Introduction to Queueing Theory

Prof. N. Selvaraju
Department of Mathematics
Indian Institute of Technology Guwahati, India

Lecture - 17
Erlang's Loss System, Erlang Loss Formula, Infinite-Server Queues

Hi and hello, everyone. What we have been seeing so far is the Birth-Death Queueing Systems, meaning queueing systems where the arrivals and departures happen one at a time. So far, we have seen a few models; let us continue our discussion with a few more models this week as well in the next few lectures. he first one that we will see now is what we call Erlang's loss system, which is in our queueing notation; if you have to denote it, this will be denoted by an $M/M/c/c$ model.

- One of the first and the most famous system in queueing theory, with wide applicability (e.g. telecommunications design).

  For example, in telecommunication designs where originally, Erlang developed this particular model in the context of telephone traffic.

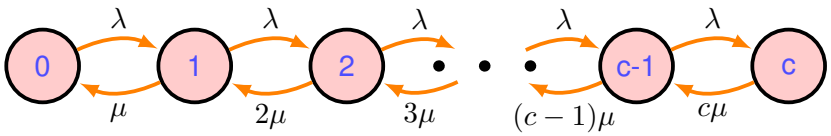
- The origin of the traffic theory or congestion theory started by the investigation of this system (by Erlang).

  So, this is one of the first models that has been considered by him; along with $M/M/c$ and $M/M/c/c$, this is a loss system. So, why do we say this is a loss system? As you see the description that the arrival process is a Poisson process with rate lambda, the service process is also the service times follow the exponential distribution, and there is c number of servers, and the capacity of the system is c. So, basically, what you see here is that there is no queue happens here. Because either all servers are busy, then any arriving customer would be a loss to the system, so that is why this is referred to as a loss system. Whereas where the systems where there is queueing happening like in $M/M/1$ or $M/M/c$ or even $M/M/c/K$, there is some amount of people can queue, whether it is a finite number or we are assuming to be an infinite number. There is always a possibility of people queueing up for service customers queueing up for service. So, those are all called delay systems because what happens there is that you have customers waiting, so then after some delay, he is going to get his service. So, those are all referred to as some delay systems as opposed to loss systems in this particular case, but here there is no delay happens. Even in the MM cK model, of course, there will be some amount of people who would not be able to enter into the system. But here, it is the case that there is no queueing happens, so it is a pure loss system.

- And as we said, this is one of the very first models, and this model can be regarded as a special case of $M/M/c/K$

with $K = c$, and modelled by a BDP with rates

$$\lambda_n = \begin{cases} \lambda, & n < c \\ 0, & n \geq c \end{cases} \qquad \text{and} \qquad \mu_n = n\mu, n = 1, 2, \ldots, c.$$



- The stationary distribution can be obtained in the usual manner as

$$p_n = \frac{(\lambda/\mu)^n}{n!} \Bigg/ \left(\sum_{i=0}^{c} \frac{(\lambda/\mu)^i}{i!} \right) = \frac{\dfrac{r^n}{n!}}{\displaystyle\sum_{i=0}^{c} \frac{r^i}{i!}}, \qquad 0 \leq n \leq c, \;\; r = \lambda/\mu.$$

You will get this as the distribution, and as you would see that this is nothing but a truncated Poisson distribution. So, once we have this stationary distribution in place, we can again look at various performance measures with respect to this particular system which might depend on the number in the system.

- The most important measure of the system is p_c.

That means, what is the probability that you find the system in this particular state c at which what happens is that there is no entry of any more customers being allowed into the system. So, typical happens suppose if a telephone line has some c channels or right or a network which we call talk in terms of bandwidth, so what is the capacity right. If it is not able to connect, then the call gets dropped. So, that, so that is where this quantity will come in. So,

$$p_c = \frac{\dfrac{r^c}{c!}}{\displaystyle\sum_{i=0}^{c} \frac{r^i}{i!}} = B(c, r)$$

which is referred to as **Erlang's loss formula or Erlang-B formula or Erlang's first formula**.

We have already seen an Erlang-C formula, in the context of $M/M/c$, which is the probability of delay; it is an important measure. Similarly, what is the probability of the loss here. There what is the probability of queueing is what you are interested in the Erlang-C formula. And here is the probability of loss, so this is what is the probability that the system state would be in c at any point of time. So, this is what is referred to Erlang's loss formula and Erlang's-B formula Erlang's first formula; here, this is all called the blocking probability this is what is called the call the blocking probability. But then, there are true notions of blocking here that you need to understand. It so happens that both the notions coincide in this particular case of the Erlang loss system, but it may not be true in general for any other loss system. We will be going to see one later on where you will see that these two quantities are really different. But it so happens here that both this concept of blocking is same as is

equal or is one and the same in the case of Erlang. So, this itself would be simply referred to as Erlang's loss formula.

♦ (Time blocking) p_c is the probability that all c servers are occupied at an arbitrary time (= the fraction of time that all c serves are occupied)

We know from the notion of stationary distribution and limiting distribution that this is what you get. When the Markov process has a stationary distribution, and if you start with the stationary distribution, then the Markov process is what we call the stationary Markov process. For a stationary Markov process, these quantities would be given by p_c the equilibrium probability of finding the system in that particular state. So, in so in that sense, time blocking would be given by this p_c for a stationary Markov process by the equilibrium distribution or stationary distribution, and that would then equal to p_c. So, p_c is what we called here or which is also called as capital $B(c, r)$ as the Erlang-B formula, is the probability that all c servers are occupied at an arbitrary point of time and which is also the fraction of time all c servers are occupied. This is clear from our knowledge of the Markov process that this is what you are going to be interpreting for this, and this is what is referred to as time blocking. Now there is a notion of call blocking.

♦ (Call blocking) p_c is also the probability that an arriving customer finds all c servers occupied (= the fraction of arriving customers that are lost). This is due to the Poisson arrivals and the PASTA property.

So, we are not concerned about an outside view of the system at an arbitrary point of time; we are looking at the system at the point of arrival of a customer; what happens to the system is what is more interested in such systems. That is why this call blocking when a call arrives. What happens to that call? So, what is the probability of this blocking here? Now, here it so happens this p_c is also the probability that an arriving customer would find all c servers occupied, which would also be equal to the fraction of time, the fraction of arriving customers that are lost. So, this p_c here has now as you can see a four interpretations as we see here because time blocking is also equal to call blocking. And why this is? How is this happening? Because of the nature of Poisson arrivals here and the PASTA property, that is what is giving us this call blocking, the same as the time blocking in this case. So, and hence totally in such case without any distinction, we will be simply calling this a blocking probability in this particular case, when there is no difference between this that one might simply refer to as blocking probability. This is what is given by Erlang's formula, and this is what is relevant in this particular case.

- Due to the importance of $B(c, r)$ in practical problems, calculators are readily available (e.g., http://www.erlang.com)

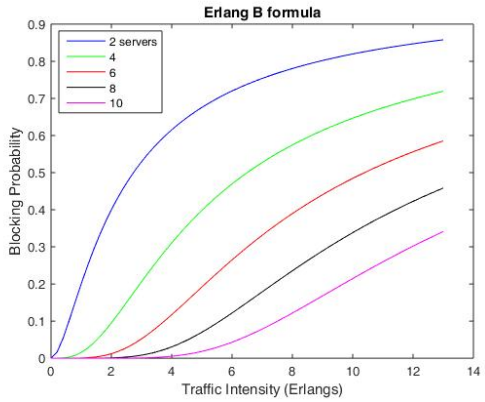
So, you can utilize you can look at this particular thing for an online calculator of this Erlang's formula, which is for ready reference. Like any other tables that you know have been developed for in different fields like in statistics like, if there are tables of distributions it is given normal distribution and so on. So, in the same way, this is also available tables as well as calculators to get the value of $B(c, r)$ is available. But the beauty of this result or the significance of this result is because of certain insensitivity properties that this particular result holds. What do we mean by that? We see that this formula it can be shown later, but we will not be showing it, but it can be easily seen when we see MG type of models like when you do a similar analysis you would arrive at this particular result.

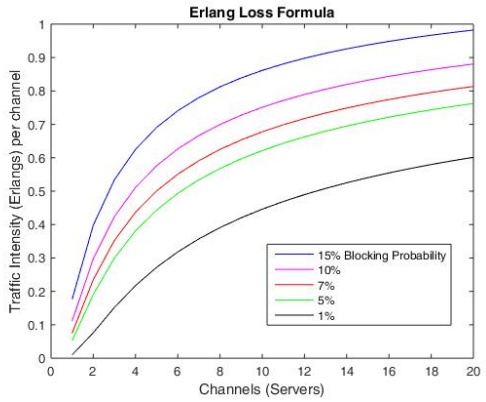
- This formula is valid for any $M/G/c/c$, independent of the form of the service-time distribution.

  Whatever be the service time distribution, this formula holds true, a very significant result. And that is the power of this result; that is what we call the robustness of the result of Erlang-B formula or the insensitivity of this result to the service time distribution.

- The two formulae by Erlang (delay and loss) are fundamental results in teletraffic engineering and queueing theory, and are still used today.
  ♦ They relate quality of service (QoS) to the number of available servers.

As a typical example or application of this formula or how one looks at these particular formulas, both of them plots related to this Erlang-B formula or loss formula. The one and the same though I have given two different ways, it is the title, just to get familiarized that these two are one and the same.





We just depicted this here just because of the importance of this formula, but a similar kind of analysis can be done with respect to any performance measures, and that is what is all about performance analysis. Performance modeling is when you take a system, you create a model and get the quantities, performance analysis you do this; this is up to you like what kind of analysis that is required to be done and what you wanted to do like you can carry on with this kind of things. Now some remarks are in order with respect to this particular model.

- While $r = \lambda/\mu$ is the offered load, $r(1 - B(c, r))$ is the carried load.

  Because p_c is the fraction of arrivals that are lost or load that is lost, so, $1 - B(c, r)$ would be the load that is going inside. So, $r(1 - B(c, r))$ would be the carried load.

- The throughput, defined as the rate at which customers depart from the system after being (admitted and) served, is given by $\lambda(1 - B(c, r))$

  Because out of this λ proportion of customers who are arriving in a Poisson fashion, p_c is the fraction of customers that is lost, again because of this call blocking idea, not because of time blocking ideas here. So, that is $1 - B(c, r)$, so $\lambda(1 - B(c, r))$ would be the throughput for this system.

  ▶ This is the rate at which customers are accepted for service (these two rates must be equal in steady-state)

- Loss and delay systems

  And as we said, this is the loss system, and in a loss system, sometimes, such as this particular Erlang loss system, these kinds of wordings are used. So, you need to get familiarized, which is basically this is a system that is referred to as a system where blocked calls are cleared; they are not in the queue. The moment you are not able to get access to the server, you are lost; you are cleared like you are out of the queueing system. So, that is as opposed to a delay system. So, that is where the delay means where you are being kept in the system, but with the delay, you will be able to access the server.

- In a loss system (such as this), blocked customers are said to be cleared.

- The expected number of busy channels equals the carried load (Prove this!), and hence the expected number of idle channels equals $c - r(1 - B(c, r))$.

  So, the expected number of busy channels is $r(1 - B(c, r))$. So, how do you compute this? Prove this as I said like you can look it up, or otherwise, you can simply compute. You have the distribution of what is the probability that the number of channels n some $1, 2, 3, 4, 5$ number of channels is busy. So, what is the number of channels busy, multiply to the corresponding probabilities will give you this. Now, $c - r(1 - B(c, r))$ is the free channel, so c minus the expected number of busy channels will give you the number of free channels. These are all performance measures that you would talk about with respect to this particular system.

  Exercise. *Determine L, L_q, W and W_q.*

  And again, all these quantities, for example, throughput, carried load or busy channels, idle channels; everything you can look at in terms of parameters and analyze deeply so on. So, this is what we are saying for every model it is possible to do, and that depends upon the requirement that you will be doing that.

But this formula, if you look at it carefully here, you need to compute this p_c you need to compute $r^c/c!$ and $\sum_{i=0}^{c} r^i/i!$ which involves a critical quantity which is $c!$. But as long as c is very small, you can directly compute $c!$. But if c is a large number, then this factorial itself cannot be handled by the computer within its capacity. So, you need to find a way of getting this; computing this factorial. Like 170, if it crosses, then you are running into trouble.

- But many real-life applications have a large value of c.

The telephone line capacity if you think or nowadays in a place where the call-centers are there, 200 lines is not an uncommon phenomenon that you might find in a big call center operations. So, this c which is a number of channels or number of servers, assuming large values is not an uncommon phenomenon. So, one needs to handle it, but in that case, this direct application of this formula runs into trouble, or it will take a long time even if it is doable in a complicated way.

- An alternative approach is to use an iterative relationship, by observing (Prove this!) that

$$B(c, r) = \frac{rB(c-1, r)}{c + rB(c-1, r)}, \quad c \geq 1, \text{ with initial condition } B(0, r) = 1.$$

So, start with $B(0,r) = 1$, so then $B(1,r)$ you can compute $B(2,r)$ you can compute and so on. So, you can compute for the same r the required number of servers, c could be thousand you can easily compute it very easily. But whereas, a direct application using that factorial function would be very difficult to implement. So, this is the relationship that you know we can use to compute. Say, for example, this is a very simple one, but you can go further and see how much you can go.

Example.
Let $\lambda = 6, \mu = 3, c = 4$. Calculate the fraction of customers blocked for an $M/M/c/c$ system.

Since $r = 2$, we want to compute $B(4,2)$.
From the iterative process, we have $B(0,2) = 1, B(1,2) = 2/3, B(2,2) = 2/5, B(3,2) = 4/19$ and $B(4,2) = 2/21$.

Of course, for the same system, you may have other questions that you can also answer through the other performance measure you might develop. Now, what is the relationship between these Erlang's-B and C formulas.

- $B(c,r)$ can also be used to determine performance metrics of $M/M/c$ model.
- Recall that $C(c,r) = 1 - F_{T_q}(0)$ is the probability of delay in an $M/M/c$ model. This can be computed using the following relationship (Prove this!) between the two formulas

$$C(c,r) = \frac{cB(c,r)}{c - r + rB(c,r)}$$

- Also, for $M/M/c$ model, recall that we can write L, L_q, W and W_q in terms of $C(c,r)$ (Try this!). For example, L_q can be given by

$$L_q = C(c,r)\frac{\rho}{1-\rho} = C(c,r)\frac{r}{c-r}$$

Example.
Say, for example, in the same example, if you want to compute $C(4,2)$ what is the probability of delay. This is the previous one says this is the probability of loss the blocking probability is what this $B(4,2)$. So, the corresponding $C(4,2)$ would be if it was an infinite capacity queueing system, a delay system as opposed to a loss system, then what would have been the probability of delay is what this one is, $4/23$ and $L_q = 4/23$.

I can compute L and so on for the corresponding one; so, this relationship can also be exploited to obtain the quantities related to Erlang using the C formula or corresponding $M/M/c$ model. So, this is the loss system that e have seen. Now, what we will see next is, again, this is a no queue system, but this is not a loss system in a way.

But this is called infinite server queues or queues with unlimited service capacity, which we denote by $M/M/\infty$, which means there is an infinite number of servers; this is basically a self-service system. If I have to say this could be something like there is a radio broadcast going on, and then the number of customers who want to listen to that particular broadcast, pretty much everyone can listen to that. So, there is no need to wait before you listen to that. Similarly, suppose there is a TV show which is being aired, and then you are watching the TV show. So, pretty much everyone on the earth, if you want to watch, they can always watch. So, that is what in the self-service model is what this system that.

- The system can be modelled using a BDP with $\lambda_n = \lambda, \mu_n = n\mu, \forall n$.

  Do not ask me whether people who are tuning in to a particular radio show; will arrive according to your Poisson process.

  So, that is the critical question, but if it is so, if the data shows that people are tuning in according to your Poisson process, then this model is applicable, and they listen to number the duration, they listen to an exponential amount of time again this M can be questioned like this every assumption can be questioned, but under the assumptions, the model is true. And we have a BDP for this, and that will have this particular rate that you have here.

  The steady-state system size distribution is obtained as

$$p_n = \frac{r^n}{n!} p_0, \quad p_0 = \left(\sum_{n=0}^{\infty} \frac{r^n}{n!} \right)^{-1} = e^{-r}$$

$$\text{Thus,} \quad p_n = \frac{r^n e^{-r}}{n!}, \quad n \geq 0$$

- The $Poi(r)$ distributed system size distribution holds true for an $M/G/\infty$ queue in general (robustness).
- Here, $L = r = \frac{\lambda}{\mu}, \quad L_q = 0 = W_q, \quad W = \frac{1}{\mu}$.

So, you have a partial result or partial confirmation or confidence-boosting measure with respect to this particular form of the distribution that you better have. So, this is true for the $M/G/\infty$ model. So, this particular steady-state distribution which is $Poi(r)$, is true as long as the arrival process is Poisson and it is insensitive or independent of the service time distribution what is required is the mean rate what is the mean service rate is what is it will depend upon, which is from there only you are going to get μ. So, this is true, and this is what you have here with respect to this infinite server queue. So, we will possibly end here this in this lecture. So, we will continue with some more types of models in the next lecture.
Thank you, bye.