Lecture 8

In this lecture we will reuse our work on Birth-Death processes and the M/M/1 queue to consider new types of queue. In this lecture we will consider:

- Multiplexing — ways to share an network connection as $M/M/1$ queues.
- The $M/M/m$ queue
- The $M/M/\infty$ queue

A reminder of the main results from last lecture.

A general birth-death process has a birth rate λ_k and a death rate μ_k in state k. (The death rate in state 0 is assumed to be 0). This was modelled as a Markov chain and lead us to the equilibrium probabilities (probability that the queue is of length k)

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

where

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{\infty} \prod_{i=1}^{k} \frac{\lambda_{i-1}}{\mu_i}} \tag{2}$$

For our $M/M/1$ process with births at rate λ and deaths at rate μ for all k this simplifies to

$$\pi_k = \rho^k \pi_0 \tag{3}$$

(where $\rho = \lambda/\mu$ with $0 < \rho < 1$ is the *utilisation* of the system). And

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{\infty} \rho^k} \tag{4}$$

From which we calculate the average queue length as:

$$N = \frac{\rho}{1 - \rho} \tag{5}$$

Multiplexing

It is usual in the internet that several (maybe thousands) of users share the same data transmission line. Using one line to send several signals is known as multiplexing. Three schemes are commonly used:

- Statistical Multiplexing (the free-for-all option — everybody tries to cram down the same wire and hope)
- Time Domain Multiplexing (the timetable option — you go at 10 past, I will go at 20 past)
- Frequency Domain Multiplexing (the radio channels option — you send at 99.9 kHz, I will send at 102kHz)

All we really need to know for this course is that the first method (statistical multiplexing) all the users compete for the line — effectively they share the same queue. On the other hand, in the other two methods, the available space is split between the users in some method. They don't have to compete but, instead, they are each allocated a private channel which is a fraction of the whole thing.

Let us take this as an example of our $M/M/1$ queue. Imagine we have a router which can send μ packets/second to the outside world. We have n customers each of whom want to send λ_i packets/second. Our two choices are statistical multiplexing (let all customers share the bandwidth) or time/frequency domain multiplexing based on some allocation of available space. Let us assume that the total demand from all our customers is $\lambda = \sum_{i=1}^{n} \lambda_i$ packets/second. Let us further assume that, if we implement time or frequency domain multiplexing we do it according to demand so we allocate each customer a share proportional to their demand. So each customer gets a share:

$$\mu_i = \mu\lambda_i/\lambda \tag{6}$$

(Trivial exercise — prove that $\mu = \sum_{i=1}^{n} \mu_i$). What are the average queues for the various systems?)

Statistical Multiplexing

We remember the fact that an aggregation of n independent Poisson processes, each with a rate of λ_i is the same as a single Poisson process with a rate of λ. Therefore, this is our default answer, the $M/M/1$ queue — we get one system where the number of people queuing is given by $N_q = \rho/(1-\rho)$ and $\rho = \lambda/\mu$ and therefore $T_q = \frac{1}{\mu-\lambda}$.

Time/Frequency Domain Multiplexing

In this case, we have n independent channels. The ith channel is an $M/M/1$ queue with $\rho_i = \lambda_i/\mu_i$. From the definition earlier, therefore, substituting from (6) we get $\rho_i = \lambda_i(\lambda/\mu\lambda_i) = \lambda/\mu = \rho$.

So, we now have n queues which each have $\rho_i = \rho$ and therefore each have an average queue length $N = \rho/(1-\rho)$. The expected total number of queuing packets in the system is the total from each of these n queues which is $n\rho/(1-\rho)$ and the average delay per packet is now (from Little's theorem again) $T = \frac{n}{\mu-\lambda}$. (Trivial exercise — what is the average delay in each queue call it T_i. Why $(\sum_{i=1}^{n} T_i)/n = T$ not generally true?)

So, by splitting the bandwidth into n channels we have multiplied the queuing delay by n. A somewhat startling result. Why would we go to the bother of time or frequency division multiplexing? Sometimes, statistical multiplexing is impossible for various reasons. However, the most common reason is that customer i has paid for lots of bandwidth and customer j (where $i \neq j$) has not. Therefore we allocate a larger share to customer i who does then feel the benefit.

The $M/M/m$ queue

Recall from our queuing theory definitions that an $M/M/m$ queue has a Poisson (Memoryless) input process a Poisson output and m servers. The situation we are thinking of is that where, for example, we queue in the Post Office or banks with m different possible teller windows working.

Each teller windows serves a customer according to a Poisson process with an output rate of μ. Therefore, if the total number of people in the post office is n where $n \leq m$ then the service rate is $n\mu$ when $n > m$ then the service rate is $m\mu$. Again this is a specific case of our general birth death process. We can draw a Markov chain equivalent to this process.

We have a birth death process with the following parameters: $\lambda_i = \lambda$ for all i and μ_i given by:

$$\mu_i = \begin{cases} 0 & i = 0 \\ i\mu & 0 < i \leq m \\ m\mu & i > m \end{cases} \tag{7}$$

We should note that the utilisation is therefore $\rho = \lambda/m\mu$. (And as usual we require that $0 < \rho < 1$).

Substituting into (1) we get

$$\pi_k = \begin{cases} \pi_0 \prod_{i=1}^{k} \frac{\lambda}{i\mu} = \pi_0 \frac{(m\rho)^k}{k!} & k < m \\ \pi_0 \left(\prod_{i=1}^{m-1} \frac{\lambda}{i\mu}\right)\left(\prod_{i=m}^{k} \frac{\lambda}{m\mu}\right) = \pi_0 \frac{m^m \rho^k}{m!} & k \geq m \end{cases} \tag{8}$$

And from $\sum_{i=0}^{\infty} \pi_i = 1$:

$$\pi_0 = \left[1 + \sum_{i=1}^{m-1} \frac{(m\rho)^i}{i!} + \sum_{i=m}^{\infty} \frac{m^m \rho^i}{m!}\right]^{-1} \tag{9}$$

which simplifies to:

$$\pi_0 = \left[\sum_{i=0}^{m-1} \frac{(m\rho)^i}{i!} + \frac{(m\rho)^m}{m!(1-\rho)}\right]^{-1} \tag{10}$$

Now, we know that, if the number of customers in the system is less than or equal to m then all customers will be at a server. We might ask what is the probability of arriving and finding all servers busy (and therefore having to queue for a server). Therefore:

$$\mathbb{P}\left[\text{All Servers Full}\right] = \sum_{i=m}^{\infty} \pi_i = \sum_{i=m}^{\infty} \frac{\pi_0 m^m \rho^i}{m!} = \frac{\pi_0 (m\rho)^m}{m!} \sum_{i=m}^{\infty} \rho^{i-m} \tag{11}$$

We will refer to this quantity as P_Q. Which gives us:

$$P_Q = \mathbb{P}\left[\text{All Servers Full}\right] = \frac{\pi_0 (m\rho)^m}{m!(1-\rho)} \tag{12}$$

where π_0 is given by equation 10. This formula is known as Erlang's C formula after a pioneer of queuing theory.

The next thing we might ask is how many customers (on average) are queuing (rather than being served). This is sometimes known as N_Q and can be given by:

$$N_Q = \sum_{i=m}^{\infty} (i-m)\pi_i = \sum_{i=0}^{\infty} i\pi_{i+m} \tag{13}$$

(since for states below m there are no customers queuing).

Substituting the expression for π_{i+m} from (1) we can get:

$$N_Q = \sum_{i=0}^{\infty} i\pi_0 \frac{m^m \rho^{m+i}}{m!} = \frac{\pi_0(m\rho)^m}{m!} \sum_{i=0}^{\infty} i\rho^i \tag{14}$$

This gives

$$N_Q = \frac{\pi_0(m\rho)^m}{m!} \frac{\rho}{(1-\rho)^2}. \tag{15}$$

Notice that from (12) we get

$$N_Q = P_Q \frac{\rho}{1-\rho}. \tag{16}$$

A Quick Aside Involving Expectation Values

N_Q is the expected size of the queue. If Q_t is the size of the queue at some time t then:

$$\mathrm{E}\left[Q_t\right] = \sum_{i=1}^{\infty} i\mathbb{P}\left[Q_t = i\right].$$

If we want to know the expected size of the queue given that a queue exists then we want to calculate $\mathrm{E}\left[Q_t | Q_t > 0\right]$. Now, we know from basic probability theory that

$$\mathbb{P}\left[X|Y\right] = \frac{\mathbb{P}\left[X, Y\right]}{\mathbb{P}\left[Y\right]}.$$

Therefore, if we want to know the expectation of the queue size given that a queue exists we have:

$$\mathrm{E}\left[Q_t | Q_t > 0\right] = \sum_{i=1}^{\infty} \frac{i\mathbb{P}\left[Q_t = i, Q_t > 0\right]}{\mathbb{P}\left[Q_t > 0\right]}.$$

Now $\mathbb{P}\left[Q_t = i, Q_t > 0\right] = \mathbb{P}\left[Q_t = i\right]$ for $i > 0$ and we earlier defined $P_Q = \mathbb{P}\left[Q_t > 0\right]$ therefore

$$\mathrm{E}\left[Q_t | Q_t > 0\right] = \frac{\mathrm{E}\left[Q_t\right]}{P_Q} = \frac{N_Q}{P_Q}$$

Note that the expected queue size conditioned on the fact that the customer has to queue is therefore:

$$\frac{N_Q}{P_Q} = \frac{\rho}{1-\rho} \tag{17}$$

which can be thought of as the obvious result that, when all the servers are working, the system is equivalent to an $M/M/1$ queue with a service rate of $m\mu$ (remember that $\rho = \lambda/m\mu$).

From Little's theorem, the average time waiting in the queue W is:

$$W = \frac{N_Q}{\lambda} = \frac{\rho P_Q}{\lambda(1-\rho)} \tag{18}$$

Of course once in queue, customers are each served at a rate μ and therefore the average time in the system for a customer (queuing and being served) is:

$$T = \frac{1}{\mu} + W = \frac{1}{\mu} + \frac{P_Q}{m\mu - \lambda} \tag{19}$$

Using Little's Theorem (again!) gives us the average number of customers in the system

$$N = \lambda T = \frac{\lambda}{\mu} + \frac{\lambda P_Q}{m\mu - \lambda}, \tag{20}$$

which equates to

$$N = m\rho + \frac{\rho P_Q}{1 - \rho} \tag{21}$$

The $M/M/\infty$ queue

Finally, we briefly consider the case where $m = \infty$ — our dream system where there is always someone waiting to serve you, no matter how many people arrive.

Using a similar derivation to the previous example, change equation 1 to:

$$\pi_k = \pi_0 \left(\frac{\lambda}{\mu}\right)^k \frac{1}{k!} \tag{22}$$

Therefore we have, from our condition that $\sum_{i=0}^{\infty} \pi_i = 1$:

$$\pi_0 = \left[1 + \sum_{i=1}^{\infty} \left(\frac{\lambda}{\mu}\right)^n \frac{1}{n!}\right]^{-1}, \tag{23}$$

which we alertly notice is the equation for an exponential and therefore

$$\pi_0 = e^{-\lambda/\mu}. \tag{24}$$

And substituting in (22) we get

$$\pi_k = \left(\frac{\lambda}{\mu}\right)^k \frac{e^{-\lambda/\mu}}{k!} \tag{25}$$

which is, obviously, a Poisson system with parameter λ/μ.

The average number in the system from Little's equation is:

$$N = \frac{\lambda}{\mu} \tag{26}$$

and the delay is

$$T = \frac{1}{\mu} \tag{27}$$

We could have saved ourselves this derivation by simply making the observation that, in the $M/M/\infty$ system, nobody stands in a queue before joining a server. Therefore everyone is served instantly by a Poisson process of rate μ.