Modelling data networks

Richard G. Clegg (richard@richardclegg.org)

1 Introduction to network modelling

There are many difficulties to modelling the Internet, for a well-known and excellent summary see [5]

- The Internet is big (and growing).
- The Internet is heterogeneous to a large degree.
- No central maps exist of the Internet.
- The Internet is not always easy to measure.
- The Internet is rapidly changing.
- It is extremely important to be able to model the Internet.

The Internet cannot possibly be modelled, yet we must model the Internet. How can this be resolved?
The intention of this lecture is to teach you three things:

1. A general approach to such modelling problems.
2. Some specific mathematical techniques necessary for certain modelling problems (basic stochastic processes, Markov chains, queuing theory).
3. An approach to how to do modelling of data networks in real life.

- How you model the network depends critically on the problem you are solving.
- What are you trying to show with your model?
- Metrics: what are we trying to measure?
    1. Throughput?
    2. Goodput (data getting to your application - without protocol overheads and retransmissions)?
    3. System efficiency?
- Validation: what real data can be used to check the model?
- Sensitivity: what happens if your assumptions change?
    1. What if the demand on the system is slightly different?
    2. What happens if delays and bandwidths are changed?
    3. What happens if users stay longer or download more?

Important questions for modelling.

1. How much of the network do we model?
    - Whole Internet (then we can't even model every computer – every AS?)
    - A few typical nodes?
    - A sub net?
    - A single queue and buffer?

2. What level of modelling is appropriate?
   - Mathematical – solution "instant" (or quick) but which mathematical techniques are useful?
   - Detailed simulation
   - Combined approach (equations abstract away some details with approximations)
3. How far down the network stack need we go?

1.1 Example model – peer-to-peer network

Modelling Task: Test the possible improvements expected if we try a locality aware peer selection policy on a global bittorrent network.
What must our model include?

1. The distribution of nodes (peers) on the overlay network (not the whole network).
2. The delay and throughput between these peers (must depend on distance to some extent).
3. How users arrive and depart.
4. What users choose to download.

Note that this might already be a vast modelling task with hundreds of thousands or even millions of nodes.

- Research existing P2P models, do any fit? Don't reinvent the wheel.
- Real data: What real-life measurements exist to validate against?
- If we are modelling a new peer selection we must be sure our model covers existing peer selection well.
- Metrics: what must we measure in our model?
  1. Overall throughput/goodput?
  2. Distribution of time taken for peers to make their download?
  3. Total resources used in system?
- Validation: Instrumented P2P clients exist – how do they compare to our simulation.
- Sensitivity: Different distribution of users? Different delays and throughputs?

1.2 Example model – Buffer provisioning model

Modelling task: Given a router with a buffer, how does the buffer size in packets affect the probability of packet loss?
What must our model include?

1. A model of the incoming packets to the buffer.
2. The rate at which packets leave the buffer.
3. Possibly distribution of packet lengths in bytes.
4. Possibly the feedback (TCP) between packet loss and arrival rate.

- Research: what is known about the statistics of Internet traffic?
- What is the distribution of inter-arrival times and packet lengths?
- Metrics:
  1. Packet loss.
  2. Packet delay.
- Sensitivity: What if we change the following parameters:
  1. The total arrival rate.
  2. The bandwidth of the outgoing link.
- Validation: Real traffic traces (CAIDA has a collection).

1.3 Example model – TCP throughput

Modelling Task: Test a possible improvement to the TCP model which aims to improve fairness and throughput when flows share a link.

What must our model include?

1. Individual packet model with existing TCP protocol as accurately as possible.
2. A reasonable estimate of how long each connection lasts and the rate at which new connections.
3. A model of the probability of round trip time for the parts of the connection not on the link being modelled.
4. A model of the probability of packet loss on the link (due to buffer overflow?)

- Can existing network models help (ns-2 could be an obvious choice)?
- What if the existing protocol shares a link with flows using the old protocol.
- Metrics:
  1. Throughput and goodput.
  2. Fairness between flows.
- Sensitivity, what if we change these parameters:
  1. Number of flows using existing and new protocol.
  2. Bandwidth of link.
  3. Round trip time of flows.
  4. Probability of packet loss.
- Validation: Does our model agree with real measurements?

1.4 Other things to model

- Of course depending on the nature of your modelling, there may well be other aspects of the network to be modelled.
- Some examples might be:
  1. Reliability of nodes and links.
  2. An overlay network.
  3. Possible hostile attacks to the network.
- In all cases, an important starting point is to find out what research already exists in the area.
- Are any real-life data sets available which could inform your modelling? Could you gather such data?

Figure 1.4 shows a basic diagram for a simple model of a generic network. Objects (packets, files, work to be performed) arrive at nodes (computers, routers, data centers). The object arrival is modelled as a stochastic process. The objects are processed at nodes (this can be seen as a queue) and then pass onwards either leaving the network or moving onwards to become an input to other nodes.

A network model could be viewed as these components.

- Arrival process: A statistical process describing how objects (packets) arrive in the network – statistical process modelling.
- Queueing process: A model which describes how objects (packets) are processed by a network node – queuing theory.
- A "network topology" – the wiring diagram which shows how these things connect together.

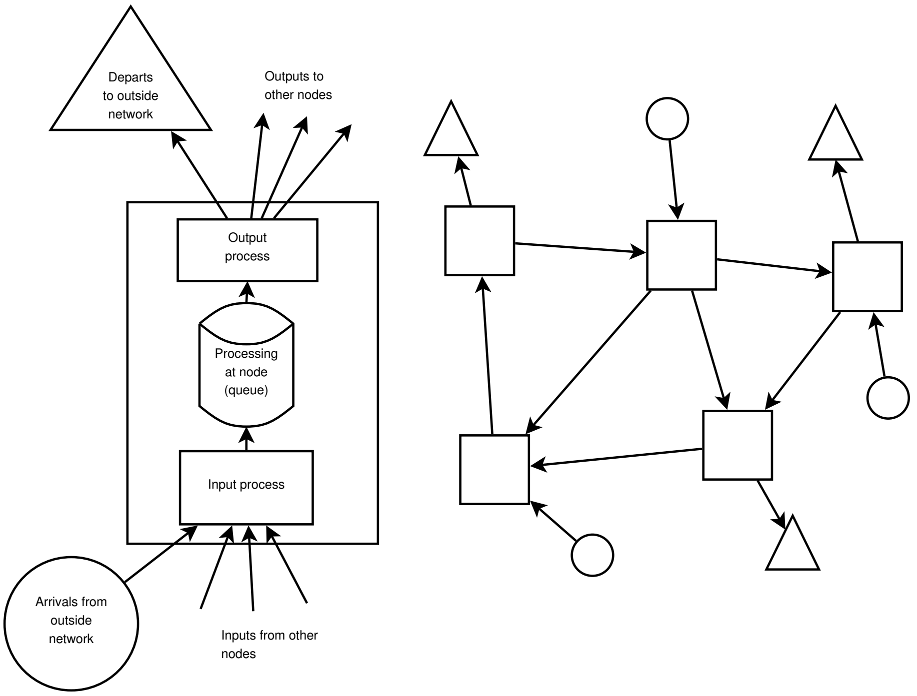


Figure 1: A diagram showing a generic model of a data network

The rest of these notes will, therefore, cover:

- Statistical processes – in particular the Poisson process.
- Markov chains – a useful modelling tool in themselves and a prerequisite for...
- Queuing theory – the mathematical study of how things join and leave queues.

Finally they will cover:

- A summary of basic research in the Internet.
- A brief description of a well-known network modelling tool – ns-2.

2 Statistical Processes

2.1 Introduction to stochastic processes

Let $X(t)$ be some value (or vector of values) which varies in time t. Think of the stochastic process as the rules for how $X(t)$ changes with t. Note: t may be discrete ($t = 0, 1, 2, \ldots$) or continuous.

A Poisson process is a process where the change in $X(t)$ from time t_1 to t_2 is a Poisson distribution, that is $X(t_2) - X(t_2)$ follows a Poisson distribution.

A simple stochastic process is the drunkard's walk (or random walk). A man walks home from the pub. He starts at a distance $X(0)$ from some point. At every step he (randomly) gets either one unit closer (probability p) or one unit further away.

$$X(t+1) = \begin{cases} X(t) + 1 & \text{probability } p \\ X(t) - 1 & \text{probability } 1 - p. \end{cases}$$

With this model one can answer questions like "where, on average, will he be at time t"? s

Take some simple parameters for the drunkard's walk – $p = 0.5, X(0) = 0$. Figure 2 shows what the system looks like with these parameters.

- What is the expected value of $X(t)$, that is, $\mathrm{E}\,[X(t)]$?

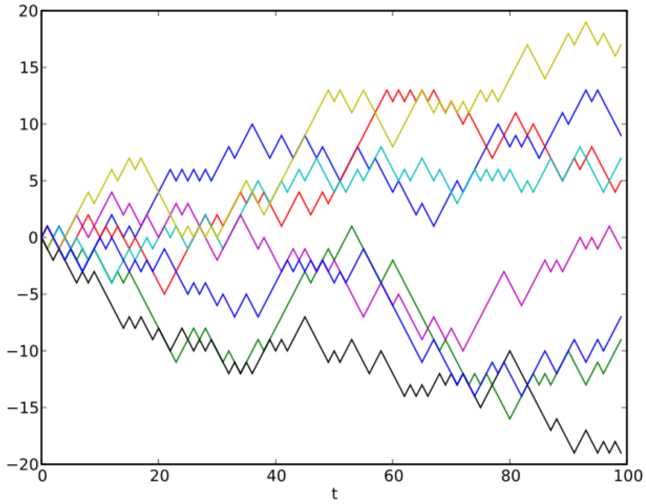


Figure 2: The drunkard's walk with $p = 0.5$, $X(0) = 0$ – diagram from Wikipedia

- $\mathrm{E}[X(t)] = 0.5(X(t-1)+1) + 0.5(X(t-1)-1) = 0.5(X(t-1)+X(t-1)) + 0.5(1-1) = X(t-1)$.
- Therefore $\mathrm{E}[X(t)] = X(0) = 0$ – the poor drunk makes no progress towards his house (on average).
- $\mathrm{E}\left[X(t)^2\right] = 0.5(X(t-1)+1)^2 + 0.5(X(t-1)-1)^2 = X(t-1)^2 + 1$.
- Therefore $\mathrm{E}\left[X(t)^2\right] = t$ – on average the drunk does get further from the starting pub.
- This seemingly silly example has many uses in physics and chemistry (Brownian motion) – not to mention gambling (coin tosses).

2.2 A note about expectation values

Definition 2.1. The *expected value* or *expectation* of the function $g(X)$ on a discrete random variable X is given by

$$\mathrm{E}[g(X)] = \sum_{i=1}^{\infty} g(x_i)\mathbb{P}[X = x_i],$$

where x_i are all the possible values of X (that is all the members of its sample space) and $\mathbb{P}[X = x_i]$ is the probability X has the value x_i is that value.

A simple example might help – what is the expected value rolled on a single die? The die has numbers 1,2,3,4,5,6 and each has probability 1/6. So if X is the number shown on the die $EX = \sum_{i=1}^{6} i(1/6)$ which is 3.5. If you roll a die you will see this is the average score.

What is the average squared score? That is, if we roll the same dice and take the square of the number, what is the average of that? $EX^2 = \sum_{i=1}^{6} i^2(1/6)$ which comes to 91/6, around 15.17.

2.3 The Poisson process

Let $X(t)$ (where $t \geq 0$ – that is t can be any real number, not just integers, this is a continuous process) with $X(0) = 0$ be a Poisson process with rate λ. Let t_2, t_1 be two times such that $t_2 > t_1$. Let $\tau = t_2 - t_1$.

$$\mathbb{P}[X(t_2) - X(t_1) = n] = \exp[-(\lambda\tau)]\left[\frac{(\lambda\tau)^n}{n!}\right].$$

In other words, the number of arrivals in some time period τ follows a Poisson distribution with rate $\lambda\tau$.

Why is the Poisson process special?

- The Poisson process is in many ways the simplest stochastic process of all.
- This is why the Poisson process is so commonly used.
- Imagine your system has the following properties:
  - The number of arrivals does not depend on the number of arrivals so far.
  - No two arrivals occur at exactly the same instant in time.
  - The number of arrivals in time period τ depends only on the length of τ.
- The Poisson process is the *only* process satisfying these conditions (see section 2.4 for proof).

Some remarkable things about Poisson processes

- The mean number of arrivals in a period τ is $\lambda\tau$ (see notes).
- If two Poisson processes arrive together with rates λ_1 and λ_2 the arrival process is a Poisson process with rate $\lambda_1 + \lambda_2$.
- If you randomly "sample" a Poisson process – e.g. pick arrivals with probability p, the sampled process is Poisson, rate $p\lambda$.
- The time before the next arrival in a Poisson process does not change depending on when you arrive.
- In general this makes Poisson processes easy to deal with.
- Many things in computer networks really are Poisson processes (e.g. people logging onto a computer or requesting web pages).
- The Poisson process is also "memoryless" as the next section explains.

If a packets are arriving in a Poisson process and a packet has just arrived what is the likely length of time before we see another packet? This is called the *interarrival time* – the time between arrivals. For a Poisson process this follows the exponential distribution. An exponential distribution for a variable T takes this form:

$$\mathbb{P}\left[T \leq t\right] = \begin{cases} 1 - \exp[-(\lambda t)], & t \geq 0, \\ 0 & t < 0. \end{cases}$$

This is easily shown – the probability of an arrival occurring before time t is one minus the probability of no arrivals occurring up until time t. The probability of no arrivals occurring during a time period t is $(\lambda t)^0 \exp[-(\lambda t)]/0! = \exp[-(\lambda t)]$. It can be shown that the mean time to the next arrival is $1/\lambda$.

There is something strange to be noticed here – the distribution of our interarrival time T was given by $\mathbb{P}\left[T \leq t\right] = 1 - \exp[-(\lambda t)$ for $t \geq 0$. However, if looked at the Poisson process at any instant and asked "how long must we wait for the next arrival?" the answer is just the same $1/\lambda$. Exactly the same argument can be made for any arrival time. The probability of no arrivals in the next t seconds does not change because an arrival has just happened. The expected waiting time for the next arrival does not change if you have been waiting for just one second, or for an hour or for many years – the average time to the next arrival is still the same $1/\lambda$. This is known as the *memoryless* property of Poisson processes

2.4 Aside – deriving the Poisson processes

This section of notes is an aside included for interest and will not be examined. There are many ways to derive a Poisson process. Let $A(t)$ (for $t \geq 0$) be the number of customers arriving in the interval $[0, t]$ (with $A(t) = 0$). Consider the following requirements,

1. The number of arrivals in disjoint time periods are independent.
2. For a small time period δt the probability of a single arrival in the period is given by

$$\mathbb{P}\left[A(t + \delta t) - A(t) = 1\right] = \lambda \delta t + o(\delta t),$$

   where λ is known as the *rate* of the process and $o(\delta t)$ is some function (which maybe negative or positive) which tends to zero in the limit as δt tends to zero.

3. The probability that two or more arrivals occur in the same small time period is negligible. More formally,
$$\mathbb{P}\left[A(t+\delta t) - A(t) \geq 2\right] = o(\delta t).$$

From the above we can derive

$$\mathbb{P}\left[A(t+\delta t) - A(t) = 0\right] = 1 - \lambda\delta t + o(\delta t).$$

Define $P_n(t) = \mathbb{P}\left[A(t) = n\right]$. Now, consider how $P_n(t)$ evolves in some short time period δt.

$$\begin{aligned} P_n(t+\delta t) &= P_n(t)(1-\lambda\delta t) + P_{n-1}(t)\lambda\delta t + o(\delta t) \qquad n > 0, \\ P_0(t+\delta t) &= P_0(t)(1-\lambda\delta t) + o(\delta t). \end{aligned}$$

These can be rewritten as

$$\begin{aligned} \frac{P_n(t+\delta t) - P_n(t)}{\delta t} &= \frac{-P_n(t)(\lambda\delta t) + P_{n-1}(t)\lambda\delta t + o(\delta t)}{\delta t} \qquad n > 0, \\ \frac{P_0(t+\delta t) - P_0(t)}{\delta t} &= \frac{P_0(t)(-\lambda\delta t) + o(\delta t)}{\delta t}. \end{aligned}$$

These equations are sometimes called differential difference equations and are often used in queuing theory. Now take the limit as $\delta t \to 0$ which causes the terms $o(\delta t)$ to vanish and gives

$$\begin{aligned} \frac{dP_n(t)}{dt} &= -\lambda P_n(t) + \lambda P_{n-1}(t) \qquad n > 0 \\ \frac{dP_0(t)}{dt} &= -\lambda P_0(t). \end{aligned}$$

Now, the equation for $P_0(t)$ can be trivially solved and this gives

$$P_0(t) = Ce^{-\lambda t},$$

where C is a constant which is shown to be 1 from the boundary condition $P_0(0) = 1$ (since $A(0) = 0$).
Substituting $n = 1$ gives,

$$\frac{dP_1(t)}{dt} = -\lambda P_1(t) + \lambda P_0(t) = -\lambda P_1(t) + \lambda e^{-\lambda t}.$$

Solving gives

$$P_1(t) = \lambda t e^{-\lambda t},$$

which can be checked by substitution into the previous equation.
Continuing this analysis will give

$$\begin{aligned} P_2(t) &= \frac{(\lambda t)^2 e^{-\lambda t}}{2!} \\ P_3(t) &= \frac{(\lambda t)^3 e^{-\lambda t}}{3!}, \end{aligned}$$

and so on.
In general the relationship

$$P_n(t) = \frac{(\lambda t)^n e^{-\lambda t}}{n!},$$

is suspected and this can be shown trivially by induction.
Now, this implies,

$$\mathbb{P}\left[A(t) = n\right] = P_n(t) = \frac{(\lambda t)^n e^{-\lambda t}}{n!},$$

and since one of our other conditions was that the arrivals in disjoint time periods are independent then we could repeat this analysis to find the number of packets arriving since some time t and get,

$$\mathbb{P}\left[A(t+\tau) - A(t) = n\right] = \mathbb{P}\left[A(\tau) = n\right] = \frac{(\lambda\tau)^n e^{-\lambda\tau}}{n!},$$

which is the equation for a Poisson process. It can therefore be seen that the three simple conditions defined in the beginning are sufficient that a Poisson process is the only process which meets those conditions.

2.5 Aside — the mean and variance of a Poisson distribution

This section of notes is included for interest and will not be examined.s Assume the variable X has a Poisson distribution with parameter λ. Its expectation is given by

$$\begin{aligned}
\mathrm{E}\left[X\right] = \sum_{n=0}^{\infty} n\mathbb{P}\left[X=n\right] &= \sum_{n=1}^{\infty} n\frac{\lambda^n e^{-\lambda}}{n!} \\
&= \sum_{n=1}^{\infty} \lambda\frac{\lambda^{n-1} e^{-\lambda}}{(n-1)!} \\
&= \lambda\sum_{n=0}^{\infty} \frac{\lambda^n e^{-\lambda}}{n!} \\
&= \lambda.
\end{aligned}$$

A similar calculation will show that $\overline{X}$ (the sample mean) is an unbiased estimator of λ. Proceeding in the same way,

$$\begin{aligned}
\mathrm{E}\left[X^2\right] = \sum_{n=0}^{\infty} n^2\mathbb{P}\left[X=n\right] &= \sum_{n=1}^{\infty} n^2\frac{\lambda^n e^{-\lambda}}{n!} \\
&= \lambda\sum_{n=1}^{\infty} n\frac{\lambda^{n-1} e^{-\lambda}}{(n-1)!} \\
&= \lambda\sum_{n=0}^{\infty} (n+1)\frac{\lambda^n e^{-\lambda}}{n!} \\
&= \lambda\sum_{n=0}^{\infty} \left[n\frac{\lambda^n e^{-\lambda}}{n!} + \frac{\lambda^n e^{-\lambda}}{n!}\right] \\
&= \lambda^2 + \lambda.
\end{aligned}$$

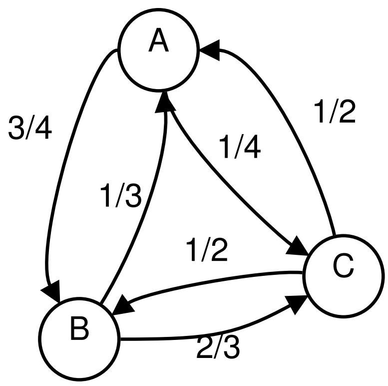
The variance is therefore given by

$$\sigma^2 = \mathrm{E}\left[X^2\right] - \mathrm{E}\left[X\right]^2 = \lambda.$$

3 Markov chains

3.1 What is a Markov Chain?

Now consider the idea of things moving about randomly in space. For example consider a slightly bewildered professor who can never remember what lecture theatre he is supposed to be in. Every day he tries a new lecture theatre to see if his class is there. He changes according to the diagram below.



Now, the next step is to ask questions about where the professor is at a given day. If he starts in room A on day one then on day 2 he has a 1/4 chance of being in C and a 3/4 chance of being in B . On day 3 he has a 3/8 chance of being in A ($1/3 \times 3/4 = 1/4$ via $A \to B \to A$ plus $1/2 \times 1/4 = 1/8$ via $A \to C \to A$). We could carry on making calculations like this for day 4 day 5 and so on but it would get boring. We need a smarter way.

Define the transition probabilities. Let us number instead of name our rooms, 0, 1 and 2 instead of A, B and C. Let p_{ij} be the probability that if the prof began at i he moves to j in one day. So, for example $p_{11} = 0$ (there is zero probability the prof begins in B and ends in B). $p_{12} = 2/3$ (the probability that the prof at B moves to C is 2/3). We can now define the *transition matrix* P. The transition matrix is the matrix of these probabilities:

$$\mathbf{P} = \begin{bmatrix} p_{00} & p_{01} & p_{02} \\ p_{10} & p_{11} & p_{12} \\ p_{20} & p_{21} & p_{22} \end{bmatrix}.$$

Assume we know some initial vector of probabilities of his location, for example, we know he is definitely in lecture room 0 and call this $\boldsymbol{\lambda}_0$.

$$\boldsymbol{\lambda}_0 = [100].$$

In general define $\lambda_{i,j}$ as the probability the prof is in state j on day i and $\boldsymbol{\lambda}_i = [\lambda_{i,0}\lambda_{i,1}\lambda_{i,2}]$ as the vector of all states on day i. We can then use the following equation

$$\boldsymbol{\lambda}_1 = \boldsymbol{\lambda}_0 \mathbf{P},$$

This does not seem to help much but we can also say

$$\boldsymbol{\lambda}_i = \boldsymbol{\lambda}_{i-1} \mathbf{P},$$

This tells us how to work out the probabilities of where the prof is on a given day when we have the starting probabilities and the transition matrix.

We might be interested in something more final such as where the prof "ends up". Define $\boldsymbol{\pi}$ as the vector $[\pi_0, \pi_1, \pi_2]$ where $\pi_j = \lim_{i \to \infty} \lambda_{i,j}$ that is π_j is the probability at the "final" day that the prof is in room j. Two questions arise: Does this exist and does it depend on where the prof was on the first day. The answer for most Markov chains usually encountered (those which are connected, finite and not periodic) is that these probabilities do exist and that they do not depend on the initial conditions.

These probabilities are known as equilibrium probabilities and can be simply calculated in a lot of cases using what are know as the equilibrium conditions – that is $\sum_i \pi_i = 1$ (the probabilities sum to one) and for all j then the flow into a state is equal to the probability of the state $\sum_j \pi_j p_{ji} = \pi_i$. In matrix terms this is

$$\boldsymbol{\pi} = \boldsymbol{\pi} \mathbf{P}.$$

For our professor example this gives us these equations (remembering that $p_{00}, p_{11}, p_{22} = 0$ then

$$\begin{aligned} \pi_0 + \pi_1 + \pi_2 &= 1 && \text{probabilities sum to one} \\ \pi_1 p_{10} + \pi_2 p_{20} &= \pi_0 && \text{balance for room 0} \\ \pi_0 p_{01} + \pi_2 p_{21} &= \pi_1 && \text{balance for room 1} \\ \pi_0 p_{02} + \pi_1 p_{12} &= \pi_2 && \text{balance for room 2} \end{aligned}$$

Note that we do need the first equation because the balance equations contain one dependent equation. These could be solved using standard techniques for simultaneous equations to give: $\pi_0 = 16/55$, $\pi_1 = 21/55$ and $\pi_2 = 18/55$.

3.2 Markov Chain simple examples

This section contains some simple Markov chain examples for practice. Before going on to do some examples, a recap.

- p_{ij} is the *transition probability* – the probability of moving from state i to state j the next iteration of the chain.
- The *transition matrix* P is the matrix of the p_{ij}.
- π_i is the *equilibrium probability* – the probability that after a "long time" the chain will be in state i.

- The sum of π_i must be one (the chain must be in some state).
- Each state has a *balance equation* $\pi_i = \sum_j \pi_j p_{ji}$ for all j. (In matrix terms $\pi_b = \boldsymbol{\pi} P$.)
- The balance equations together with the sum of π_i will solve the chain.

Note that the balance equations have n (the number of states) equations for n unknowns – however the balance equations always contain a redundant equation (from $n-1$ of them you can derive the remaining one). Exercise for student: why is this?

3.3 The google page rank example

Did you know google owes part of its success to Markov chains? "Pagerank" (named after Larry Page) was how google originally ranked search queries. Pagerank tries to work out which web page matching a search term is the most important. Pages with many links to them are very "important" but it is also important that the "importance" of the linking page counts. Here we consider a very simplified version. (Note that Larry Page is now a multi-billionaire thanks to Markov chains).

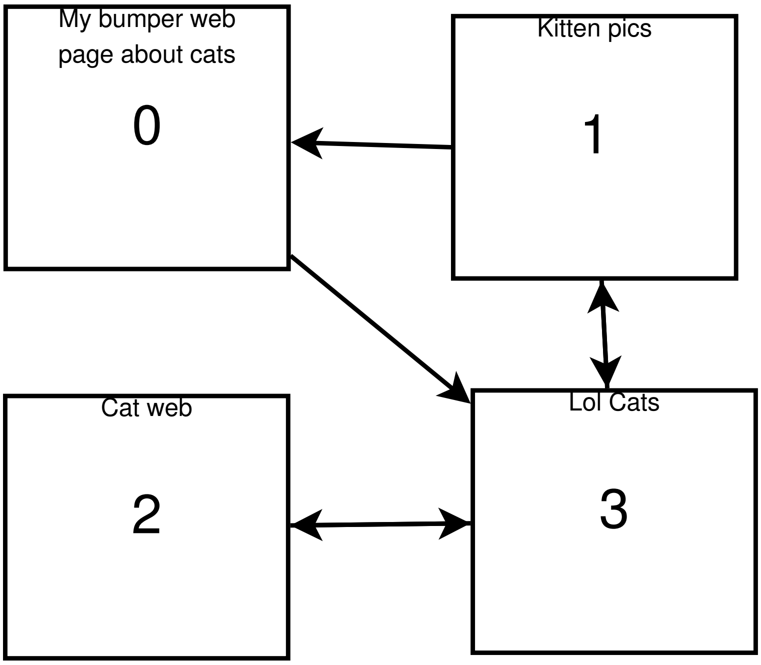


Figure 3: An example showing web pages containing the search term cat or kitten

Figure 3 shows some web pages about cats and how they link to each other. Imagine these four web pages are every web page about cats and kittens on the web. An arrow indicates a link from one page to another – e.g. "Cat Web" and "Lol cats" link to each other. Now think of a user randomly clicking on "cat" links. What page will the user visit most often – this is a Markov chain. "Lol cats" links to two other pages so 1/2 probability of visiting "Cat" next. Cat web only links to "Lol cats" so probability 1 of visiting that next.

The transition matrix is

$$P = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1/2 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 1 \\ 0 & 1/2 & 1/2 & 0 \end{bmatrix}.$$

And therefore

$$\begin{aligned} \pi_0 &= \pi_1/2 \\ \pi_1 &= \pi_3/2 \\ \pi_2 &= \pi_3/2 \qquad \text{ignore redundant equation for } \pi_3 \\ \pi_0 + \pi_1 + \pi_2 + \pi_3 &= 1 \end{aligned}$$

We have $\pi_1 = \pi_2$ from lines 2 and 3. $\pi_1 = 2\pi_0 = \pi_3/2$ from line 1 and 3. $\pi_1/2 + \pi_1 + \pi_1 + 2\pi_1 = 1$ from line 4 and above lines. Therefore, $\pi_1 = 2/9 \quad \pi_0 = 1/9 \quad \pi_2 = 2/9 \quad \pi_3 = 4/9$.

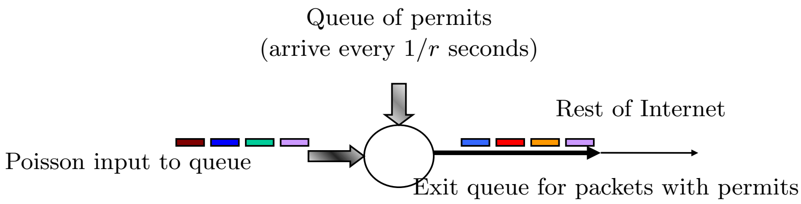
So this example shows “Lol cats” is the most important page, followed by “Cat web” and “Kitten pics” equally important. Note that pages 0,1 and 2 all have only one incoming link but are not equally important. Nowadays google has made many optimisations to their algorithm (and this is a simplified version anyway). Nonetheless this “random walk on a graph” principle remains important in many network models.

3.4 The leaky bucket example

- A “leaky bucket” is a mechanism for managing buffers and to smooth downstream flow.
- What is described here is sometimes known as a “token bucket”.
- A queue holds a stock of “permit” generated at a rate r (one permit every $1/r$ seconds) up to a maximum of W.
- A packet cannot leave the queue without a permit – each packet takes one permit.
- The idea is that a short burst of traffic can be accommodated but a longer burst is smoothed to ensure that downstream can cope.
- Assume that packets arrive as a Poisson process at rate λ.
- A Markov model will be used [3, page 515].

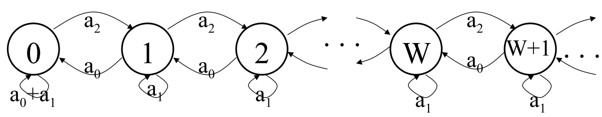
Use a discrete time Markov chain where we stay in each state for time $1/r$ seconds (the time taken to generate one permit). Let a_k be the probability that k packets arrive in one time period. Since arrivals are Poisson,

$$a_k = \frac{e^{-\lambda/r}(\lambda/r)^k}{k!}.$$



- In one time period (length $1/r$ secs) one token is generated (unless W exist) and some may be used sending packets.
- States $i \in \{0, 1, \ldots, W\}$ represent no packets waiting and $W - i$ permits available. States $i \in \{W + 1, W + 2, \ldots\}$ represent 0 tokens and $i - W$ packets waiting.
- If k packets arrive we move from state i to state $i + k - 1$ (except from state 0).
- Transition probabilities from i to j, $p_{i,j}$ given by

$$p_{i,j} = \begin{cases} a_0 + a_1 & i = j = 0 \\ a_{j-i+1} & j \geq i - 1 \\ 0 & \text{otherwise} \end{cases}$$



Let π_i be the equilibrium probability of state i. Now, we can calculate the probability flows in and out of each state.

For state one

$$\begin{aligned}\pi_0 &= a_0\pi_1 + (a_0 + a_1)\pi_0 \\ \pi_1 &= (1 - a_0 - a_1)\pi_0/a_0.\end{aligned}$$

For state $i > 0$ then $\pi_i = \sum_{j=0}^{i+1} a_{i-j+1}\pi_j$. Therefore,

$$\begin{aligned}\pi_1 &= a_2\pi_0 + a_1\pi_1 + a_0\pi_2 \\ \pi_2 &= \frac{\pi_0}{a_0}\left(\frac{(1 - a_0 - a_1)(1 - a_1)}{a_0} - a_2\right).\end{aligned}$$

In a similar way, we can get π_i in terms of $\pi_0, \pi_1, \ldots, \pi_{i-1}$.

- We could use $\sum_{i=0}^{\infty} \pi_i = 1$ to get result but this is difficult.
- Note that permits are generated every step except in state 0 when no packets arrived (W permits exist and none used up).
- This means permits arrive at rate $(1 - \pi_0 a_0)r$.
- Rate of tokens arriving must equal λ unless the queue grows forever (each packet gets a permit).
- Therefore $\pi_0 = (r - \lambda)/(ra_0)$.
- Given this we can then get π_1, π_2 and so on.

To complete the model we want to calculate T average delay of a packet.

- If we are in states $\{0, 1, \ldots, W\}$ packet exits immediately with no delay.
- If we are in states $i \in \{W + 1, W + 2, \ldots\}$ then we must wait for $i - W$ tokens $(i - W)/r$ seconds to get a token.
- The proportion of the time spent in state i is π_i.
- The final expression for the delay is

$$T = \frac{1}{r}\sum_{j=W+1}^{\infty} \pi_j(j - W).$$

- For more analysis of this model see [3, page 515].

4 Continuous time Markov chains

The Markov chain considered previously was a homogeneous discrete time Markov chain where homogeneous in this case implies that the transition probabilities remain constant and discrete time implies that the process has a certain step length (in the original bungling professor example it was assumed that the hippy travelled once and once only every day). The continuous time Markov chain weakens this assumption by allowing a transition between states of the chain to occur at any time. Instead of having discrete steps and transition probabilities, instead the states act like Poisson processes with given flow rates connecting them. Define λ_{ij} where $i \neq j$ as the flow rate between state i and j.

Consider first the discrete time Markov chain which has a time step δt and which has its transition matrix defined by

$$\mathbf{P}(\delta t) = \begin{bmatrix} 1 - p_{00}\delta t & p_{01}\delta t & p_{02}\delta t & \ldots \\ p_{10}\delta t & 1 - p_{11}\delta t & p_{12}\delta t & \ldots \\ p_{20}\delta t & p_{21}\delta t & 1 - p_{22}\delta t & \ldots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

In order that this is a Markov chain then it must be the case that $p_{ii} = \sum_{j \neq i} p_{ij}$ in order that the rows add up to one. It must also be the case that δt is sufficiently small that $\forall i, j : p_{ij}\delta t < 1$. The continuous time Markov chain could be thought of as the limit of this process as $\delta t \to 0$. However, the matrix defined above would simply become the identity matrix. A new approach is needed.

Define the following (assuming that the states of the chain are numbered $(0, 1, 2, \dots)$).

- $X(t)$ is the state of the chain at some time $t \geq 0$.
- $\mathbf{f}(t) = (f_0(t), f_1(t), \dots)$ is the vector of probabilities at time t, formally $f_i = \mathbb{P}[X(t) = i]$.
- $q_{ij}(t_1, t_2)$ where $t_1 < t_2$ is $\mathbb{P}[X(t_2) = j | X(t_1) = i]$.

Since the context is still homogeneous chains then these probabilities are just a function of $\tau = t_2 - t_1$. Hence, define for $i \neq j$

$$q_{ij}(\tau) = q_{ij}(t_2 - t_1) = q_{ij}(t_1, t_2) = \mathbb{P}[X(\tau) = j | X(0) = i].$$

Define the limit

$$q_{ij} = \lim_{\tau \to 0} \frac{q_{ij}(\tau)}{\tau}.$$

Consider the transition rates between states i and j which were defined as p_{ij} (where $i \neq j$). As the process is like a Poisson then for a given state i and a small time period δt

$$f_i(t + \delta t) = f_i(t) - \sum_{j \neq i} f_i(t) p_{ij} \delta t + \sum_{j \neq i} f_j(t) p_{ji} \delta t + o(\delta t).$$

Taking the difference, dividing through by δt and taking the limit as $\delta t \to 0$ (a differential difference equation again) then

$$\frac{df_i(t)}{dt} = -\sum_{j \neq i} f_i(t) p_{ij} + \sum_{j \neq i} f_j(t) p_{ji}.$$

Similarly, this could be derived in terms of $q_{ij}(t)$ so

$$f_i(t + \delta t) = f_i(t) - \sum_{j \neq i} f_i(t) q_{ij}(\delta t) + \sum_{j \neq i} f_j(t) q_{ji}(\delta t).$$

As before this becomes

$$\frac{df_i(t)}{dt} = -\sum_{j \neq i} f_i(t) q_{ij} + \sum_{j \neq i} f_j(t) q_{ji}.$$

Therefore, for $i \neq j$ it is the case that $q_{ij} = p_{ij}$. Now, q_{ii} still needs definition. It is handy to define

$$q_{ii} = -\sum_{i \neq j} q_{ij}.$$

Now, define the matrix

$$\mathbf{Q} = \begin{bmatrix} q_{00} & q_{01} & q_{02} & \cdots \\ q_{10} & q_{11} & q_{12} & \cdots \\ q_{20} & q_{21} & q_{22} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

It can now be seen that

$$\frac{d\mathbf{f}(t)}{dt} = \mathbf{f}(t)\mathbf{Q}.$$

This is similar to the equation for the discrete time homogenous Markov chain but the rows of $\mathbf{Q}$ add up to zero not one. Notice that

$$\mathbf{Q} = \mathbf{P}(1) - \mathbf{I},$$

where $\mathbf{I}$ is the identity matrix.

Assume the chain is finite and there are no disconnected states. Now the equilibrium probabilities can be calculated. In this case

$$\boldsymbol{\pi} = \lim_{t \to \infty} \mathbf{f}(t).$$

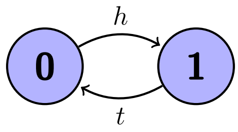


Figure 4: Talking on the phone continuous time Markov chain

Therefore
$$\boldsymbol{\pi}\mathbf{Q} = \frac{d \lim_{t\to\infty} \mathbf{f}(t)}{dt} = \frac{d}{dt} \lim_{t\to\infty} \mathbf{f}(t) = \frac{d\boldsymbol{\pi}}{dt} = 0,$$
where the equality to zero comes from the fact that by definition $\boldsymbol{\pi}$ is constant. The equation $\boldsymbol{\pi}\mathbf{Q} = 0$ is the continuous Markov chain equivalent of $\boldsymbol{\pi}\mathbf{P} = \boldsymbol{\pi}$ for discrete chains.

This gives a new version of our balance equations. For all i then
$$\sum_j \pi_j q_{ji} = 0$$

Expanding q_{jj} from its definition and multiplying by -1 gives
$$\sum_{j\neq i} \pi_i q_{ij} - \sum_{j \neq i} \pi_j q_{ji} = 0.$$

This can also be seen as a balance of flows into and out of the state. Also as usual $\sum_i \pi_i = 1$.

4.1 The "talking on the phone" example

If I am talking on the phone, I will hang up as a Poisson process with a rate h (for hang up). If I am not talking on the phone, I will decide to start a new call as a Poisson process with rate t (for talk). At a given time what is the probability I am talking on the phone? Unsurprisingly this can be modelled as a Markov chain. This example may seem "trivial" but several such chains could be use to model how occupied the phone network is.

Our chain has two states 0 (talking) and 1 (not talking) and the transition matrix:
$$\mathbf{Q} = \begin{bmatrix} -h & h \\ t & -t \end{bmatrix}.$$

- State 0 – (output) $h\pi_0 = t\pi_1$ (input)
- State 1 – (output) $t\pi_1 = h\pi_0$ (input)

We also need $\pi_0 + \pi_1 = 1$ which gives from state 0 $h\pi_0 = t(1 - \pi_0)$. Rearrange to $\pi_0 = t/(h+t)$ and $\pi_1 = h/(h+t)$. Interpretation – the proportion of time talking (state 0) is proportional to the talking rate t and the proportion hung up is proportional to the hangup rate.

5 Queuing Theory

5.1 Little's Theorem

More information can be found in the book, Bertsekas & Gallagher, section 3.2 (note that there are some minor errors and notational inconsistencies in their version). Little's Theorem is, in some ways, obvious. First we will introduce the theorem in a handwaving manner and then make the details of the definition more precise. Little's Theorem states:
$$N = \lambda T, \tag{1}$$
where N is the average number of customers in a queue, T is the average time a customer spends queuing and λ is the average rate of arrivals to the queue. In many ways this theorem represents an obvious truth — if a lot of people are in a queue (N is large) then they will have long delays (T is large); if few people arrive

in a queue (λ is small) then the average number of people in the queue is small (N is small). This lecture will be spent making this intuitive idea more rigorous.

Let us first make precise the definitions of N, λ and T and then make clear the assumptions on which the theorem rests. Let us first make some definitions:

- $N(\tau)$ is the number of customers in the system at time τ.
- $\alpha(\tau)$ is the number of customers who arrived in the interval $[0, \tau]$.
- $\beta(\tau)$ is the number of customers who have departed in the interval $[0, \tau]$.
- t_i is the time at which the ith customer arrived.
- $T(i)$ is the time spent queuing by the ith customer.

If N_t is the mean value of $N(\tau)$ taken over the interval $[0, t]$ then it is clear that:

$$N_t = \frac{1}{t} \int_0^t N(\tau) d\tau \tag{2}$$

Let us assume:

$$N = \lim_{t \to \infty} N_t \tag{3}$$

(Note that this limit is *not* guaranteed to exist — imagine, for example, a queue which keeps growing.) If the limit exists, N is the *steady state time average* of $N(\tau)$. We next define the average arrival rate over the time period $[0, t]$.

$$\lambda_t = \frac{\alpha(t)}{t} \tag{4}$$

and, again, we assume that the following limit exists:

$$\lambda = \lim_{t \to \infty} \lambda_t \tag{5}$$

Finally, the average delay experienced by those customers who enter the system at times in $[0, t]$ is given by:

$$T_t = \sum_{i=1}^{\alpha(t)} \frac{T(i)}{\alpha(t)} \tag{6}$$

And, for a third time, we assume that the following limit exists:

$$T = \lim_{t \to \infty} T_t. \tag{7}$$

A partial proof is given in section 5.2

5.2 Aside: proof of Little's theorem

This section is not part of the main lecture but proves Little's Theorem with the restrictive conditions that, in addition to the limits above existing, the queue is empty at time 0 ($N(0) = 0$), that queuing is First in First Out (FIFO) and that the queue becomes empty infinitely often beyond any given time τ. Figure 5 shows a FIFO queue which is initially empty.

We note that at any time τ:

$$N(\tau) = \alpha(\tau) - \beta(\tau). \tag{8}$$

It is clear that if we choose a time t when the system again becomes empty then we can calculate the area of the shaded area $A(t)$:

$$A(t) = \int_0^t N(\tau) d\tau \tag{9}$$

However, equally, we can consider the shaded area to be composed of horizontal strips of height 1 and width $T(i)$ (for the ith customer). In this case, we have:

$$A(t) = \sum_{i=1}^{\alpha(t)} T(i) \tag{10}$$

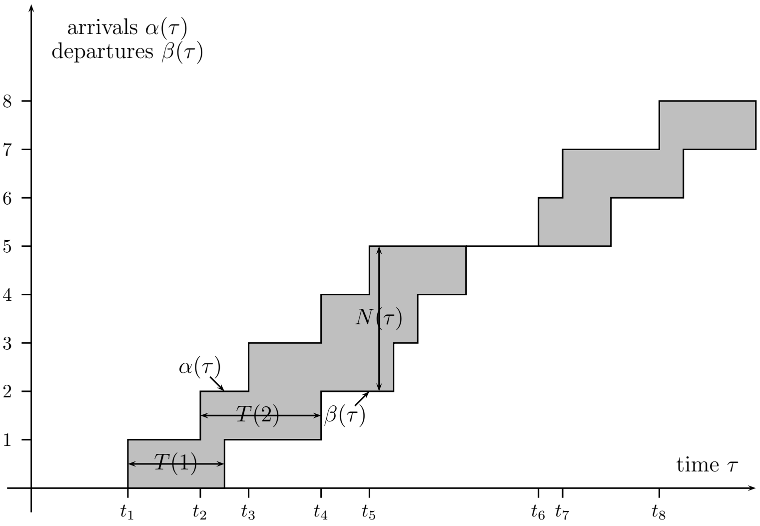


Figure 5: Little's Theorem in a FIFO System

Setting these equations equal and dividing each side by t gives us:

$$\frac{1}{t}\int_0^t N(\tau)d\tau = \frac{1}{t}\sum_{i=1}^{\alpha(t)} T(i) = \frac{\alpha(t)}{t}\frac{\sum_{i=1}^{\alpha(t)} T(i)}{\alpha(t)} \tag{11}$$

Therefore we have:

$$N_t = \lambda_t T_t, \tag{12}$$

which, if we take the limit as $(t \to \infty)$ becomes Little's Theorem as required. Note that some of the assumptions made here are, in fact, unnecessary as we shall see. In fact, Little's Theorem only requires the following:

1. The limit $\lambda = \lim_{t\to\infty} \alpha(t)/t$ exists
2. The limit $\delta = \lim_{t\to\infty} \beta(t)/t$ exists
3. The limit $T = \lim_{t\to\infty} T_t$ exists
4. $\delta = \lambda$

5.3 Little's Theorem example

Here is an example where Little's Theorem could be useful:

- You are building a website and want to know how big a server you need.
- You believe your website will attract 24,000 visitors a day – 1,000 visitors an hour.
- You believe the average visitor will spend 6 minutes on the website.
- How many visitors does your server need to cope with?
- λ = 1,000 per hour, $T = 0.1$ hours.
- From $N = \lambda T$, $N = 100$, the average number of visitors at a time is 100.
- But because arrival is in "peaks" better plan for a peak hour.

5.4 Queuing theory notation

Queuing theory uses a particular notation (Kendall's notation) to describe a queuing system. The *arrival process* describes the distribution of the interarrival times.

- M – memoryless (Exponential) – a Poisson process.
- D – deterministic – equally spaced.
- G – general (no specific distribution).
- Also Ph (phase), EK (Erlangian)

The *service time* distribution (from same choice of letters) determines how long it will take to process an item in the queue. The *number of servers* describes how many servers deal with the queue. For example $M/D/1$ is a Poisson input to a single queue which processes in constant time.

5.5 The birth-death process

A birth-death process is a process where the population k may increase or decrease according to certain rules. Specifically, when the population is k then the population may increase to $k+1$ in the manner of a Poisson process with rate λ_k and may decrease to $k-1$ in the manner of a Poisson process with rate μ_k. The parameter λ_k is known as the birth rate for population k and the parameter μ_k is known as the death rate for population k. It is usually assumed that $\mu_0 = 0$ (if there is no population then nobody can die and that k begins at 0 (or some positive integer.

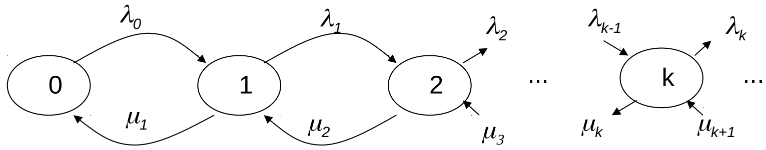


Figure 6: A birth-death process as a Markov chain

Figure 6 shows the birth-death process as a Markov Chain. This gives the transition matrix:

$$\mathbf{Q} = \begin{bmatrix} \lambda_0 & \lambda_0 & 0 & 0 & \cdots \\ \mu_1 & (\lambda_1+\mu_1) & \lambda_1 & 0 & \cdots \\ 0 & \mu_2 & (\lambda_2+\mu_2) & \lambda_2 & \cdots \\ 0 & 0 & \mu_3 & (\lambda_3+\mu_3) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

The balance equation for π_0 is

$$\mu_1\pi_1 = \lambda_0\pi_0$$

and for π_k with $k > 0$ is

$$lambda_{k-1}\pi_{k-1} + \mu_{k+1}\pi_{k+1} = (\lambda_k + \mu_k)\pi_k.$$

From the first equation we can get

$$\pi_1 = \lambda_0\pi_0/\mu_1.$$

From the equation for $k = 1$ we can get

$$\lambda_0\pi_0 + \mu_2\pi_2 = \lambda_1\pi_1 + \mu_1\pi_1,$$

Substituting the equation for π_1 and rearranging gives

$$\pi_2 = \frac{\lambda_1\lambda_0\pi_0}{\mu_2\mu_1}.$$

Similarly

$$\pi_3 = \frac{\lambda_2\lambda_1\lambda_0\pi_0}{\mu_3\mu_2\mu_1}.$$

This leads to the suspicion that the general form is

$$\pi_k = \pi_0 \prod_{i=1}^{k} \frac{\lambda_{i-1}}{\mu_i}. \tag{13}$$

Equation (13) can obtain an expression for any π_k in terms of π_0 but it is still necessary to find an expression for π_0. This can be done using the fact that the sum of all the π_k must be 1. Therefore

$$\begin{aligned}\pi_0 &= 1 - \sum_{k=1}^{\infty} \pi_k \\ &= 1 - \sum_{k=1}^{\infty} \pi_0 \prod_{i=1}^{k} \frac{\lambda_{i-1}}{\mu_i}.\end{aligned}$$

This can be rearranged to give

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{\infty} \prod_{i=1}^{k} \frac{\lambda_{i-1}}{\mu_i}}. \tag{14}$$

These two equations (13) and (14) can be used to find the equilibrium probabilities for any birth-death process providing the product can be evaluated. To do this it is valuable to introduce the concept of utilisation.

5.6 Utilisation

Utilisation is defined as the proportion of the queues capacity which is being used. However, this definition is somewhat vague. The utilisation, ρ is given by the equation

$$\rho = \frac{\lambda}{\mu},$$

where λ is the mean arrival rate for the system and μ is the maximum service rate. A clear definition of μ is hard to obtain. In cases where there is some N such that $\mu_i = C$ for all $i > N$ it is clear that $\mu = C$ (this is the case for the M/M/1 model). In cases where the birth death process has some maximum population K and $\mu_i \geq \mu_{i-1}$ for $0 < i \leq K$ then $\mu = \mu_K$. In cases where μ_i increases and does not reach a limit then μ is infinite. Fortunately, in most of the cases dealt with in this course the value to use will be clear. In the $M/M/1$ system $\rho = \lambda/\mu$.

The point where $\rho = 1$ is a critical point for a queuing system as this is the point where all the traffic that could possibly be handled is arriving. As will be seen, this is a point where the system breaks down. If $\rho \geq 1$ then an infinite birth-death process "goes to infinity".

5.7 Finally, the M/M/1 model solved

Getting a full solution to the M/M/1 model (including transients) is difficult but the equilibrium solution is easy to find. For the M/M/1 process the arrival process is a constant Poisson process with a rate λ and the server process is a Poisson process with rate μ. Substituting into the birth death equations gives

$$\pi_k = \prod_{i=1}^{k} \frac{\lambda_{i-1}}{\mu_i} \pi_0 = \rho^k \pi_0$$

for $k \in (1, 2, \ldots)$ with $\rho = \lambda/\mu$ as the utilisation. Using the second equation for birth death processes gives

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{\infty} \rho^k} = \frac{1}{1 + \rho/(1-\rho)} = 1 - \rho.$$

This can be though of as the probability that the queue is empty and as can be seen this falls linearly from 1 (when the system utilisation is zero, that is no arrivals occur and the system is always empty) to 0 (when the system is at full utilisation and the queue never falls to zero).

The obvious final step in the solution is to calculate the expected queue length when the system is in equilibrium. Call this $\mathrm{E}[q]$, it is given by

$$\mathrm{E}[q] = \sum_{i=0}^{\infty} i\pi_i = \sum_{i=0}^{\infty} i(1-\rho)\rho^i.$$

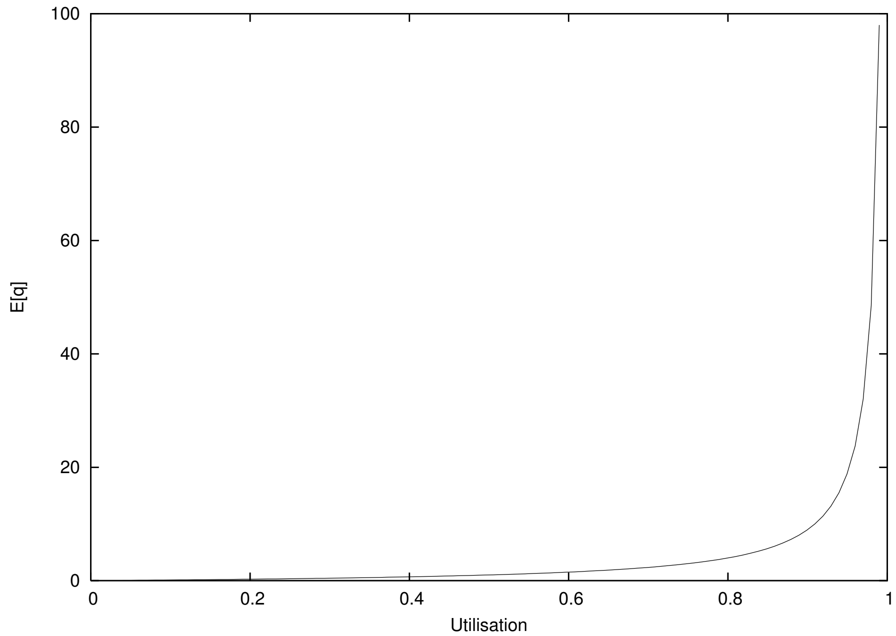


Figure 7: Utilisation versus expected queue size for an M/M/1 queue.

A nice trick can be used here. First rewrite as

$$\begin{aligned}\sum_{i=0}^{\infty} i(1-\rho)\rho^i &= (1-\rho)\rho\sum_{i=0}^{\infty} i\rho^{i-1}\\ &= (1-\rho)\rho\sum_{i=0}^{\infty}\frac{d\rho^i}{d\rho}\\ &= (1-\rho)\rho\frac{d}{d\rho}\sum_{i=0}^{\infty}\rho^i\\ &= (1-\rho)\rho\frac{d}{d\rho}\frac{1}{1-\rho}\\ &= (1-\rho)\rho\frac{1}{(1-\rho)^2}\\ &= \frac{\rho}{1-\rho}.\end{aligned}$$

The expected queue length begins at 0 and heads to infinity as ρ approaches one. (Remember that this solution is only valid for an ergodic chain with $\rho < 1$.)

Little's Theorem will give the average delay as

$$T = \frac{\rho}{\lambda(1-\rho)} = \frac{1/\mu}{1-\rho} = \frac{1}{\mu-\lambda}.$$

While this lecture stops here, the birth-death process model can easily be extended for other queues. For example the M/M/k queue has Exponential arrivals, Exponential service time and k servers. This means that the birth rate is λ for all states but the death rate increases up to $k\mu$ for state k as more servers are added.

6 Network research summary

6.1 Real life network measurements

We learned how to completely model a queue with Poisson process input and Poisson service time. Is modelling the internet solved? Think back to the start of the lecture we need to think about the input process, the server

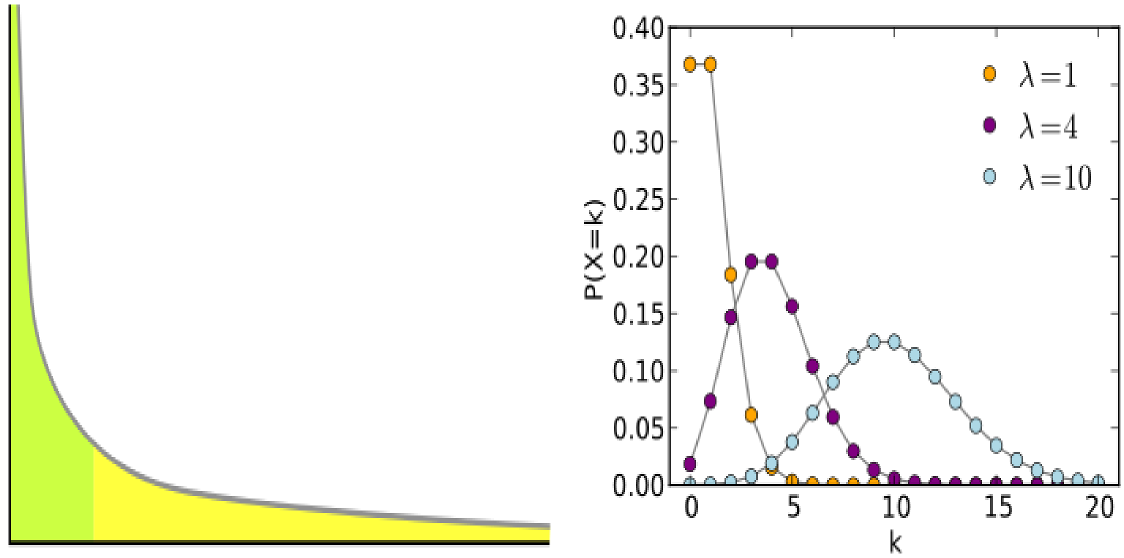


Figure 8: Heavy tail distribution (left) versus Poisson (right)

process and the network topology. In fact there is a rather beautiful queuing theory result about networks with all Poisson input (and some other conditions) – the Jackson Network. But is the assumption of a Poisson input realistic? What is the server process?

A variable X has a heavy-tailed distribution with $\beta \in (0, 2)$ if

$$\mathbb{P}\left[X > x\right] \sim x^{-\beta},$$

where $\sim$ means asymptotically proportional to as $x \to \infty$. See figure 8. This is related to power laws. Here there are many many small values but also *extreme values* are still quite common. Examples in real life include: Heights of trees, frequency of words, populations of towns.

The following internet distributions have (approximate) heavy tails:

1. Files on any particular computer.
2. Files transferred via ftp.
3. Bytes transferred by single TCP connections.
4. Files downloaded by the WWW.

This is more than just a statistical curiosity. Consider what this distribution would do to queuing performance (no longer Poisson). Non mathematicians are starting to take an interest in heavy tails (reference to "the long tail". This can be useful in sales for example (small selling items have a lot of sales if you group them together). *Why do we care?* This drastically impacts performance of queues, loading of servers, etc etc.

The node-degree distribution in AS networks is particularly well-studied. Let $P(k)$ be the proportion of nodes with degree k (having k neighbours). To a good approximation

$$P(k) \sim k^{-\alpha},$$

where α is a constant and $\sim$ is asymptotically proportional to.

Power law topology of the AS graph was shown by [Faloutsos x3]. This graph has some interesting properties — some extremely highly connected nodes, what happens if they fail? Same type of graph as:

1. Links on websites, Wikipedia and many other similar online systems.
2. Academic citations in papers.
3. Human frientship, sexual contacts.

How do we generate a scale-free network? One option is the Albert–Barabasi [Barabasi 99] "Preferential attachment" model

Start with a small "core" network. When a new node arrives, attach it to an old node with the following probability

$$\mathbb{P}\left[\text{Attaching to node } i\right] = \frac{d(i)}{\sum_{j \in \text{all nodes}} d(j)},$$

where $d(i)$ is the degree of node i.

This model "grows" a network with a powerlaw. Many similar models have been created which are more general. *Why do we care?* Structure of network tells us about resilience, number of hops to connect and how we should model large scale internet.

More recently researchers have questioned how exact these power law fits are. All of this is still a rapidly evolving reseearch area. New results are changing our ideas of what is the most accurate model. This is a very active research area – new publications and new results every month. What is the "true" model for internet traffic and topology? Nobody knows – I and others are trying to find out.

6.2 Software Defined Networks and OpenFlow

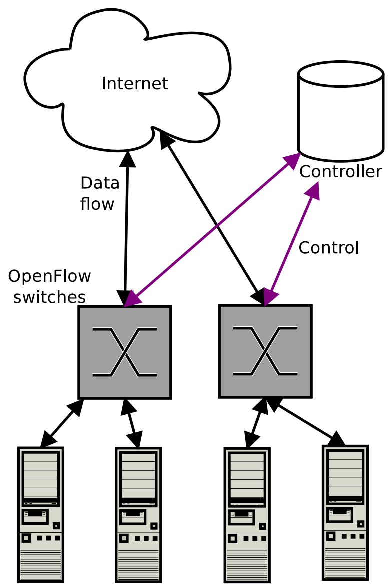
Software Defined Networks: Allows more flexible use of the control plane by separating it from the data plane. Programmable control of your network.

OpenFlow: A limited form of SDN which allows programmatic direction of data based on layer-2, layer-3 and layer-4 headers.

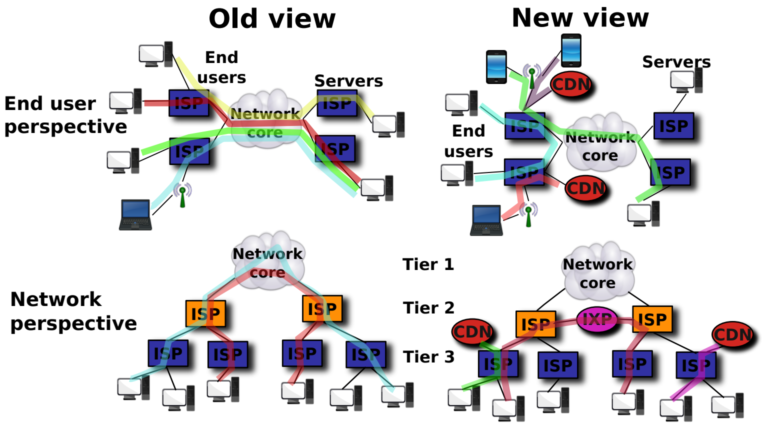
Controller contains list of matching rules and actions The first packet of every non matched flow is sent to the controller. Controller uses standard coding languages (python, Java, C++) to decide how to deal with packet and to insert rules.

Why do we care?

Can test new ideas on our own computers as if it were a real network. Can deploy new ideas on a testbed with relative ease. Don't need to learn how to wrangle each new router. Flexible and interchangeable routers.



6.3 Flattening of the network



Old: A hierarchical network with content far from the user.

New: A flatter network with content nearer to the user.

Content pushed towards the edges
Old: Client gets content from server
New: Content pushed to content distribution networks (CDNs)

Network topology flattens
Old: Heirarchical network, eyeball ISPs connect to transit providers then tier ones.
New: Internet eXchange Points (IXPs) connect ISPs directly

Why do we care? Structure of network is changing. Content is closer to the user.

6.4 Information Centric Networking

How do we connect to a web address?

Put URL in browser `http://www.richardclegg.org/erdos`, translate first part to IP address `89.16.173.124`, connect to that machine, ask it for that data labelled `erdos`.

What's wrong with this? We don't want to connect to that machine, we want to retrieve that content (web page).

How else could we do it? Address maps to content, not machine and does so in a cryptographically signed way. Provably get the data you want from a nearby host.

Why do we care? Interesting new trend. If deployed, will radically change how networks operate.

7 Tools

Much software is available to help you model networks. Here are some selected examples.

7.1 The ns-3 simulation

- ns-3 is a freely available event-driven simulator which simulates packet-level traffic.
- It is available from `http://nsnam.isi.edu/nsnam`
- It does allow you to connect real traffic sources into your simulation.
- It has many prebuilt modelling scenarios which may be useful for the modelling you wish to do.
- The scripts used for these examples are available at `http://www.richardclegg.org/modelling_networks`
- It is available from `https://www.nsnam.org/`

7.2 OpenFlow and mininet

Mininet is a tool which can be used for modelling network topologies. Real network connections are made on a single machine. We can use this in combination with the Floodlight OpenFlow controller to demonstrate OpenFlow networking.

7.3 Which tool

ns-3 is useful for large simulations and ns-3 is useful if you need real sources of traffic to interact with your simulation. Mininet is useful for smaller systems where the whole network can be emulated.

8 Final thoughts

- Select an appropriate level of modelling — if you need to model the whole Internet you cannot do packet level modelling. If you need to model intricate protocol details for packets you cannot model the whole Internet.
- Check against real data where possible that your modelling assumptions are justified.
- Is your experiment repeatable? Do you get similar results if you try slightly different starting scenarios?
- Remember sensitivity analysis: What happens if the bandwidth is a little less? What if the demand is a little more?
- Can statistical analysis of your results help?
- Don't reinvent the wheel – what is out there to help you already?

9 Bibliography

References

[1] R. J. Adler, R. E. Feldman, and M. S. Taqqu, editors. *A Practical Guide to Heavy Tails.* Birkhäuser, 1998.

[2] A. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286:509, 1999.

[3] D. P. Bertsekas, and R. G. Gallagher Data Networks *Longman Higher Education* 1986

[4] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power–law relationships of the Internet topology. *Comput. Commun. Rev.*, 29:251–262, 1999.

[5] S. Floyd and V. Paxson. Difficulties in simulating the Internet. *IEEE/ACM Trans. on Networking*, 9(4):392–403, 2001. `http://www.icir.org/floyd/papers/simulate_2001.pdf`

[6] S. Zhou and R. J. Mondragón. Accurately modelling the Internet topology. *Phys. Rev. E*, 70(066108), 2004.