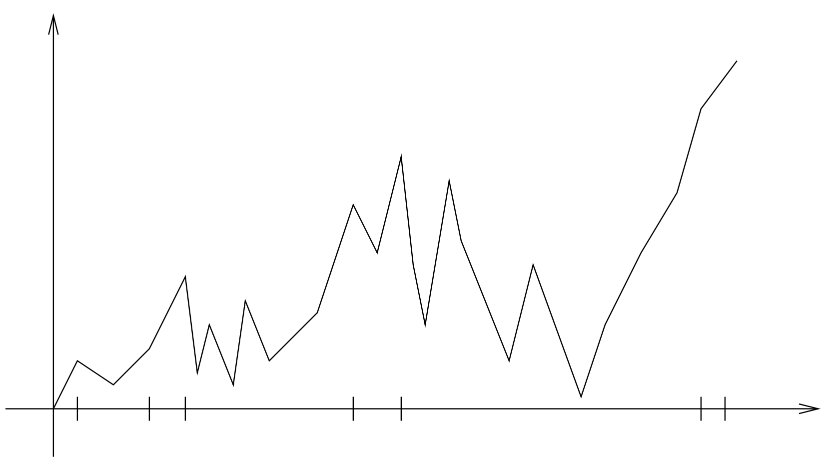


FIGURE 33. Times when the process hits record values.

With probability 1, this defines an infinite increasing sequence, and necessarily

$$\sum_{r=1}^{\infty} p(N_{R(r)}) \le \sum_{n=0}^{\infty} p(n) < \infty.$$

Thus, by the Borel-Cantelli Lemma, the channel will only unjam after finitely many record values, i.e. there will be a finite total number of successful transmissions. Another way of saying this is

$$\mathbb{P}(\exists J < \infty : Z_t = * \ \forall t \geq J) = 1.$$

Remark. J is not a stopping time, i.e. we cannot tell when we've got to it. However, (almost) every sample path will eventually jam all the time.

Remark. In a real system, the number of stations will be large but finite, say M. The number of new arrivals will in that case have a binomial rather than exponential distribution, $Y_t \sim \text{Binom}(M - N_t, q)$; M large means that $Mq \gg 1$. In that case, the equilibrium distribution of the backlog has the following form:

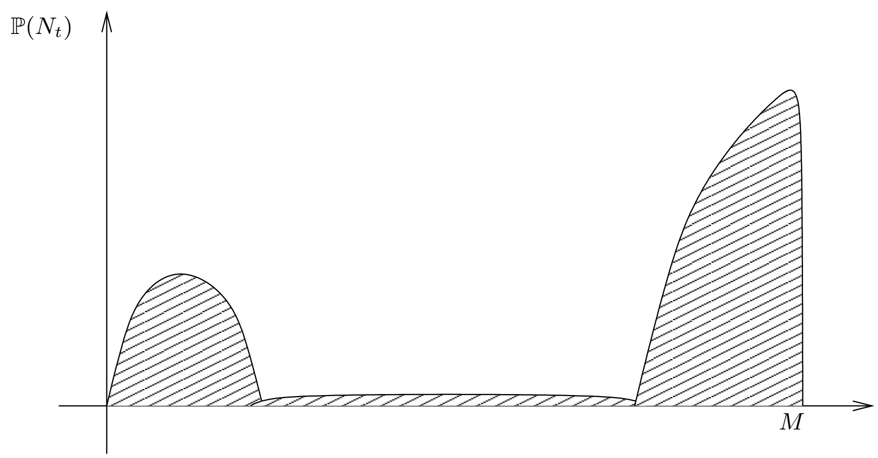


FIGURE 34. Equilibrium distribution of the backlog

Qualitatively, this is the same behaviour as in the infinite system, although of course if we wait over a long enough time period, the channel will in fact unjam.

What is the intrinsic problem?

The probability of a single retransmission attempt given that the backlog is $N_t = n$ is $nf(1-f)^{n-1}$. If we maximize this with respect to f, we will get

$$0 = \frac{\partial}{\partial f} = n(1-f)^{n-1} - n(n-1)f(1-f)^{n-2} \implies f = \frac{1}{n}$$

in which case the probability of a single retransmission attempt is $(1 - \frac{1}{n})^{n-1} \to e^{-1}$ as $n \to \infty$.

That is, there isn't an intrinsic reason why stations transmitting independently at random couldn't have a positive throughput, but the stations would need to know the value of n, the backlog.

Suppose that the stations can observe the channel state $Z_1, Z_2, \ldots$, and maintain a counter S_t, e.g.,

$$S_{t+1} = \begin{cases} S_t - 1, & Z_t = 0 \\ S_t, & Z_t = 1 \\ S_t + 1, & Z_t = * \end{cases}$$

or more generally,

$$S_{t+1} = \max(1, S_t + aI[Z_t = 0] + bI[Z_t = 1] + cI[Z_t = *]).$$

(This is maintained by all stations, including ones that do not yet have a packet to transmit.) Suppose then that the stations that have a packet to transmit or retransmit will do so with probability $f = 1/S_t$. We would expect this to work if S_t tracks N_t reasonably well, at least when N_t is large. (Here, N_t is the number of stations with packets to transmit or retransmit – we are not distinguishing between new and old arrivals.)

In this model, N_t alone and S_t alone are not Markov chains, but (N_t, S_t) is a Markov chain. Moreover,

$$\mathbb{E}[S_{t+1} - S_t | N_t = n, S_t = s] = a(1-\frac{1}{s})^n + b(\frac{n}{s})(1-\frac{1}{s})^{n-1} + c\left(1 - (1-\frac{1}{s})^n - (\frac{n}{s})(1-\frac{1}{s})^{n-1}\right)$$

at least for s large enough (so that the $max(1, \cdot) = \cdot$). If we look at n, s large with $n/s = \kappa$, then

$$\mathbb{E}[S_{t+1} - S_t | N_t = n, S_t = s] \to (a - c)e^{-\kappa} + (b - c)\kappa e^{-\kappa} + c \tag{17}$$

If

$$(a, b, c) = (2 - e, 0, 1) \text{ or } (1 - e/2, 1 - e/2, 1) \text{ or many other choices,}$$

then the drift (17) is negative when $\kappa > 1$ and positive when $\kappa < 1$ (check!). That is, the phase portrait of N vs. S looks like the left-hand side of Figure 35.

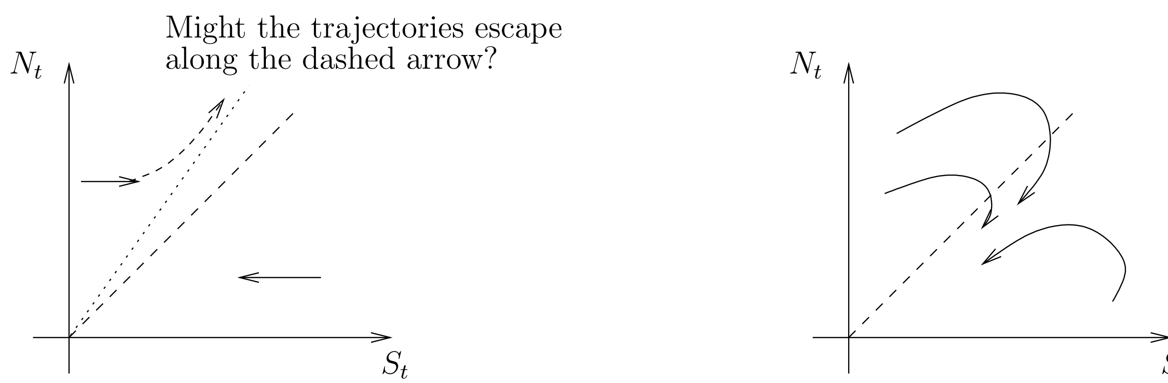


FIGURE 35. Phase portrait of N_t and S_t.

We need to make sure that the trajectories starting with $\kappa > 1$ actually reach the diagonal – it is possible that as S_t increases, N_t increases even further, and asymptotically κ is bounded by some constant $\kappa' > 1$ (see the dashed trajectory in the figure). For that, we must make sure that when $\kappa > 1$ and S is increasing, N doesn't increase too quickly with it. We look at the drift of N:

$$\mathbb{E}[N_{t+1} - N_t | N_t = n, S_t = s] = \nu - \frac{n}{s}(1 - \frac{1}{s})^{n-1} \to \nu - \kappa e^{-\kappa}, \tag{18}$$

and the condition we would like is

$$\frac{\nu - \kappa e^{-\kappa}}{(a - c)e^{-\kappa} + (b - c)\kappa e^{-\kappa} + c} < \kappa \text{ when } \kappa > 1.$$

We can check that this holds for all of the above examples of (a, b, c), provided $\nu < e^{-1}$ (and we wouldn't expect to be able to accommodate $\nu > e^{-1}$ anyway).

Remark. This isn't a proof of the recurrence of the Markov chain (N_t, S_t) for two reasons. First, for a proof we would need to be a bit less handwavy with the phase portrait. (For example, if you look at the trajectories, they have a nasty habit of heading away from the origin before converging to diagonal and heading inwards.) Second, we would need some sort of general theory for converting the drift analysis into a proof of recurrence. You can find more details in Bruce Hajek's notes, `http://www.ifp.illinois.edu/~hajek/Papers/networkanalysis.html`. (If you look on our course website, these notes go under the code name *CNA* for "Communication network analysis".)

This is a reasonable model for stations that can observe the channel all the time. However, in some cases (e.g. the internet) a station can only know the success or failure of its own packets.

5.2. **Acknowledgement-based schemes.** Suppose that a station knows only the history of its own transmission attempts. The Ethernet model is as follows: after r unsuccessful attempts to transmit, a station will wait for a time B_r that is distributed uniformly on $\{1, 2, \ldots, 2^r\}$. (This is known as the binary exponential back-off.)

Remark. We are still assuming that each packet corresponds to a unique station – i.e., each packet has its own transmission history and back-off time (as is actually implemented in the Ethernet protocol).

The assumption that back-off is *exponential* is essential – it can be shown that if the probability of a retransmission attempt during the next time slot falls off less quickly than an exponential, then we cannot have a positive throughput. The assumption that the back-off is *binary* exponential, i.e. that we have 2^r rather than a^r for some other a, is not at all essential: historically we get 2^r because that's easier to implement in a binary register.

It can be shown that for $\nu < \log 2 \approx 0.693$ there will be infinitely many successful transmissions (with probability 1).

This does not imply the recurrence of the system! Suppose that there were a positive recurrent Markov chain describing the system, and let π_i be its equilibrium probability of retransmitting i packets in slot $(t, t+1)$ (where $i = 0, 1$). Equating arrival and departure rates we get

$$\nu = \pi_1 e^{-\nu} + \pi_0 \nu e^{-\nu}$$

(in this model we again distinguish between new arrivals, which attempt transmission immediately, and retransmission attempts). Since clearly $\nu \leq 1$ and also $\pi_0 + \pi_1 = 1$, the right-hand side is bounded by $e^{-\nu}$. This implies $\nu \leq e^{-\nu}$, which means $\nu < 0.567$.

That is, for $0.567 < \nu < 0.693$, the system cannot be described by a recurrent Markov chain, but will nevertheless have infinitely many successful transmissions. In fact, the existence of a positive recurrent scheme having any positive throughput $\nu > 0$ is still an open problem!

6. Effective bandwidth

Until now, our model of communication assumed that individual requests request some amount of bandwidth and use all or none of it. (We then used this model to develop the Erlang formula and our analysis of loss networks.) However, it is possible that the bandwidth profile of a request is different.

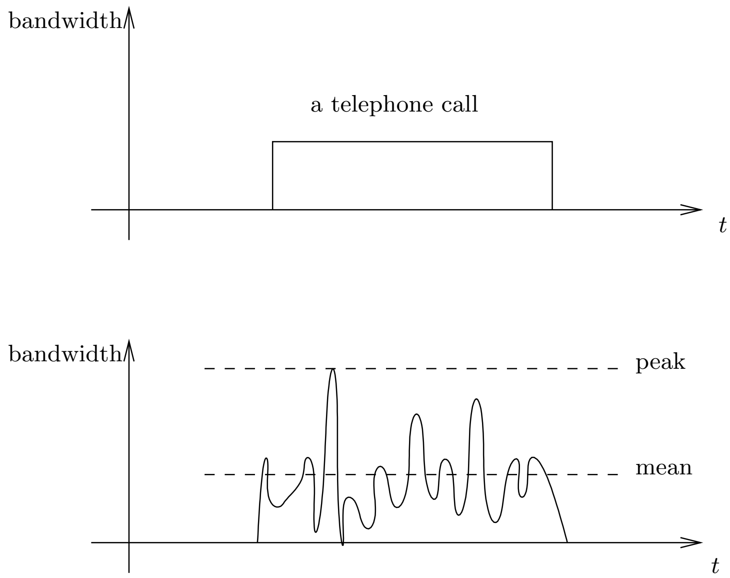


FIGURE 36. Possible user bandwidth profiles; mean and peak

If we are feeding several such users through a system of queues, how should we estimate the bandwidth allocation per user? (I.e., how should we do the admission control of letting or not letting another user into the network?)

- We could use the previous model of loss networks, and allocate to each user her peak rate. This would be stable, but rather extravagant.
- We could use the mean rate instead, saying that another call is allowed into the network if and only if with it the mean rate at which calls arrive to each of the queues in the network will be < 1. This may still be stable (although in a network, this is quite a tricky statement to prove!), but even if it is, it will result in large queues (and with finite buffers, it will result in packets being lost).

We would like to find some sensible description between these two extremes.

Let $X_1, \ldots, X_n$ be iid random variables, $\mathbb{E}[X_1] < 0$ and $\mathbb{P}(X_1 > 0) > 0$. Let $M(s) = \log \mathbb{E} e^{sX_1}$.

Now, for any random variable Y and any $s \geq 0$ we have the *Chernoff bound*

$$\mathbb{P}(Y \geq 0) = \mathbb{E}[I[Y \geq 0]] \leq \mathbb{E}[e^{sY}]$$

This gives us the bound

$$\log \mathbb{P}(Y \geq 0) \leq \inf_{s \geq 0} \log \mathbb{E}[e^{sY}] \tag{19}$$

Applying this to the sequence X_i, we get

$$\frac{1}{n} \log \mathbb{P}(X_1 + \ldots + X_n \geq 0) \leq \inf_{s \geq 0} M(s)$$

since the X_i are iid, and $M_{X_1+\ldots+X_n}(s) = M_{X_1}(s) + \ldots + M_{X_n}(s) = nM_{X_1}(s)$.

We next want to show that this bound is asymptotically exact as $n \to \infty$.

Theorem 6.1 (Cramér's theorem)**.**

$$\lim_{n \to \infty} \frac{1}{n} \log \mathbb{P}(X_1 + \ldots + X_n \geq 0) = \inf_{s \geq 0} M(s).$$

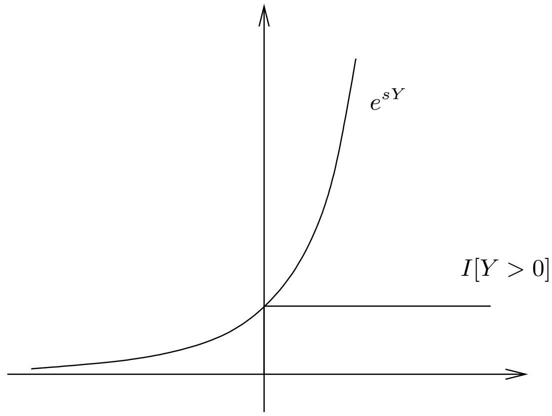


FIGURE 37. $e^{sY} \geq I[Y > 0]$

Remark. In lecture, this theorem was stated as

$$\lim_{n\to\infty} \frac{1}{n} \log \mathbb{P}\left(\frac{1}{n}(X_1 + \ldots + X_n) \geq 0\right) = \inf_{s\geq 0} M(s).$$

The two probabilities are, of course, the same: the average of $X_1, \ldots, X_n$ is positive if and only if the sum is. However, the average is in many ways a more sensible object to look at, and when we prove the lower bound we will be looking at the average.

Sketch of proof. We showed $\leq$ already; let's show the lower bound. We rewrite the event we're interested in as

$$\frac{1}{n}(X_1 + \ldots + X_n) \geq 0,$$

and we will show that for any $\epsilon > 0$,

$$\lim_{n\to\infty} \frac{1}{n} \log \mathbb{P}\left(-\epsilon < \frac{1}{n}(X_1 + \ldots + X_n) < \epsilon\right) \geq \inf_{s\geq 0} M(s).$$

(In particular, together with the upper bound, this tells us that if $\frac{1}{n}(X_1+\ldots+X_n) \geq 0$ then in fact it is quite near 0.)

Let s^* be the value that achieves the infimum. Then we have $M'(s^*) = 0$. Now,

$$M'(s^*) = \frac{d}{ds} \log \mathbb{E}[e^{sX_1}]|_{s=s^*} = \frac{\mathbb{E}[X_1 e^{s^* X_1}]}{\mathbb{E}[e^{s^* X_1}]} = \mathbb{E}[X_1 e^{s^* X_1 - M(s^*)}]$$

Let's make that into an expectation of another variable, $\tilde{X}_1$, which has the probability density function

$$f_{\tilde{X}_1}(x) = e^{-M(s^*)} e^{s^* x} f_{X_1}(x).$$

We check that this is a density function:

$$\int f_{\tilde{X}_1}(x) dx = e^{-M(s^*)} \int e^{s^* x} f_{X_1}(x) = e^{-M(s^*)} \mathbb{E}[e^{s^* X_1}] = 1.$$

We check that $\tilde{X}_1$ has mean 0:

$$\mathbb{E}[\tilde{X}_1] = \int x f_{\tilde{X}_1}(x) dx = \int x e^{-M(s^*)} e^{s^* x} f_{X_1}(x) dx = \mathbb{E}[X_1 e^{s^* X_1 - M(s^*)}] = M'(s^*) = 0.$$

Now,

$$\begin{aligned}
\mathbb{P}(-\epsilon < \frac{1}{n}(X_1 + \ldots + X_n) < \epsilon) &= \int_{-n\epsilon < x_1 + \ldots + x_n < n\epsilon} \cdots \int f_{X_1}(x_1)dx_1 \ldots f_{X_n}(x_n)dx_n \\
&= \int_{-n\epsilon < x_1 + \ldots + x_n < n\epsilon} \cdots \int e^{M(s^*)}e^{-s^*x_1} f_{\tilde{X}_1}(x_1)dx_1 \ldots e^{M(s^*)}e^{-s^*x_n} f_{\tilde{X}_n}(x_n)dx_n \\
&= \int_{-n\epsilon < x_1 + \ldots + x_n < n\epsilon} \cdots \int e^{nM(s^*)}e^{-s^*(x_1+\ldots+x_n)} f_{\tilde{X}_1}(x_1)dx_1 \ldots f_{\tilde{X}_n}(x_n)dx_n \\
&\geq e^{n(M(s^*) - \epsilon s^*)} \int \cdots \int I[-n\epsilon < x_1 + \ldots + x_n < n\epsilon] f_{\tilde{X}_1}(x_1)dx_1 \ldots f_{\tilde{X}_n}(x_n)dx_n \\
&= e^{n(M(s^*) - \epsilon s^*)} \mathbb{P}\left(-\epsilon < \frac{1}{n}(\tilde{X}_1 + \ldots + \tilde{X}_n) < \epsilon\right)
\end{aligned}$$

Since $\tilde{X}_1, \ldots, \tilde{X}_n$ are iid of mean 0, the last probability tends to 1 by the strong law of large numbers. Therefore,

$$\lim_{n\to\infty} \frac{1}{n} \log \mathbb{P}\left(-\epsilon < \frac{1}{n}(X_1 + \ldots + X_n) < \epsilon\right) \geq M(s^*) - \epsilon s^*.$$

Since this holds for any ϵ, we get that the probability of $\frac{1}{n}(X_1 + \ldots + X_n) \approx 0$ goes as $e^{nM(s^*)} = \exp(n \inf_{s\geq 0} M(s))$, as required. □

Remark.

- We glossed over some technicalities:
  - Why is the cumulant generating function $M(s)$ defined. This is a fundamental assumption – we need $M(s)$ to be finite on some neighbourhood of 0, otherwise large deviations results for heavy-tailed random variables are nowhere near as simple.
  - Why is the infimum $M(s^*)$ attained. The function $M(s)$, when finite, is convex and differentiable (in the interior of the set where it is finite); we do need to assume that s^* is in the interior of this domain. One possibility is for $M(s) < \infty$ for all s while $\mathbb{P}(X_1 > 0) > 0$, because as $s \to \infty$ we have $M(s)/s = \frac{1}{s}\log \mathbb{E}[e^{sX_1}] \to \operatorname{ess\,sup} X_1$ (the essential supremum, i.e. $\sup\{x : \mathbb{P}(X_1 > x) > 0\}$). Thus, provided X_1 is ever positive, $M(s) \to \infty$ as $s \to \infty$.
  - The change of measure argument as we've stated it requires X_1 to have a probability density function, so doesn't admit discrete-valued random variables. This is a matter of notation – we can use the law of X_1 to generate the law of $\tilde{X}_1$, and it will all go through for discrete variables as well.
  - Actually, we haven't quite given a lower bound on the probability that $X_1 + \ldots + X_n \geq 0$. What we have given is (easily converted into) a lower bound on the probability that $\frac{1}{n}(X_1 + \ldots + X_n)$ belongs to some open set (the open sets we have considered are $(-\epsilon, \epsilon)$). This gives the required form for $\mathbb{P}(X_1 + \ldots + X_n > 0)$, and since the upper bound holds for ≥ 0, we are done.
- It's easy to see that

$$\liminf_{n\to\infty} \frac{1}{n} \log \mathbb{P}(X_1 + \ldots + X_n > 0) > -\infty,$$

provided $\mathbb{P}(X_1 > 0) = p > 0$. Indeed, to have $X_1 + \ldots + X_n > 0$ it is sufficient to have each $X_i > 0$, so this event has probability at least p^n, giving

$$\frac{1}{n}\log \mathbb{P}(X_1 + \ldots + X_n > 0) \geq \log p.$$

We expect to be able to do a bit better than that, since we shouldn't need every single X_i to be positive.

- What we in fact showed is that the most likely way to have $\frac{1}{n}(X_1 + \ldots + X_n) \geq 0$ is for the variables to behave as though they were iid but with an *exponentially tilted* distribution that has mean 0, i.e. $\tilde{X}_1, \ldots, \tilde{X}_n$.

6.1. **Effective bandwidth.** Consider a single link of capacity C, and J types of users sharing it. There are n_j requests of type $j = 1, \ldots, J$. Let

$$S = \sum_{j=1}^{J}\sum_{i=1}^{n_j} X_{ji},$$

where X_{ji} are independent random variables, and for a fixed $j = j^*$ the X_{j^*i} are iid. Let

$$M_j(s) = \log \mathbb{E}[e^{sX_{ji}}] \text{ for any } i.$$

X_{ji} is the load of a source of class j.

Suppose that $\mathbb{E}S < C$, but $\mathbb{P}(S > C) > 0$. How can we control this probability? By Chernoff's bound (19),

$$\log \mathbb{P}(S > C) \leq \log \mathbb{E}[e^{s(S-C)}] = \sum_{j=1}^{J} n_j M_j(s) - sC.$$

(Cramér's theorem tells us that if we take $\inf_{s\geq 0}$ on the right-hand side, and let $n_j \to \infty$ on the left-hand side, we will get asymptotic equality.)

In particular,

$$\inf_{s\geq 0}\left(\sum_{j=1}^{J} n_j M_j(s) - sC\right) \leq -\gamma \implies \mathbb{P}(S \geq C) \leq e^{-\gamma}. \tag{20}$$

This is useful for deciding whether we can add another call of class k and still retain the service quality guarantee. Let

$$A = \{\mathbf{n} : \sum_{j=1}^{J} n_j M_j(s) - sC \leq -\gamma \text{ for some } s \geq 0\}.$$

A is the acceptance region, and (20) shows that $\mathbf{n} \in A \implies \mathbb{P}(S > C) \leq e^{-\gamma}$.

The complement of A is convex in $\mathbb{R}^J_+$, since A^c is the intersection of half-spaces (one for each s).

Suppose that the system is operating at close to capacity, i.e. at some $\mathbf{n}^*$ near the boundary of A. How can we tell whether we can accept another call of class k? That is, what does the boundary of A near $\mathbf{n}^*$ look like?

The tangent hyperplane at $\mathbf{n}^*$ is given by

$$\sum_{j=1}^{J} n_j M_j(s^*) - s^*C = -\gamma,$$

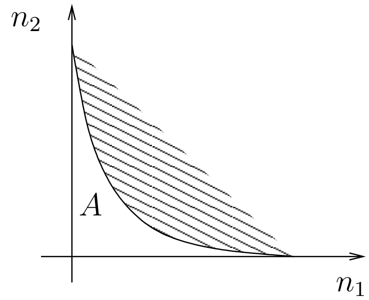


FIGURE 38. The convex complement of the acceptance region.

where s^* attains the infimium in (20) for $\mathbf{n} = \mathbf{n}^*$. The half-space in which we are interested is

$$\sum_{j=1}^{J} n_j \frac{M_j(s^*)}{s^*} \leq C - \frac{\gamma}{s^*}. \tag{21}$$

If this inequality holds, then we are guaranteed to have $\mathbb{P}(S > C) \leq e^{-\gamma}$.

Let $\alpha_j(s) = \frac{M_j(s)}{s} = \frac{1}{s}\log \mathbb{E}[e^{sX_{ji}}]$. We call $\alpha_j(s)$ the *effective bandwidth* of a source of class j, and we call $C - \frac{\gamma}{s}$ the *effective capacity* of the link.

We have found a version of admissions control: compute the total effective bandwidth used by the calls already in progress plus the new call that wants to come onto the network, and check whether it is still below the effective capacity. (If it is, we admit the new call; if not, we block it.) If we were in a system with many links, then call types would be indexed by r (previously, route, but now also the bandwidth requirement). On each link j we would have a value s_j, and the effective capacity of a call of type r would be $\alpha_r(s_j)$; this replaces A_{jr}, the number of circuits that a call of type r used on link j. The effective capacity of the link would be $C_j - \frac{\gamma}{s_j}$. A call is now admitted or not according to whether adding it would make the effective bandwidth requirement exceed effective capacity on any of the links.

Our admission control is conservative, and for two reasons. Firstly, the set A is conservative: there are vectors $\mathbf{n} \in A^c$ for which $\mathbb{P}(S > C) \leq e^{-\gamma}$. On the other hand, the lower bound in Cramér's theorem tells us that there is an asymptotic regime in which A is the correct set to use. A more serious issue is that we are replacing A with something bounded by a hyperplane, and thus losing portions of the acceptance region. Thus, it is interesting to see what the shape of the acceptance region is and how much we would lose.

Remark. How do we pick s? It's difficult to do a priori. On the other hand, in a live network, we can estimate the operating point $\mathbf{n}^*$ from the relative numbers of calls of different types, and then let $s = s^*(\mathbf{n}^*)$ be the value achieving the infimum.

Remark. Where does effective bandwidth lie with respect to the mean and the peak? By Jensen's inequality,

$$\frac{1}{s}\log \mathbb{E}[e^{sX}] \geq \frac{1}{s}\log e^{s\mathbb{E}X} = \mathbb{E}X;$$

on the other hand,

$$\frac{1}{s}\log \mathbb{E}[e^{sX}] \leq \frac{1}{s}\log e^{s \cdot \operatorname{ess\,sup} X} = \operatorname{ess\,sup} X,$$

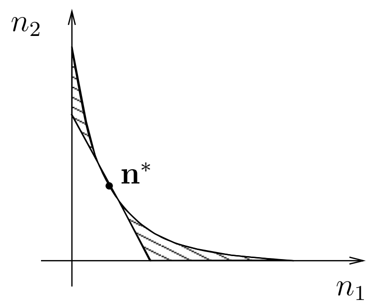


FIGURE 39. Approximating the acceptance region by a half-space bounded by the tangent at $\mathbf{n}^*$. The shaded regions are in A but will not be admitted into the loss-like network.

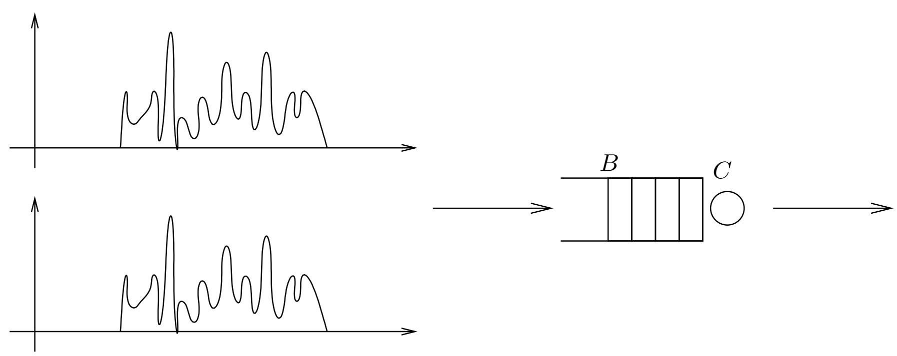


FIGURE 40. Many users feeding through a buffer

where ess sup (essential supremum) is defined as $\sup\{x : \mathbb{P}(X > x) > 0\}$ (this is what we would meaningfully call the peak bandwidth).

As $s \to 0$, we get the lower bound, i.e. the mean; as $s \to \infty$, we get the upper bound, i.e. the peak; in between effective bandwidth is an increasing function of s (observe that effective capacity also increases in s).

6.2. **Queue with buffering.** A more realistic model of a network will include a buffer: that is, we can store some finite amount of overflow. For a general queueing system it is hard to get any results at all. However, it is useful to have limit theorems, which give simple conclusions.

Let $X_{ji}[t_1, t_2]$ be the workload arriving from ith source of type j during the interval $[t_1, t_2]$. Assume that $(X_{ji})_{i,j}$ are independent, and that each process has *stationary increments*: that is, $X_{ji}[t_1, t_2]$ depends only on $t_2 - t_1$.

Example 6.1.
- $X_{ji}[t_1, t_2] = (t_2 - t_1)X_{ji}$, where X_{ji} is some random variable. (Note we have not assumed ergodicity, or independence of the increments across time.) This essentially reproduces the previous model: buffer overflow would occur if and only if $\sum X_{ji} > C$ (and then buffer size is irrelevant).
- $M/G/1$ queue: $X_{ji}[0, t] = \sum_{i=1}^{N(t)} \gamma_i$. Here, the γ_i are iid random variables, and $N(t)$ is the Poisson process of arrivals (independent of the service times).

Suppose that the queue starts out empty at time $-\infty$ (that is, there is a time far enough back into the past when the queue was empty). The stationary queue size is

$$Q(0) = \sup_{0 \le \tau < \infty} X[-\tau, 0] - C\tau \quad \text{where } X[t_1, t_2] = \sum_{j=1}^{J} \sum_{i=1}^{n_j} X_{ji}[t_1, t_2].$$

(The stationary queue size is time-homogeneous, so we could have chosen an arbitrary time T instead of 0; but we have enough t-like variables floating around.)

Remark. Clearly, the queue at time 0 should be at least $X[-\tau, 0] - C\tau$ for any τ, because it cannot serve work any faster than that. On the other hand, if $\tau < \infty$ achieves the supremum, then we have $X[-\tau, -t] \ge C(\tau - t)$ for all $0 \le t \le \tau$, so the queue cannot possibly idle during $(-\tau, 0)$, so it must be draining at the full rate C the entire time. Of course, there is a valid question of why the queue was empty at time $-\tau$...

Let $L(C, B, \mathbf{n}) = \mathbb{P}(Q(0) > B)$ when service rate is C and $\mathbf{n} = (n_1, \ldots, n_J)$. The event $Q(0) > B$ corresponds to buffer overflow, and we would like to control its probability. (Actually, we aren't quite modelling buffer overflow, because traffic that exceeds buffer size isn't *lost*; but perhaps it is tucked away into expensive storage. Think of it as the system being in pain.)

Theorem 6.2 (Botvich & Duffield, Courcoubetis & Weber, Simonian & Guibert)**.**

$$\lim_{N\to\infty} \frac{1}{N} \log L(CN, BN, \mathbf{n}N) = \sup_{t\ge 0} \inf_{s \ge 0} st \sum_{j \in J} n_j(\alpha_j(s,t) - s(B + Ct))$$

where

$$\alpha_j(s,t) = \frac{1}{st} \log \mathbb{E} e^{sX_{ji}[0,t]}$$

is the effective bandwidth.

Sketch of proof.

$$\mathbb{P}(Q_0^N > BN) = \mathbb{P}(X^N[-t, 0] > (B + Ct)N \text{ for some } t)$$

This is bounded from below by $\mathbb{P}(X^N[-t, 0] > (B + Ct)N)$ for any fixed t, and from above by $\sum_{t \ge 0} \mathbb{P}(X^N[-t, 0] > (B + Ct)N)$.

Now, for any fixed t we have the Chernoff bound

$$\frac{1}{N} \log \mathbb{P}\left(\sum_{j=1}^{J} \sum_{i=1}^{n_j N} X_{ji}[-t, 0] > (B + Ct)N \right) \to \inf_s st \sum_{j=1}^{J} n_j \alpha_j(s,t) - s(B + Ct).$$

This gives us the lower bound directly. On the other hand, the terms decay exponentially in N, and we can ignore all but the largest exponent in the sum, so in fact the upper bound is the same. □

Remark. What is the meaning of s^* and t^* that achieve the infimum?

t^* is relatively easy: it is telling us that if buffer overflow occurs, then the most likely way for this to happen is over a time scale t^*. That is, if I record the queue size and look at it from far enough away (i.e. for large enough N), I will see something like Figure 41.

s^* is a bit less intuitive. What we get from t^* is that over the time period $[-t^*, 0]$ the arrival process X has a mean arrival rate of $C + \frac{B}{t^*}$ rather than S. Now, from Cramér's

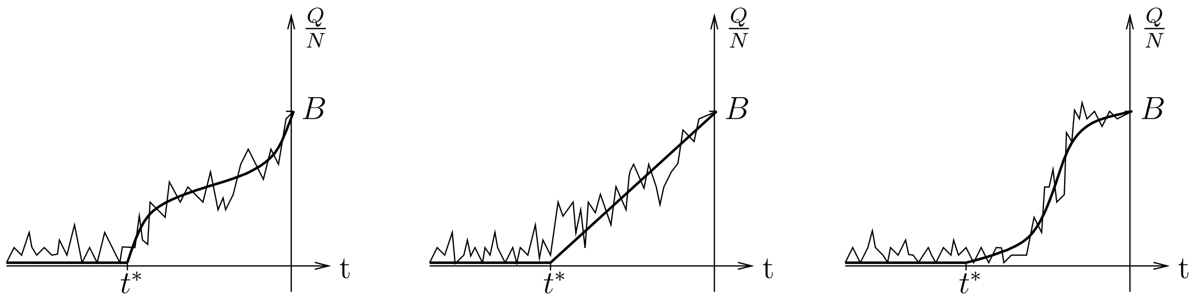


FIGURE 41. Most likely paths to buffer overflow. The precise path depends on the long-range dependence features of X_{ji}. The thick line is the limit as $N \to \infty$, the thin line is a potential instance for finite N.

theorem we know that the most likely way for that to happen is to tilt the distribution of X exponentially until it has mean $C + \frac{B}{t^*}$. The fact that I have a single parameter s^*, rather than s_j^* for each j, tells me that in fact *all* X_{ji} have a bigger mean arrival rate than usual. That is, the most likely reason for buffer overflow is (e.g.) to have twice the usual number of calls but with the usual ratio of voice to video, rather than a sudden surge of interest in video calling relative to voice calls. (That is, s^* determines the most likely tilt for the distribution of X – and all of X_{ji} – given that the average over a period of length t^* must be at least $C + \frac{B}{t^*}$.)

7. Internet congestion control

In the previous section we looked at organising a network of users with time-varying bandwidth requirements into something that resembles a loss network, by finding a notion of effective bandwidth and effective capacity. However, the Internet paradigm has been not to have any sense of "calls" – instead, all that travels on the network are packets with addresses. The users then treat the network as something stupid (in particular, incapable of admission control), and any control is done by the end users themselves.

7.1. **Control of elastic network flows.** In our model, users will be more or less identified with flows. Each flow passes through some network resources. The questions we would like to answer are: how should available resources be shared between competing streams of elastic traffic? We would like the allocation to be fair to the users, and we would also like it to stabilize the network.

Elastic traffic means that users prefer to share; it corresponds to concave utility functions. For example, if I am downloading a web page, I would prefer a moderate speed rather than a toss-up between very high and very low speeds. (If I am in a voice call, there may well be some bandwidth below which the voice will be undecipherable, so I would prefer all or nothing.)

The basic model we consider is as follows. Let J be a set of *resources*, let R be a set of *routes*. A route r is a subset $r \subset J$.

Let $A_{jr} = 1$ if $j \in r$, and $A_{jr} = 0$ otherwise. (You might want to think about how, and whether, the model generalises to A_{jr} with different values.)

For $j \in J$ let C_j be the *capacity* of resource j.

Routes correspond to users. If x_r is the rate on route r, then the utility to user r is $U_r(x_r)$. We are assuming that the traffic is elastic, i.e. $U_r(\cdot)$ is increasing and concave. In fact, for

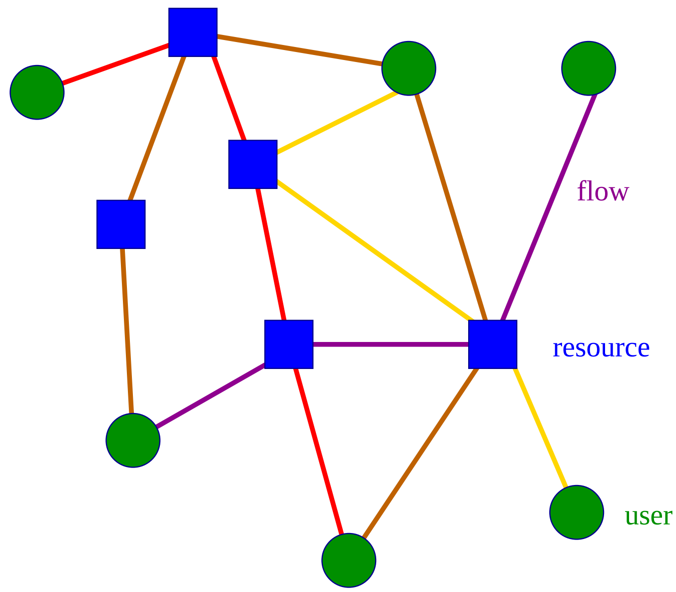


FIGURE 42. A model network with end users, resources, and flows.

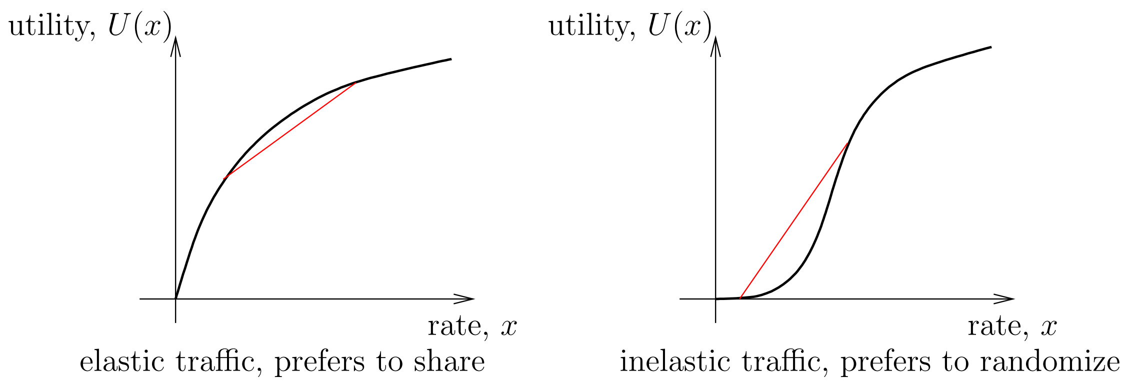


FIGURE 43. Elastic and inelastic demand

convenience we will assume $U_r(\cdot)$ is strictly concave, continuously differentiable on $(0, \infty)$, and satisfies $U_r'(0) = \infty$ and $U_r'(\infty) = 0$.

Let C be the vector of capacities $C = (C_j, j \in J)$, and let x be the vector of flow rates $x = (x_r, r \in R)$. The feasible region is given by

$$Ax \leq C.$$

The SYSTEM(U, A, C) problem is

$$\begin{aligned} &\max \sum_{r \in R} U_r(x_r) \\ &\text{s.t. } Ax \leq C \\ &\text{over } x \geq 0. \end{aligned} \tag{22}$$

That is, we want to maximize the aggregate utility subject to capacity constraints. The difficulty with this model is firstly that the system does not know U_r, the individual users' utilities; and secondly, that nobody in the system might know the full matrix A (e.g., on the internet, nobody knows the full routing table). We'll deal with these in order.

We introduce two more optimization problems.

The $\text{USER}_r(U_r;\lambda_r)$ problem is

$$\max U_r\left(\frac{w_r}{\lambda_r}\right) - w_r \quad \text{over } w_r \geq 0. \tag{23}$$

This corresponds to user r choosing an amount w_r that he will pay per unit time. He will then receive a flow proportional to w_r, namely $x_r = w_r/\lambda_r$. (No capacity constraints come into this.)

The $\text{NETWORK}(A, C; w)$ problem is

$$\begin{aligned} &\max \sum_{r\in R} w_r \log x_r \\ &\text{s.t. } Ax \leq C \\ &\text{over } x \geq 0. \end{aligned} \tag{24}$$

This looks like the system problem, but with particular (logarithmic) utility functions. The constants w_r are chosen by the users.

Remark. We could choose some other family of utility functions, rather than $w_r \log(\cdot)$, and still get a decomposition result. The logarithmic utility function, as we will see, corresponds to a notion of fairness.

Theorem 7.1 (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*

(1) *$w_r = \lambda_r x_r$ for $r \in R$*
(2) *w_r solves $USER_r(U_r;\lambda_r)$ for all $r \in R$*
(3) *x solves $NETWORK(A, C; w)$.*

Moreover, the vector x then also solves $SYSTEM(U, A, C)$.

That is, the system problem can be solved by simultaneously solving the network and user problems. (This suggests that in practice we might try to solve one of them, then the other, iteratively, hoping that the result will converge.)

Remark. Another decomposition of the system problem comes from writing down the Lagrangian. However, this particular decomposition is operationally nicer.

Proof. The Lagrangian for SYSTEM is

$$\mathcal{L}_{\text{SYSTEM}}(x, z; \mu) = \sum_{r\in R} U_r(x_r) + \mu^T(C - Ax - z)$$

(here, $z = (z_j, j \in J)$ are the slack variables and $\mu = (\mu_j, j \in J)$ the Lagrange multipliers). Optimizing over x_r,

$$\frac{\partial}{\partial x_r}\mathcal{L}_{\text{SYSTEM}}(x, z; \mu) = U_r'(x_r) - \sum_{j\in r} \mu_j$$

since the term with $\mu^T A$ will leave only those μ_j with $j \in r$.

The Lagrangian for NETWORK is

$$\mathcal{L}_{\text{NETWORK}}(x, z; \mu) = \sum_{r \in R} w_r \log x_r + \mu^T (C - Ax - z)$$

and

$$\frac{\partial}{\partial x_r} \mathcal{L}_{\text{NETWORK}}(x, z; \mu) = \frac{w_r}{x_r} - \sum_{j \in r} \mu_j$$

(Note that a priori we have no reason to suppose that the optimum for NETWORK and SYSTEM will involve the same dual variables μ_j, but we will see that when USER problem is satisfied, this is in fact so.)

Finally, for the USER the objective is $U_r\left(\frac{w_r}{\lambda_r}\right) - w_r$, and

$$\frac{\partial}{\partial w_r}\left[U_r\left(\frac{w_r}{\lambda_r}\right) - w_r\right] = \frac{1}{\lambda_r} U_r'\left(\frac{w_r}{\lambda_r}\right) - 1.$$

Now, if user r chooses w_r so that the USER objective is optimized, i.e. so that $U_r'\left(\frac{w_r}{\lambda_r}\right) = \lambda_r$, then the previous derivatives become identical under the identification

$$x_r = \frac{w_r}{\lambda_r}, \quad \lambda_r = \sum_{j \in r} \mu_j.$$

□

If μ_j is the *shadow price* per unit flow through resource j, then λ_r is the shadow price along route r.

The value of the NETWORK Lagrangian is

$$\max_{x,z \geq 0} \mathcal{L}_{\text{NETWORK}}(x, z; \mu) = \sum_{r \in R} w_r \log \frac{w_r}{\sum_{j \in r} \mu_j} + \mu^T C,$$

so (subtracting the constant $\sum_r w_r \log w_r$ and changing sign) the dual problem to NETWORK$(A, C; w)$ becomes

$$\begin{aligned} &\max \sum_{r \in R} w_r \log\left(\sum_{j \in r} \mu_j\right) - \sum_{j \in J} \mu_j C_j \\ &\text{over } \mu \geq 0. \end{aligned} \tag{25}$$

This proof suggests that we can introduce appropriate charging mechanisms to solve the SYSTEM problem without knowing the user utilities. However, in the early days of the internet it was designed to be free, and it was desirable not to phrase the mechanisms in terms of any payment system. We will now move on to discuss how this might be achieved.

7.2. **A few notions of fairness.** We would like the allocation of flow rates to the users to be fair in some sense, but what do we mean by this?

Max-min fairness. We say that $x = (x_r, r \in R)$ is *max-min fair* if it is feasible, and for any other feasible vector y,

$$\exists r : y_r > x_r \implies \exists s : y_s < x_s \leq x_r.$$

That is, for r to benefit, someone else (s) who was worse off than r needs to get hurt.

The compactness and convexity of the feasible region imply that such a vector x exists and is unique.

As a concept, max-min fairness has been discussed by political philosophers (cf. J. Rawls, *A Theory of Justice*, 1971.) However, for our purposes it seems a bit too restrictive, because it places a very high emphasis on maximizing the lowest rate.

Proportional fairness. We say $x = (x_r, r \in R)$ is *proportionally fair* if it is feasible, and for any other feasible vector y, the aggregate of proportional changes is nonpositive:

$$\sum_r \frac{y_r - x_r}{x_r} \leq 0.$$

This still places a higher value on increasing small flows (which have a smaller denominator), but it isn't quite as overwhelmingly high.

Let $w = (w_r, r \in R)$ be a vector of weights: then $x = (x_r, r \in R)$ is *weighted proportionally fair* if it is feasible, and if for any other feasible vector y,

$$\sum_r w_r \frac{y_r - x_r}{x_r} \leq 0.$$

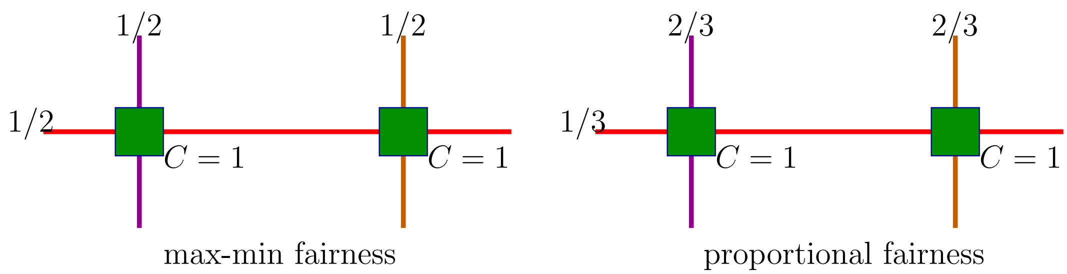


Figure 44. The difference between max-min and proportional fairness. Note that the throughput-optimizing allocation is (1,1,0).

Proposition 7.2. *A vector x solves NETWORK$(A, C; w)$ if and only if it is weighted proportionally fair.*

Proof. Consider a perturbation of x, $y = x + \delta x$ (so $y_r = x_r + \delta x_r$). The objective function $\sum_{r \in R} w_r \log x_r$ of NETWORK will change by an amount

$$\sum_{r \in R} w_r \frac{\delta x_r}{x_r} + o(\delta x).$$

By convexity of the feasible region and strict concavity of the objective function, we are at an optimum if and only if this derivative is zero; but this is precisely the condition for proportional fairness with respect to a small change. □

There are alternative interpretations of the NETWORK, or proportionally fair, allocation.

Bargaining problem (Nash, 1950). The solution to NETWORK$(A, C; w)$ with $w_r = 1$ for all $r \in R$ is the unique point satisfying the following axioms:

- Pareto efficiency. A vector is Pareto inefficient if there exists an alternative allocation that improves the utility of at least one player without reducing the utility of any other players.
- Symmetry. If we relabel the players, the optimum should simply be relabelled accordingly.

- Independence of Irrelevant Alternatives. Deleting an alternative shouldn't change the preferences among the remaining ones. (This one is more controversial than the rest of them.)

For general w, we model a situation with unequal bargaining power (so the notion of symmetry gets modified accordingly).

Market clearing equilibrium (Gale, 1960). We want to find prices $(p_j, j \in J)$ and an allocation $(x_r, r \in R)$ such that

- $p \geq 0$, $Ax \leq C$ (feasibility)
- $p^T(C - Ax) = 0$ (complementary slackness): if the price of a resource is positive then the resource is used up
- $w_r = x_r \sum_{j \in r} p_j, r \in R$: if user r has an endowment w_r, then all endowments are spent.

The claim is that a market clearing equilibrium exists, and solves NETWORK$(A, C; w)$.

7.3. **A primal algorithm.** We will now examine a distributed protocol that would behave similarly to the above model.

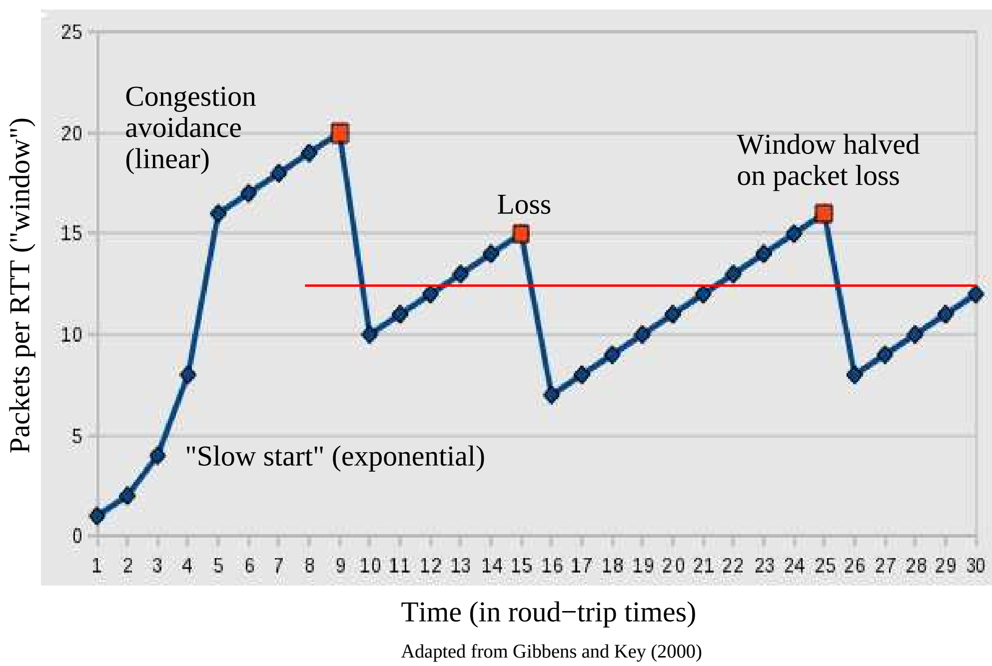


FIGURE 45. Typical window size for TCP

The TCP ("Transmission Control Protocol") was invented by Jacobson. The idea is that there is a window (of size `cwnd`) of packets that have been sent but not yet acknowledged. When the receiver gets a packet, it sends back an acknowledgement, and receiving an acknowledgement causes the window size to be increased, thus increasing the average transmission rate. On the other hand, when a packet gets lost (presumably, due to congestion and buffer overflow somewhere inside the network), the window size is halved. (The dropping of a packet is detected either because of a time-out or because the acknowledgements arrive out-of-order.)

Our network model is as before; in particular, x_r is the flow rate on route r. We assume now that this flow rate can be time-varying, and hope that the system will on average achieve the correct flow rate. (In the diagram, the horizontal red line approximates the steady-state flow rate.)

We describe a differential equation, a *primal algorithm*, that will behave similarly to the above set-up:

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

$$\mu_j(t) = p_j \underbrace{\left(\sum_{s:j \in s} x_s(t) \right)}_{\text{total flow through link } j}$$

Where $p_j(\cdot)$ is a non-negative, continuous, increasing function (some measure of "pain", perhaps drop probability). The interpretation of this is that resource j generates feedback signals at rate $\mu_j(t)$; these signals reach each user r whose route passes through resource j. The *decrease* in flow x_r is multiplicative, at a rate proportional to the stream of feedback signals received. The *increase* in flow x_r is linear, at a rate proportional to w_r (the weight).

Theorem 7.3 (Global stability)**.**

$$\mathcal{U}(x) = \sum_{r \in R} w_r \log x_r - \sum_{j \in J} \int_0^{\sum_{s:j \in s} x_s} p_j(y) dy$$

is a Lyapunov function for the primal algorithm. The unique value maximizing $\mathcal{U}(x)$ is a stable point of the system, to which all trajectories converge.

Definition. For us, a *Lyapunov function* is a real-valued function of the system state that is monotone (either increasing or decreasing) along each trajectory of the system.

Proof. Note that $\mathcal{U}(x)$ has a unique maximum which is attained. Indeed, $\mathcal{U}(\cdot)$ is strictly concave; $\mathcal{U}(x) \to -\infty$ as $x_r \to 0$ from the first term; and the conditions on $p_j(\cdot)$ ensure that $\mathcal{U}(x) \to -\infty$ as $x_r \to \infty$, for each $r \in R$. Moreover, $\mathcal{U}(x)$ is continuously differentiable, so the unique maximum occurs at the point where $\frac{\partial}{\partial x_r}\mathcal{U}(x) = 0$ for each $r \in R$.

Now,

$$\frac{\partial}{\partial x_r}\mathcal{U}(x) = \frac{w_r}{x_r} - \sum_{j \in r} p_j \left(\sum_{s:j \in s} x_s \right),$$

and so

$$\frac{d}{dt}\mathcal{U}(x(t)) = \sum_{r \in R} \frac{\partial U}{\partial x_r} \frac{d}{dt} x_r(t) = \sum_{r \in R} \frac{\kappa_r}{x_r(t)} \left(w_r - x_r(t) \sum_{j \in r} p_j \left(\sum_{s:j \in s} x_s \right) \right)^2 \geq 0.$$

This almost, but not quite, proves the desired convergence to equilibrium: we need to make sure that the derivative of $\mathcal{U}(x(t))$ isn't so small that the system never manages to reach equilibrium. But, indeed, $\mathcal{U}$ has a unique maximum and the gradient $(\frac{\partial U(x)}{\partial x_r})_r$ is continuous, so it is bounded away from 0 outside of any open neighbourhood of the optimal point

x. Therefore, $\frac{d}{dt}\mathcal{U}(x(t))$ is bounded away from 0 outside of any neighbourhood of the optimum, which means that the system can spend only a finite amount of time outside this neighbourhood (and thus converges). □

Note that

$$C_j(\sum_{s:j\in s} x_s) = \int_0^{\sum_{s:j\in s} x_s} p_j(y)dy$$

is a cost function penalizing proximity to the capacity constraint for resource j. In particular, if we chose

$$p_j(y) = \begin{cases} \infty, & y > C_j \\ 0, & y \leq C_j \end{cases}$$

then maximizing the Lyapunov function $\mathcal{U}(x)$ becomes the primal problem NETWORK$(A, C; w)$. (Of course, this specific p_j violates our assumptions, but we could approximate it with something that doesn't.) However, we shall see later that, in the presence of propagation delays in the network, large values of $p_j'(\cdot)$ compromise stability.

Remark. The algorithm we considered above is decentralised: nobody in the network needs to know all of A, it is simply the case that route r needs to be able to get feedback from the links $j \in r$, and link j needs to know about routes r passing through j (which seems realistic even in a large network).

7.4. **MulTCP (congestion avoidance).** Jacobson's TCP maintains a window of transmitted but not yet acknowledged packets; the rate x and the window size `cwnd` satisfy the approximate relation `cwnd` $= xT$, where T is the round-trip time for the packet to get from the sender to the received.

- Each positive acknowledgement increases the window size `cwnd` by 1/`cwnd`. Therefore, during the round-trip time T, during which `cwnd` packets are sent, we expect to increase `cwnd` by 1.
- Each congestion indication halves the window size. Initially, congestion indication was by dropping packets; in later implementations, however, congestion is flagged by setting a bit in the packet header (and the header of the corresponding ACK packet). This has the advantage that the packet doesn't need to be resent.

The Crowcroft and Oechslin's modification is MulTCP: users are allowed to set a parameter m, which

- multiplies by m the rate of additive increase
- makes the multiplicative decrease factor $1 - \frac{1}{2m}$

in Jacobson's algorithm: i.e., the user is pretending to be m independent TCP streams.

The differential equations for MulTCP are

$$\frac{d}{dt}x_r(t) = \frac{m_r}{T_r^2} - \left(\frac{m_r}{T_r^2} + \frac{x_r^2}{2m_r}\right) \sum_{j\in r} \mu_j(r).$$

Note that the response to dropped packets will change the window multiplicatively, but dropped packets will occur at a rate proportional to x_r – hence the appearance of x_r^2.

To derive this, note that for MulTCP the expected change in the congestion window `cwnd` per update step is approximately

$$\frac{m}{\texttt{cwnd}}(1-p) - \frac{\texttt{cwnd}}{2m}p$$

where p is the probability of congestion indication at the update step. Since the time between update steps is approximately $T/\texttt{cwnd}$, we get that the change in rate x per unit time is about

$$\frac{\left(\frac{m}{\texttt{cwnd}}(1-p) - \frac{\texttt{cwnd}}{2m}p\right)/T}{T/\texttt{cwnd}} = \frac{m}{T^2} - \left(\frac{m}{T^2} + \frac{x^2}{2m}\right)p.$$

Remark. Approximating TCP by a differential equation looks ambitious, because it doesn't look very smooth. On the other hand, if there are lots of connections, then the aggregate behaves like MulTCP with a large value of m, and is much smoother.

Theorem 7.4 (Global stability)**.**

$$\mathcal{U}(x) = \sum_r \frac{\sqrt{2}m_r}{T_r} \arctan\left(\frac{x_r T_r}{\sqrt{2}m_r}\right) - \sum_{j\in J}\int_0^{\sum_{s:j\in s} x_s} p_j(y)dy$$

is a Lyapunov function for the MulTCP differential equations. The unique value x maximizing $\mathcal{U}(x)$ is a stable point of the system, to which all trajectories converge.

Proof.

$$\frac{\partial}{\partial x_r}\mathcal{U}(x) = \frac{m_r}{T_r^2}\left(\frac{m_r}{T_r^2} + \frac{x_r^2}{2m_r}\right)^{-1} - \sum_{j\in r} p_j\left(\sum_{s:j\in s} x_s\right)$$

and so

$$\frac{d}{dt}\mathcal{U}(x(t)) = \sum_{r\in R}\frac{\partial \mathcal{U}}{\partial x_r}\frac{d}{dt}x_r(t) = \sum_{r\in R}\left(\frac{m_r}{T_r^2} + \frac{x_r^2}{2m_r}\right)^{-1}\left(\frac{d}{dt}x_r(t)\right)^2 \geq 0.$$

We finish the proof as before. □

7.5. **Current TCP algorithm?** To what extent is the TCP algorithm optimizing the right thing? The rate allocated to route r will be approximately

$$x_r = \frac{c}{T_r\left(\sum_{j\in r} p_j\right)^{1/2}}$$

This rate is optimal for a system where

$$U_r(x_r) = \text{constant} - \frac{c^2}{T_r^2 x_r},$$

(i.e. rapidly diminishing returns as x_r increases, and an inverse square law with distance), and

$$C_j(y) = \int_0^y p_j(\eta)d\eta$$

(if $p_j(\cdot)$ is the drop probability, we would prefer to see $yp_j(y)$ here).

That is, the resource share for route r is proportional to

$$\frac{1}{T_r\left(\sum_{j\in r} p_j\right)^{1/2}}.$$

Relative to proportional fairness, this over-penalizes the distance (we may well have physically long high capacity optical fibres, which end up being underused), and underpenalises the use of congested resources (the exponent 1/2 should be 1). Current implementations of TCP increase the exponent, and try to reduce the dependence on round-trip time.

7.6. **Cache location.** Consider the network in Figure 46.

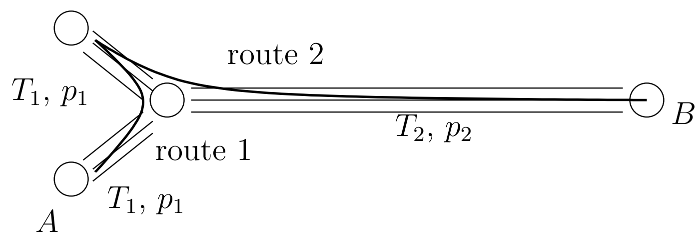


FIGURE 46. A network with a long and a short route

Recall that throughput achieved by a file transfer is approximately

$$\theta \sim \frac{1}{T\sqrt{p}},$$

where T is the round-trip time and p is the packet drop probability (a measure of congestion). Now consider the two routes indicated in the figure. On route 1, we have $T = 2T_1$ and $p \approx 2p_1$ (we are assuming p_1 and p_2 are small). On route 2, we have $T = T_1 + T_2$ and $p \approx p_1 + p_2$.

Now suppose $p_2 = 0$ (resource 2 is underused), but $T_2 = 100T_1$. Then

$$\frac{\theta_1}{\theta_2} = \frac{T_1 + T_2}{2T_1}\frac{\sqrt{p_1 + p_2}}{\sqrt{2p_1}} = \frac{101}{2\sqrt{2}} \approx 36.$$

Suppose we wanted to place a cache with lots of storage somewhere in this network, and we had a choice between nodes A and B. If we want the users to have a large throughput to this cache, we would place it at node A. However, then a user accessing the cache would use two of the congested resources, whereas if we were to place the cache at node B, the users would only be using one of the congested links. Thus, in some sense from the system point of view it would be better to have the cache at B.

This is a general problem with TCP: it has a strong tendency to overload the edges of the network and to underload the high-throughput long-distance core.

Remark. One of the current debates about the internet is on *net neutrality*: that is, are the people who lay the high-capacity optic cables allowed to charge, e.g., Amazon more money for using them (since Amazon makes quite a lot of money from using them). The suggestion is that the economic incentives in a neutral network are wrong. The calculation we have just done is a different way in which TCP skews the economic incentives for what capacity gets built where, and is in no way going to be solved by a non-neutral Internet.

7.7. **A dual algorithm.** Corresponding to the dual of the NETWORK problem, we have the dual algorithm, which adjusts

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

where

$$x_r(t) = \frac{w_r}{\sum_{k \in r} \mu_k(t)}.$$

The other parameters are $\kappa_j > 0$ and $q_j(\cdot)$, which could be $q_j(\eta) = p_j^{-1}(\eta)$, and in general is continuous, strictly increasing, with $q_j(0) = 0$. The parameter q_j is the "expected" flow rate through resource j at the current "price" μ_j.

This corresponds to keeping the "intelligence" of the system at the resources: the end systems simply do as they are told (maintaining $x_r = \frac{w_r}{\sum_{k \in r} \mu_k(t)}$), and the resources do the adjusting of μ_j. In the primal algorithm, the philosophy was that the intelligence resides at the end systems, and the network resources simply maintain values μ_j.

Having defined this dynamical system, we check its stability properties. Let

$$\mathcal{V}(\mu) = \sum_{r \in R} w_r \log \left(\sum_{j \in r} \mu_j \right) - \sum_{j \in J} \int_0^{\mu_j} q_j(\eta) d\eta.$$

Note that the first term of this is the objective of the dual problem, and the second term is the relaxation of the constraints. Now,

$$\frac{d}{dt}\mathcal{V}(\mu(t)) = \sum_{j \in J} \frac{\partial \mathcal{V}}{\partial \mu_j} \frac{d}{dt}\mu_j(t) = \sum_{j \in J} \kappa_j \mu_j(t) \left(\sum_{r : j \in r} x_r(t) - q_j(\mu_j(t)) \right)^2 \geq 0$$

with equality only at the equilibrium point. Since $\mathcal{V}(\mu)$ is continuous, strictly concave with interior maximum (it tends to $-\infty$ if $\mu_j \to 0$ or $\mu_j \to \infty$), the equilibrium point is unique and the system converges to it.

If $q_j(\eta) = C_j I[\eta > 0]$, we would get precisely the dual problem: the expected flow is precisely C_j at any positive price. Of course, this q_j violates our assumptions: it is neither strictly increasing nor continuous. This means that $\mathcal{V}$ will not be strictly concave. In particular, at the optimum, the x_r (flows) will be unique, but all we will be able to tell about μ_j is whether it is positive or not. This isn't a very healthy way to run the system: in particular, if $\mu_k = 0$ for all $k \in r$ then we set $x_r(t) = \infty$, which doesn't look very stable. [This is different from what was said in the lecture, but I believe it more.]

The equation in this case becomes

$$\frac{d}{dt}\mu_j(t) = \kappa_j \mu_j(t) \underbrace{\left(\sum_{r : j \in r} x_r(t) - C_j \right)}_{\text{excess demand}}$$

In economics, this is known as the "tâtonnement process" ("tâtonnement" is French for "groping"). It was studied in the 60's and 70's, but then the research got stuck because this describes the equilibrium of a system without providing a sensible mechanism for convergence to the equilibrium (in a real economic system, there are all sorts of issues like delays and noise that mess with this model, and it's not clear what they will be with real people in the system).

How do we set κ? In this model, the only effect that increasing κ will have is that we will converge to equilibrium faster.

However, in a real system we will have delays: it takes some time for information about μ_j to affect x_r. This means that for very large κ the system will become unstable.

Remark. Let $D_j(\xi)$ be the demand (not delay!) on route r when the price on route r is ξ, and let $p_j(y_j)$ be the target price on resource j when the load on j is y_j. If $\mu_j(t)$ is the actual price of resource j, how do we choose $\frac{d}{dt}\mu_j(t)$?

Consider

$$\mathcal{V}(\mu) = \sum_{r\in R} \int^{\sum_{j\in r}\mu_j} D_r(\xi)d\xi - \sum_{j\in J}\int_0^{\mu_j} p_j^{-1}(\eta)d\eta,$$

then

$$\frac{\partial}{\partial \mu_j}\mathcal{V}(\mu) = \sum_{r:j\in r} D_r(\sum_{k\in r}\mu_k) - p_j^{-1}(\mu_j).$$

We want to set $\frac{d\mu_j(t)}{dt}$ so that $\mathcal{V}$ is a Lyapunov function, i.e. so that

$$\frac{d}{dt}\mathcal{V}(\mu) = \sum_{j\in J}\frac{\partial V}{\partial \mu_j}\frac{d}{dt}\mu_j(t) \geq 0.$$

Note that $\frac{\partial}{\partial \mu_j}\mathcal{V}(\mu) > 0$ if and only if $p_j(\sum_{r:j\in r} D_r(\cdot)) > \mu_j$ (we assume that p_j is strictly increasing to get strict inequalities). Therefore, $\frac{d}{dt}\mathcal{V}(\mu)$ will be ≥ 0 provided

$$\frac{d}{dt}\mu_j(t) \gtrless 0 \text{ according as } p_j(y_j) \gtrless \mu_j$$

which is a very sensible update mechanism: if the price is smaller than the target, increase the price. That is, for the system without delays at least, the precise functional form of $\frac{d}{dt}\mu_j(t)$ doesn't particularly matter.

7.8. **Time delays.** We analyze a very simple, single resource model:

$$\frac{d}{dt}x(t) = \kappa\Big(w - x(t-T)p(x(t-T))\Big)$$

This gives us a differential equation, which we will linearize. Let $x(t) = x + u(t)$, where x solves $w = xp(x)$ (i.e. is the equilibrium point). Let $p = p(x)$ and $p' = p'(x) = \frac{d}{dx}p(x)$. Thus, linearizing the equation, we find

$$\frac{d}{dt}u(t) = \kappa\Big(w - \big(x + u(t-T)\big)\big(p + p'u(t-T)\big)\Big) = -\kappa(p + xp')u(t-T).$$

We now recall what the solutions to

$$\frac{d}{dt}u(t) = -\alpha u(t-T)$$

look like for various values of α:

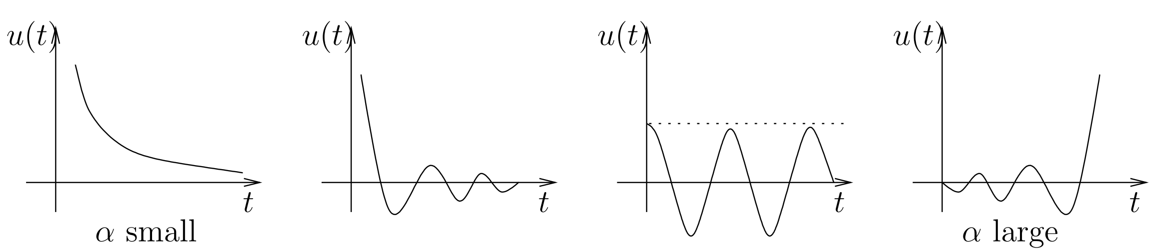


FIGURE 47. Solutions of the differential equation for different values of α

Let's find the value of α for which the periodic solution is possible: we try $u(t) = \sin(\lambda t)$. Then we need

$$\lambda \cos \lambda t = -\alpha \sin \lambda(t-T) = -\alpha \left(\sin \lambda t \cos \lambda T - \cos \lambda t \sin \lambda T\right)$$

which tells us that $\cos \lambda T = 0$, and $\alpha = \lambda = \frac{\pi}{2}\frac{1}{T}$. (Of course, we could have gotten $\alpha \sim \frac{1}{T}$ from dimensional analysis alone.) Thus, our increase in the gain κ is limited by the fact that $\kappa \cdot T$ cannot get too large.

Incidentally, if we try in general for a solution of the form $u(t) = e^{-\lambda t}$ then we will find

$$-\lambda e^{-\lambda t} = -\alpha e^{-\lambda(t-T)} = -\alpha e^{\lambda T} e^{-\lambda t}$$

and we can check that for $\alpha > \frac{1}{e}\frac{1}{T}$ this has no real solutions λ. This tells us where the transition between the first (overdamped) and the second (underdamped but still stable) region occurs.

The linearisation of the network version of this problem is

$$\frac{d}{dt}u_r(t) = -\kappa_r \left(u_r(t - T_r)\frac{w_r}{x_r} + x_r \sum_{j\in r} p_j' \sum_{s:j\in s} u_s(t - T_{jr} - T_{sj}) \right)$$

Here, T_{jr} is the time it takes for information to travel from resource j to the end system on route r, and T_{sj} is the time it takes for information to travel from end system on route s to resource j. We always have $T_{rj} + T_{jr} = T_r$, the round-trip time.

This gives stability conditions along the lines of $\kappa_r T_r$(constant depending on r) $< \pi/2$.

8. Flow level internet models

What happens to the triple (A, J, R) when someone's web page finishes downloading? The set R has one element r removed from it; the matrix A has the corresponding column removed.

We expect the rest of the system to find the new equilibrium very quickly – the time scales here are the speed-of-light delays.

Our goal now is to analyze the *flow-level* process, in which we are interested in the files being downloaded across the network. We assume that, given (A, J, R) the system finds its equilibrium rates instantaneously, and look at how this affects the rate at which the files themselves go through the network. For example, if we had a single resource, then we would be looking at a single-server queue running (some form of) processor sharing.

Let n_r be the number of flows on route r ($r \in R$), and let x_r be the rate of each flow on route r.

Suppose $(x_r, r \in R)$ is (always) chosen as the optimum of

$$\begin{aligned} &\max \sum_r w_r n_r \frac{x_r^{1-\alpha}}{1-\alpha} \\ &\text{s.t. } \sum_{r:j\in r} n_r x_r \le C_j,\ j \in J \\ &x_r \ge 0,\ r \in R \end{aligned} \tag{26}$$

Remark. (1) For $\alpha = 1$ we take $\frac{x^{1-\alpha}}{1-\alpha} = \log x$. (This has the correct derivative.)

(2) The fairness criterion for $0 < \alpha < \infty$ collects the following information:

- as $\alpha \to 0$ with $w_r \equiv 1$, this is maximizing the total flow, or throughput

- for $\alpha = 1$ with $w_r \equiv 1$, we recover proportional fairness
- for $\alpha = 2$ with $w_r = \frac{1}{T_r^2}$ we get "TCP fairness" (recall the TCP algorithm produced flow rates that optimized a certain optimization problem)
- as $\alpha \to \infty$ with $w_r \equiv 1$, we converge to max-min fairness

(3) We can rewrite the objective function as

$$\sum_r w_r n_r^\alpha \frac{(n_r x_r)^{1-\alpha}}{1-\alpha}$$

where $n_r x_r = X_r$ is the total flow rate on route r.

Write $\mathbf{x}(\mathbf{n})$ for the unique solution of (26).
Now suppose $\mathbf{n} = (n_r, r \in R)$ is a Markov process with transition rates

$$n_r \to n_r + 1 \text{ at rate } \nu_r, \quad n_r \to n_r - 1 \text{ at rate } \mu_r n_r x_r(\mathbf{n}).$$

Remark. (1) This model corresponds to *files* arriving on route r at rate ν_r. Each file has a work requirement (size distribution) that is exponential with parameter μ_r. All files on route r are transmitted at rate $x_r = x_r(\mathbf{n})$ (where $\mathbf{n}$ changes either at an arrival or at a departure).

(2) We are assuming *time-scale separation*: that is, given $\mathbf{n}$ (the number of flows), the packet-level mechanism *instantly* achieves the fairness criterion with the rates x_r, $r \in R$. Also, unlike our Lyapunov function analysis before, which had a penalty function as we approached capacity, here we are taking the ideal (sharp) capacity constraints.

(3) Although we are using exponential file sizes here, the results hold more broadly (this looks like a processor sharing discipline, which in a single-server queue is insensitive to the distribution of the service times).

Let $\rho_r = \nu_r / \mu_r$ be the load on route r.

Theorem 8.1 (Bonald, Massoulié)**.** *The Markov process* $\mathbf{n}$ *is positive recurrent (i.e., has an equilibrium distribution) if and only if*

(27)
$$\sum_{r:j\in r} \rho_r < C_j \text{ for all } j \in J.$$

Sketch of proof. First, if the stability conditions (27) are violated for some j, we argue that the process is transient by comparing it to a single $M/M/1$ queue on link j. (If j were the only capacity constraint on the network, i.e. all other links had infinite capacity, then the resulting system would look like a transient $M/M/1$ queue. Now, the actual capacity constraints cannot increase the rate at which work is processed on link j – they can slow it down, if a bottleneck elsewhere forces j to idle some of the time.)

Now, suppose the stability conditions are satisfied, and consider the drift of $\mathbf{n}$:

$$\mathbb{E}[n_r(t+\delta t) - n_r(t) | \mathbf{n}(t)] = \nu_r - \mu_r X_r(t)$$

Recall that $X = (X_r,\ r \in R)$ is the unique solution to the optimization problem

$$
\begin{aligned}
&\max \sum_r w_r n_r^\alpha \frac{X_r^{1-\alpha}}{1-\alpha} \\
&\text{s.t.} \sum_{r:j\in r} X_r \le C_j,\ j \in J \qquad (28)\\
&X_r \ge 0,\ r \in R
\end{aligned}
$$

Let

$$F(\mathbf{u}) = \sum_r \frac{w_r}{\mu_r}\rho_r^{-\alpha}\frac{u_r^{\alpha+1}}{\alpha+1}.$$

We will show that $F(\mathbf{n}(t))$ is (more or less) a Lyapunov function for the system. Note that

$$\frac{\partial F}{\partial u_r} = \frac{w_r}{\mu_r}\rho_r^{-\alpha}u_r^\alpha,\ r \in R$$

and therefore,

$$\mathbb{E}[F(\mathbf{n}(t+\delta t)) - F(\mathbf{n}(t))|\mathbf{n}(t)] \approx \sum_r \frac{w_r}{\mu_r}\rho_r^{-\alpha}n_r^\alpha(\nu_r - \mu_r X_r) = \sum_r w_r\rho_r^{-\alpha}n_r^\alpha(\rho_r - X_r) \qquad (29)$$

Consider now

$$G(\mathbf{u}) = \sum_r w_r n_r^\alpha \frac{u_r^{1-\alpha}}{1-\alpha},$$

the objective function of the optimization problem. Since X attains the maximum of $G(\cdot)$ in the feasible region of (28), we have that for every $\mathbf{u}$ in the feasible region,

$$G'(X)\cdot(\mathbf{u} - X) \le 0$$

and since G is concave,

$$G'(\mathbf{u})\cdot(\mathbf{u} - X) \le 0 \qquad (30)$$

Now, if the stability conditions (27) are satisfied, then $\exists\epsilon > 0$ such that $\mathbf{u} = (\rho_r(1+\epsilon),\ r \in R)$ is in the feasible region. Therefore, by (30),

$$\sum_r w_r n_r^\alpha(\rho_r(1+\epsilon))^{-\alpha}(\rho_r(1+\epsilon) - X_r) \le 0.$$

Combining this with (29), we get

$$\mathbb{E}[F(\mathbf{n}(t+\delta t)) - F(\mathbf{n}(t))|\mathbf{n}(t)] \le -\epsilon\sum_r w_r n_r^\alpha \rho_r^{-\alpha+1},$$

which morally is enough (it shows that F has negative drift when $\mathbf{n}$ is large, which means that trajectories of $\mathbf{n}$ ought to stay near the origin – to be more precise about it we would need to bound the jumps away from the origin as well). ☐

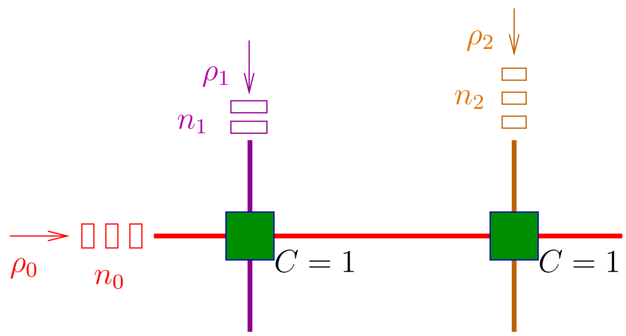


FIGURE 48. A system with three flows and two resources

8.1. **What can go wrong?** With any form of weighted α-fair allocation, as we saw last time, the stability condition is $\rho_0 + \rho_1 < 1$, $\rho_0 + \rho_2 < 1$. Provided this condition is satisfied, the Markov chain is positive recurrent, and hence has a stationary distribution.

Suppose that streams 1 and 2 are given absolute priority at their resources: that is, if $n_1 > 0$ then $n_1 x_1 = 1$ (and hence $n_0 x_0 = 0$), and if $n_2 > 0$ then $n_2 x_2 = 1$ (and again $n_0 x_0 = 0$). Then stream 0 will only get served if $n_1 = n_2 = 0$, i.e. there is no work in any of the high-priority streams.

What is the new stability condition? Resource 1 is occupied by flow 1 a proportion ρ_1 of the time; independently, resource 2 is occupied by flow 2 a proportion ρ_2 of the time. (Note that neither flow 1 nor flow 2 "sees" flow 0.) Therefore, both of these resources are free for flow 0 to use a proportion $(1-\rho_1)(1-\rho_2)$ of the time, and the stability condition is

$$\rho_0 < (1-\rho_1)(1-\rho_2).$$

This is a strictly smaller stability region than before, as illustrated in Figure 49, which shows a slice of the stability region for a fixed value of ρ_0.

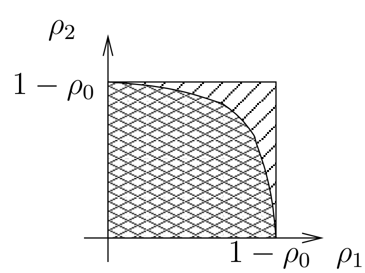


FIGURE 49. Stability regions under α-fair and priority schemes

This phenomenon is called *failure to realize capacity.* It results because there is starvation of resources: that is, high-priority flows block the other resources from working.

Remark.

- An infinite buffer between resources 1 and 2 would decouple the resources to some extent. However, realistically on the Internet buffer sizes have been scaling more slowly than flow rates. Also, putting buffering between the resources would increase the round-trip time, and we saw that this has bad effects on the stability of the system.
- If in the weighted α-fair scheme we gave flows 1 and 2 very high weights relative to flow 0, why wouldn't we reproduce this system? The reason is that we are asking the

question of positive recurrence of a Markov chain, so it doesn't matter what happens for any finite size. In any α-fair scheme, if n_0 gets sufficiently large (and provided $w_0 > 0$), it will eventually receive service.

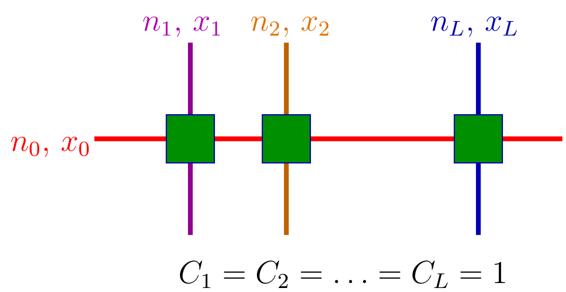


FIGURE 50. Linear network with L resources and $L + 1$ flows

Example 8.1 (Linear network with proportional fairness)**.** We consider the network in Figure 50 with $\alpha = 1$ and $w_i = 1$ for all i (i.e., the simple proportionally fair model). The optimal allocation is

$$n_0 x_0 = \frac{n_0}{n_0 + \sum_{l=1}^{L} n_l} = 1 - n_l x_l, \; l = 1, \ldots, L.$$

Indeed, we check that this optimized $\sum n_i \log x_i$. Clearly, if $n_l > 0$ then $n_0 x_0 + n_l x_l = 1$, as otherwise we can increase x_l. Therefore, the optimization problem becomes

$$\max n_0 \log x_0 + \sum_{l=1}^{L} n_l \log\left(\frac{1 - n_0 x_0}{n_l}\right)$$

Differentiating, at the optimum we have

$$\frac{n_0}{x_0} = \sum_{l=1}^{L} \frac{n_l n_0}{1 - n_0 x_0} \implies 1 - n_0 x_0 = x_0 \sum_{l=1}^{L} n_l$$

or

$$x_0 = \frac{1}{n_0 + \sum_{l=1}^{L} n_l}, \quad n_0 x_0 = \frac{n_0}{n_0 + \sum_{l=1}^{L} n_l} = 1 - n_l x_l.$$

We now compute the equilibrium distribution for this system explicitly. The general formulas are

$$q(\mathbf{n}, \mathbf{n} + e_r) = \nu_r \quad q(\mathbf{n}, \mathbf{n} - e_r) = x_r(\mathbf{n}) n_r \mu_r$$

and therefore for us

$$q(\mathbf{n}, \mathbf{n} - e_0) = \mu_0 \frac{n_0}{n_0 + \sum_{l=1}^{L} n_l}, \quad n_0 > 0$$

and

$$q(\mathbf{n}, \mathbf{n} - e_i) = \mu_i \frac{\sum_{l=1}^{L} n_l}{n_0 + \sum_{l=1}^{L} n_l}, \quad n_i > 0, \; i = 1, \ldots, L.$$

Claim. *The stationary distribution is given by*

$$\pi(\mathbf{n}) = B \binom{n_0 + \sum_{l=1}^{L} n_l}{n_0} \rho_0^{n_0} \prod_{l=1}^{L} \rho_l^{n_l},$$

where B is a normalisation constant.

Proof. We check the detailed balance equations.

$$\pi(\mathbf{n})\underbrace{q(\mathbf{n},\mathbf{n}+e_0)}_{\nu_0} = \underbrace{\pi(\mathbf{n}+e_0)}_{\pi(\mathbf{n})\frac{n_0+\sum_{l=1}^L n_l+1}{n_0+1}\frac{\nu_0}{\mu_0}} \underbrace{q(\mathbf{n}+e_0,\mathbf{n})}_{\mu_0\frac{n_0+1}{n_0+\sum_{l=1}^L n_l+1}}$$

and

$$\pi(\mathbf{n})\underbrace{q(\mathbf{n},\mathbf{n}+e_i)}_{\nu_i} = \underbrace{\pi(\mathbf{n}+e_i)}_{\pi(\mathbf{n})\frac{n_0+\sum_{l=1}^L n_l+1}{\sum_{l=1}^L n_l+1}\frac{\nu_i}{\mu_i}} \underbrace{q(\mathbf{n}+e_i,\mathbf{n})}_{\mu_i\frac{\sum_{l=1}^L n_l+1}{n_0+\sum_{l=1}^L n_l+1}}$$

as required. □

In this case, we can also compute the normalization constant B:

$$\sum_{n_0,n_1,\dots,n_L}\binom{n_0+n_1+\ldots+n_L}{n_0}\rho_0^{n_0}\rho_1^{n_1}\ldots\rho_L^{n_L}$$

$$=\sum_{n_1,\dots,n_L}\rho_1^{n_1}\ldots\rho_L^{n_L}\sum_{n_0}\binom{n_0+\ldots+n_L}{n_0}\rho_0^{n_0} \quad \text{negative binomial expansion}$$

$$=\sum_{n_1,\dots,n_L}\rho_1^{n_1}\ldots\rho_L^{n_L}\frac{1}{(1-\rho_0)^{n_1+\ldots+n_L+1}}$$

$$=\frac{1}{1-\rho_0}\sum_{n_1,\dots,n_L}\left(\frac{\rho_1}{1-\rho_0}\right)^{n_1}\left(\frac{\rho_2}{1-\rho_0}\right)^{n_2}\ldots\left(\frac{\rho_L}{1-\rho_0}\right)^{n_L} \quad \text{provided } \rho_i<1-\rho_0 \text{ for all } i$$

$$=\frac{(1-\rho_0)^{L-1}}{\prod_{l=1}^L(1-\rho_0-\rho_l)}=B^{-1}.$$

Putting everything together, in this example we have

$$\pi(\mathbf{n})=\frac{\prod_{l=1}^L(1-\rho_0-\rho_l)}{(1-\rho_0)^{L-1}}\binom{\sum_{l=0}^L n_l}{n_0}\prod_{l=0}^L\rho_l^{n_l}$$

for $\mathbf{n}\in\mathbb{Z}_+^{L+1}$.

Remark. (1) A linear network is what a system with diverse routing looks like to a single file crossing the network, so it ought to be quite a useful model.

(2) This is an example of a quasi-reversible queue. That is, if we draw a large box around the linear network, then arrivals of each file type will be Poisson processes (by assumption), but departures will be as well. (We saw this for a sequence of $M/M/1$ queues at the beginning of the course.) This lets us embed this into a bigger network as a unit, in the same way we did with $M/M/1$ queues (so we could build an open migration network of these, say). It also allows us to conclude an insensitivity result: that is, the invariant distribution π does not actually depend on the file size distribution except through its mean.

(3) The mean time that a job stays in the system is proportional to its size. Indeed, suppose that instead of sending a file of length 100 (say) through the network, we actually sent through a file of length 1; once it gets through the network, we send it back to the start, and repeat 100 times. From the point of view of the network, these

two processes are identical – the network doesn't see the file size explicitly, only the number of files.

(4) Define the *flow throughput* on route r as

$$\frac{\text{average file size on route } r}{\text{average time required to transmit a file on route } r}.$$

The average file size we know is $\frac{1}{\mu_r}$. The average time can be derived from Little's formula $L = \lambda W$, which tells us

$$\mathbb{E}(n_r) = \nu_r \cdot \text{average time required to transmit a file on route } r.$$

Combining these, and recalling $\rho_r = \nu_r/\mu_r$, we get

$$\text{throughput} = \frac{\rho_r}{\mathbb{E}n_r}.$$

Since we know $\pi(\mathbf{n})$, we can calculate this throughput.

Exercise 6. Show that the flow throughput for the linear network is

$$\begin{cases} 1-\rho_0-\rho_l, & l = 1,\dots,L \\ \frac{1-\rho_0}{1-\sum_{l=0}^{L}\frac{\rho_0}{1-\rho_0-\rho_l}}, & l = 0. \end{cases}$$

Note that $1-\rho_0-\rho_l$ is the proportion of time that resource l is idle. Thus, the throughput is as if the file on route l got the entire idle resource to itself a proportion $1-\rho_0-\rho_l$ of the time. (On route 0 it is a little more complicated, as some of the resources on the route may be forced to idle by congestion on other resources.)

The rest of the lecture went at the end of the TCP section (Section 7.6), and at the end of the Dual Algorithm section (Section 7.7).