Introduction to queuing theory

Queu(e)ing theory

Queu(e)ing theory is the branch of mathematics devoted to how objects (packets in a network, people in a bank, processes in a CPU etc etc) join and leave queues.

- Queuing is the traditional British spelling but now queueing is probably more common.
- The first papers about queuing theory were published by Erlang who was studying the Danish telephone system.
- Queuing theory involves the study of Markov chains.
- Strictly we would use continuous time Markov chains but here we will approximate with discrete time Markov chains.

Little's theorem

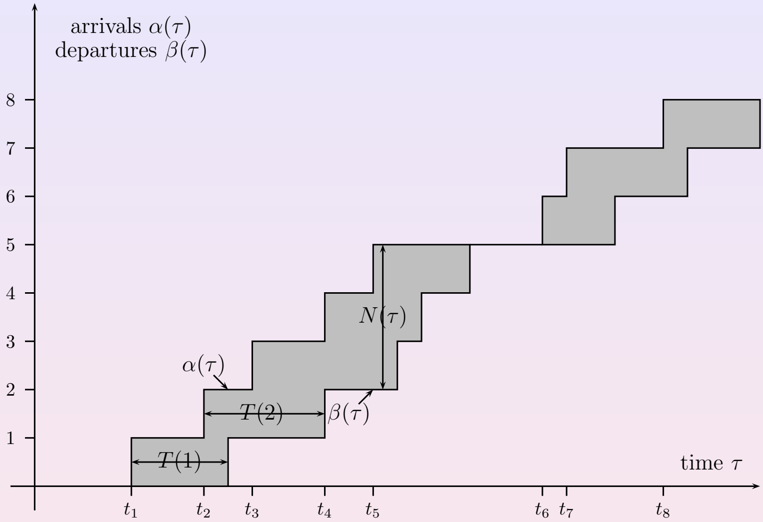
Little's theorem

Let N be the average number of customers in a queue. Let λ be the average rate of arrivals. Let T be the average time spent queuing. Then we have

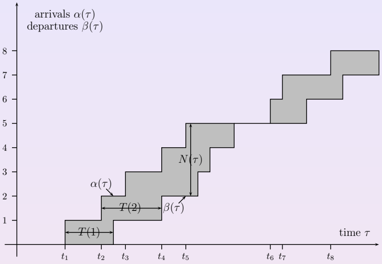
$$N = \lambda T.$$

- In fact this simple theorem hides much complexity.
- It is only true under certain conditions.

Little's theorem illustration



Little's theorem requirements



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 $T = \lim_{t\to\infty} \sum_{i=1}^{\alpha(t)} \frac{T(i)}{\alpha(t)}$ exists
4. $\delta = \lambda$

Little's theorem example

- 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?

Little's theorem example

- 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?
- $\lambda = 1{,}000$ per hour, $T = 0.1$ hours.

Little's theorem example

- 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?
- $\lambda = 1{,}000$ per hour, $T = 0.1$ hours.
- From $N = \lambda T$, $N = 100$, the average number of visitors at a time is 100.

Little's theorem example

- 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?
- $\lambda = 1{,}000$ 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.

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 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.

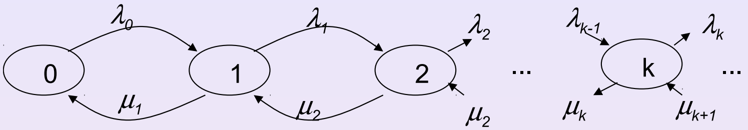
The Birth-Death process

The Birth–Death process

The birth–death process is a queue with a population which increases or decreases with rates which depend only on k the population at the time. Many queues can be modelled this way.

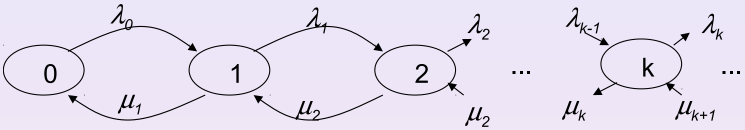
- Think of it as a queue – state 0 has no people. Arrivals are a Poisson process, rate λ_0.
- State k has births (arrivals) at rate λ_k but deaths (departures) at rate μ_k.

Starting the Birth–Death process



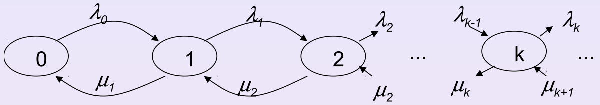
- Here we can see the arrivals and departures as a Markov chain.
- The state represents the number of people in the queue.
- An M/M/1 system would be modelled by $\lambda_k = \lambda$ for all k and $\mu_k = \mu$ for all k.
- Strictly speaking we should model this as a continuous time Markov chain.
- Here we pretend that μ_k and λ_k are the arrival probabilities in some small δt so small that $1 - \mu_k - \lambda_k > 0$.

Birth–Death process – Transition matrix



$$
\mathbf{P} = \begin{bmatrix} 1-\lambda_0 & \lambda_0 & 0 & 0 & \cdots \\ \mu_1 & 1-(\lambda_1+\mu_1) & \lambda_1 & 0 & \cdots \\ 0 & \mu_2 & 1-(\lambda_2+\mu_2) & \lambda_2 & \cdots \\ 0 & 0 & \mu_3 & 1-(\lambda_3+\mu_3) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}
$$

Birth–Death process – Balance equations



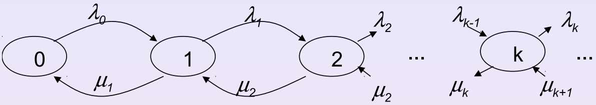
Balance equation for state 0

$\pi_0 = \mu_1 \pi_1 + (1 - \lambda_0)\pi_0$

and for state k with $k > 0$

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

Birth–Death process – Balance equations



Balance equation for state 0

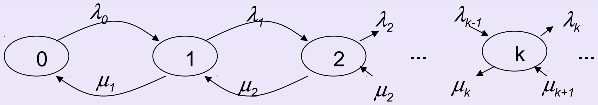
$\pi_0 = \mu_1\pi_1 + (1 - \lambda_0)\pi_0$

and for state k with $k > 0$

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

Rearrange to get: $\pi_1 = \lambda_0\pi_0/\mu_1.$

Birth–Death process – Balance equations

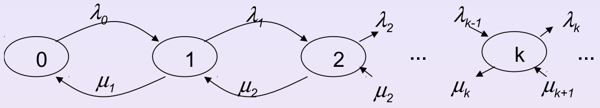


Balance equation for state 0
$\pi_0 = \mu_1 \pi_1 + (1 - \lambda_0)\pi_0$
and for state k with $k > 0$
$\pi_k = \lambda_{k-1}\pi_{k-1} + \mu_{k+1}\pi_{k+1} + (1 - \lambda_k - \mu_k)\pi_k.$

Rearrange to get: $\pi_1 = \lambda_0 \pi_0 / \mu_1$.

For state 1 $\lambda_0\pi_0 + \mu_2\pi_2 = \lambda_1\pi_1 + \mu_1\pi_1$,

Birth–Death process – Balance equations



Balance equation for state 0

$\pi_0 = \mu_1\pi_1 + (1 - \lambda_0)\pi_0$

and for state k with $k > 0$

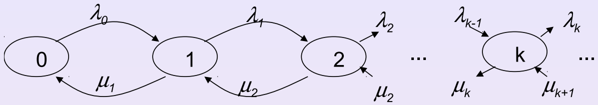
$\pi_k = \lambda_{k-1}\pi_{k-1} + \mu_{k+1}\pi_{k+1} + (1 - \lambda_k - \mu_k)\pi_k.$

Rearrange to get: $\pi_1 = \lambda_0\pi_0/\mu_1.$

For state 1 $\lambda_0\pi_0 + \mu_2\pi_2 = \lambda_1\pi_1 + \mu_1\pi_1$,

Rearrange to get: $\pi_2 = \frac{\lambda_1\lambda_0\pi_0}{\mu_2\mu_1}.$

Birth–Death process – Balance equations

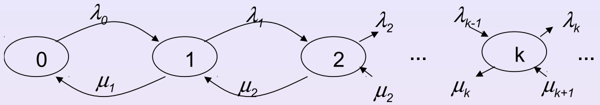


We have:

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

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

Birth–Death process – Balance equations



We have:

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

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

Can show that in general:

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

Birth–Death process – Balance equations

- Given $\pi_k = \pi_0 \prod_{i=1}^{k} \frac{\lambda_{i-1}}{\mu_i}$. now solve with $\sum_k \pi_k = 1$.
- This is complicated, the full solution is given in the notes.

Birth–Death process – Balance equations

- Given $\pi_k = \pi_0 \prod_{i=1}^{k} \frac{\lambda_{i-1}}{\mu_i}$. now solve with $\sum_k \pi_k = 1$.
- This is complicated, the full solution is given in the notes.
- $$\pi_0 = \frac{1}{1 + \sum_{k=1}^{\infty} \prod_{i=1}^{k} \frac{\lambda_{i-1}}{\mu_i}}.$$

Birth–Death process – Balance equations

- Given $\pi_k = \pi_0 \prod_{i=1}^{k} \frac{\lambda_{i-1}}{\mu_i}$. now solve with $\sum_k \pi_k = 1$.
- This is complicated, the full solution is given in the notes.
- $$\pi_0 = \frac{1}{1 + \sum_{k=1}^{\infty} \prod_{i=1}^{k} \frac{\lambda_{i-1}}{\mu_i}}.$$
- This may not seem to help much – but we also have an equation for π_k in terms of π_0.
- Given μ_k and λ_k all the π_k can be worked out and hence the average queue length.
- To get further and finally solve M/M/1 we need the concept of utilisation.

Utilisation

Utilisation

Utilisation (utilization if you are American) ρ is given by the equation

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

where λ is the mean arrival rate and μ is the maximum possible service rate of the system (when all servers are working).

- Utilisation is a good measure of the "fullness" of the system.
- A system at low utilisation is likely to be empty much of the time.
- Utilisation can also be thought of as the proportion of time the system is "busy".

Solving the M/M/1

- Finally we are ready to solve M/M/1
- Substituting in $\pi_k = \prod_{i=1}^{k} \frac{\lambda_{i-1}}{\mu_i} \pi_0 = \rho^k \pi_0$.

Solving the M/M/1

- Finally we are ready to solve M/M/1
- Substituting in $\pi_k = \prod_{i=1}^{k} \frac{\lambda_{i-1}}{\mu_i} \pi_0 = \rho^k \pi_0$.
- Also $\pi_0 = \frac{1}{1+\sum_{k=1}^{\infty} \rho^k} = \frac{1}{1+\rho/(1-\rho)} = 1 - \rho$.

Solving the M/M/1

- Finally we are ready to solve M/M/1
- Substituting in $\pi_k = \prod_{i=1}^{k} \frac{\lambda_{i-1}}{\mu_i} \pi_0 = \rho^k \pi_0$.
- Also $\pi_0 = \frac{1}{1+\sum_{k=1}^{\infty} \rho^k} = \frac{1}{1+\rho/(1-\rho)} = 1 - \rho$.
- Hence for M/M/1 $\pi_k = \rho^k (1-\rho)$.
- The mean queue length is $E[Q] = \sum_{k=1}^{\infty} k\pi_k = \sum_{k=1}^{\infty} \rho^k (1-\rho)$.
- A neat trick gives us the answer.

A neat trick to solve M/M/1 expected queue

$E[Q] = \sum_{i=0}^{\infty} i(1-\rho)\rho^i$

A neat trick to solve M/M/1 expected queue

$$E[Q] = \sum_{i=0}^{\infty} i(1-\rho)\rho^i$$

$$E[Q] = (1-\rho)\rho \sum_{i=0}^{\infty} i\rho^{i-1}$$

A neat trick to solve M/M/1 expected queue

$$E[Q] = \sum_{i=0}^{\infty} i(1-\rho)\rho^i$$

$$E[Q] = (1-\rho)\rho \sum_{i=0}^{\infty} i\rho^{i-1}$$

$$E[Q] = (1-\rho)\rho \sum_{i=0}^{\infty} \frac{d\rho^i}{d\rho}$$

A neat trick to solve M/M/1 expected queue

$$E[Q] = \sum_{i=0}^{\infty} i(1-\rho)\rho^i$$

$$E[Q] = (1-\rho)\rho \sum_{i=0}^{\infty} i\rho^{i-1}$$

$$E[Q] = (1-\rho)\rho \sum_{i=0}^{\infty} \frac{d\rho^i}{d\rho}$$

$$E[Q] = (1-\rho)\rho \frac{d}{d\rho} \sum_{i=0}^{\infty} \rho^i$$

A neat trick to solve M/M/1 expected queue

$$E[Q] = \sum_{i=0}^{\infty} i(1-\rho)\rho^i$$

$$E[Q] = (1-\rho)\rho \sum_{i=0}^{\infty} i\rho^{i-1}$$

$$E[Q] = (1-\rho)\rho \sum_{i=0}^{\infty} \frac{d\rho^i}{d\rho}$$

$$E[Q] = (1-\rho)\rho \frac{d}{d\rho} \sum_{i=0}^{\infty} \rho^i$$

$$E[Q] = (1-\rho)\rho \frac{d}{d\rho} \frac{1}{1-\rho}$$

A neat trick to solve M/M/1 expected queue

$$E[Q] = \sum_{i=0}^{\infty} i(1-\rho)\rho^i$$

$$E[Q] = (1-\rho)\rho \sum_{i=0}^{\infty} i\rho^{i-1}$$

$$E[Q] = (1-\rho)\rho \sum_{i=0}^{\infty} \frac{d\rho^i}{d\rho}$$

$$E[Q] = (1-\rho)\rho \frac{d}{d\rho} \sum_{i=0}^{\infty} \rho^i$$

$$E[Q] = (1-\rho)\rho \frac{d}{d\rho} \frac{1}{1-\rho}$$

$$E[Q] = (1-\rho)\rho \frac{1}{(1-\rho)^2}$$

A neat trick to solve M/M/1 expected queue

$$E[Q] = \sum_{i=0}^{\infty} i(1-\rho)\rho^i$$

$$E[Q] = (1-\rho)\rho \sum_{i=0}^{\infty} i\rho^{i-1}$$

$$E[Q] = (1-\rho)\rho \sum_{i=0}^{\infty} \frac{d\rho^i}{d\rho}$$

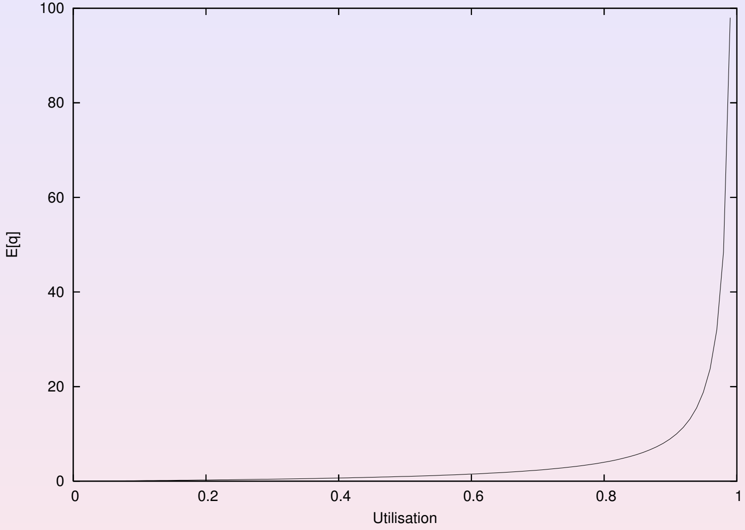
$$E[Q] = (1-\rho)\rho \frac{d}{d\rho} \sum_{i=0}^{\infty} \rho^i$$

$$E[Q] = (1-\rho)\rho \frac{d}{d\rho} \frac{1}{1-\rho}$$

$$E[Q] = (1-\rho)\rho \frac{1}{(1-\rho)^2}$$

$E[Q] = \frac{\rho}{1-\rho}$. At last! the solution for M/M/1.

Queue size versus utilisation for M/M/1



Queuing theory summary

- For the M/M/1 queue we have $E[Q] = \rho/(1-\rho)$.
- As the utilisation goes to 1 the queue length goes to infinity.
- The mean waiting time can be found from Little's theorem $N = E[Q]$.
- Therefore $T = \frac{\rho}{\lambda(1-\rho)} = \frac{1/\mu}{1-\rho} = \frac{1}{\mu-\lambda}$.
- If we wanted an M/M/k queue (k servers) this is $\lambda_i = \lambda$ for all i and $\mu_i = i\mu$ for $i < k$ and $\mu_k = k\mu$.
- If we wanted an M/M/k/l queue (maximum l people in the queue) we would say $\lambda_k = 0$ for $k \geq l$.
- So it can be seen that we have solved a lot more in this lecture than simply M/M/1

Queuing theory summary

- This lecture can only scratch the surface of queuing theory.
- Little's Theorem relates queue size, arrivals and mean queuing time $N = \lambda T$.
- The birth-death process is a very general way to look at queues of arrivals where arrivals and departures are related to Poisson processes.
- The birth-death process can be completely solved and the probability of every queue length calculated in terms of λ_k and μ_k.
- Utilisation is a measure of the fullness of the system $\rho = \lambda/\mu$.
- M/M/1, M/M/k and M/M/k/l queues can be solved as birth-death processes.