Introduction to Queueing Theory

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

Lecture - 18

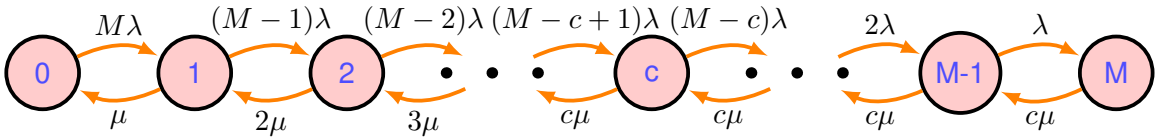
Finite-Source Queues, Engset Loss System, State-Dependent Queues, Queues with Impatience

Hi and hello, everyone. What we will see next is this particular Queueing type system, which is called Queues with Finite Input Source. What is the difference? In all the models that we have considered so far, we have assumed that "the calling population" meaning that the population from which the arrivals are to occur, is infinite or large enough that it does not make a difference to the arrival rates. So, it is large, quite large, that is what we have assumed. So, that is what we say it is a calling population, that is, the population from which the customers arrive in the queueing system. So, we assumed it to be infinite technically, but it is large enough is what proper way to describe that. But there are systems where the calling population is finite. In fact, every calling population is finite; because of largeness, we assume that to be infinite. But there are situations where it is; the size of the population is very minimal, in comparison to the number of servers or something that we are considering into that part. So, and hence the future event occurrences are functions of the system state naturally that come into that mode whenever this calling population is finite. Say a typical application of this model or typical example of this model is what we call a **machine repairmen problem** so that you can understand the context, but this same context can be applied in many different situations where exactly similar kinds of things can happen. Machine interference problem or machine repairmen problem. So, what do we have here?

- We assume c servers (repairmen) are available for M ($\geq c$) machines.
  Service times (time required to repair a machine) $\sim Exp(\mu)$, and the time for breakdown (or life time) of a machine $\sim Exp(\lambda)$.
  Number in the system corresponds to the number of broken machines.

  So, there are machines that work for an exponential λ amount of time, then fail. And whenever it is the repair is started, the repair time duration is exponential with parameter μ; then the machine becomes up, and so on.

- Transition rate diagram for the finite-source queue:



This is what is the finite source BDP model, which you can always remember with respect to this kind of situation that you have in mind machine repairmen. So, this could be in many different situations; this can come like there could

be a fleet of, say, c cars with M drivers. So, the drivers can come at exponential rate, and then they can be using the car or act as a driver for an exponential amount of time if you assume then the exactly same thing happens and so on. There will be many situations which you can fit here.

- The system can be described by a BDP with state-dependent rates:

$$\lambda_n = \begin{cases} (M-n)\lambda, & 0 \le n < M \\ 0, & n \ge M \end{cases} \quad \text{and} \quad \mu_n = \begin{cases} n\mu, & 1 \le n < c \\ c\mu, & n \ge c \end{cases}$$

- The stationary system state probabilities are obtained as

$$p_n = \begin{cases} \frac{M!/(M-n)!}{n!} r^n p_0, & 1 \le n < c \qquad (r = \frac{\lambda}{\mu}) \\ \frac{M!/(M-n)!}{c^{n-c}c!} r^n p_0, & c \le n \le M \end{cases}$$
$$= \begin{cases} \binom{M}{n} r^n p_0, & 1 \le n < c \\ \binom{M}{n} \frac{n!}{c^{n-c}c!} r^n p_0, & c \le n \le M \end{cases}$$

- Denote the coefficient in front of p_0 in p_n by a_n (i.e., $p_n = a_n p_0$). Then

$$p_0 = \frac{1}{1 + a_1 + a_2 + \cdots + a_M}$$

- Average number of customers in the system:

$$L = \sum_{n=1}^{M} n p_n = p_0 \sum_{n=1}^{M} n a_n$$

- To obtain L_q, W, W_q, we have to first find the effective mean rate of arrivals into the system:

$$\lambda_{\text{eff}} = \sum_{n=0}^{M} (M-n)\lambda p_n = \lambda(M - L)$$

So, L is the average number of customers in the system, meaning the average number of failed machines, and M is the total number of machines; $M - L$ is the average number of working machines. So, lambda times each one of them. So, this is what the effective arrival rate is, turning it to be, as, one would think intuitively.

Now once we have this lambda effective, then Lq in the usual way you can obtain as

$$L_q = L - \frac{\lambda_{\text{eff}}}{\mu} = L - r(M - L)$$

And

$$W = \frac{L}{\lambda(M-L)}, \qquad W_q = \frac{L_q}{\lambda(M-L)}$$

- For $c = 1$, the expression for the system-state probabilities reduces to

$$p_n = \binom{M}{n} n! r^n p_0, \quad 0 \le n \le M$$

and rest of the analysis is similar.

- Invariance results for finite-source queues:

▶ The steady-state distribution is valid for any finite-source systems with exponential service, independent of the nature of distribution of time to breakdown, as long as the lifetimes are independent with mean $\frac{1}{\lambda}$.

The distribution really does not matter, which is a quite good invariance property because the working time of the machine now I can assume to be any distribution; still, this same results will hold, that is what it means.

▶ If $c = M$, the repair distribution can be G as long as failure times are exponential.

There could be a generalization of this particular finite source model from many different angles; the simplest and the simple generalization is the one where you are also assuming some spares are available for the machines.

- **Generalization with spares:** We assume that there are M machines in operation plus an additional Y spares.

That means if any one of them fails, then the spare can come into play for the operations to continue. In the meantime, the failed machine can be repaired.

- At any given time, there are at most M machines in operations, so the rate of failures is at most $M\lambda$ (i.e., spares that are not in operation do not contribute to the failure rate).

$$\lambda_n = \begin{cases} M\lambda, & (0 \le n \le Y) \\ (M - n + Y)\lambda, & (Y \le n < Y + M) \\ 0, & (n \ge Y + M), \end{cases} \quad \{n \text{ represents the number of failed machines}\}$$

$$\mu_n = \begin{cases} n\mu, & (0 \le n < c) \\ c\mu, & (n \ge c). \end{cases}$$

- This model can also be analyzed similarly as that of the finite-source queue model. The only thing the expression will get a little bit trickier because of this expansion or the different rates at different points, the state-dependent rates that are coming into the picture. Otherwise, the analysis can be done in a similar fashion; that is what you can see.

This is a simple example like whatever we talked about, what kind of questions that can arise, and how one can get the things using this machine.

Example.
Assume that two workers are responsible for 10 milling machines in a factory.

Machines run on the average for 20 minutes and then require an average 5 minute of maintenance service, both times being exponentially distributed.

Here, $\lambda = 1/20$ and $\mu = 1/5$ per minute. Then $r = 1/4$.

We first compute p_0 as $p_0 = 0.065$. Then we can get $L = \sum np_n = 3.17$ machines, which represent the average number of machines under repair.

This means $M - L = 6.83$ is the average number of running machines.

The expected downtime of a machine that requires repair is $W = 9.28$ minutes.

Exercise. *Compute the average fraction of idle time of each worker as* $p_1(1/2) + p_0$.

Question: Whether to add one more worker? Impact?

So, this is a typical example. So, all these things like you can once you have the expression then you can compute this very easily, and then you can get the answers to whatever question that you are looking for it. Now, there is we can also look at waiting times in such finite source queues because this is also a little relevant as of compared to the other cases. In some cases, we never bothered about it because it is more similar to the other. But there wherever there is some differences might come, we look at here.

- Observe that $a_n \neq p_n$, for the general finite-source queue (machine-repair problem without spares).
- Proceeding as in the case of $M/M/c/K$, we can derive the arrival-point probabilities $\{a_n\}$ as,

$$\begin{aligned} a_n &= P\left\{N = n \text{ in the system} \mid \text{arrival about to occur}\right\} \\ &= \frac{P\left\{\text{arrival about to occur} \mid N = n\right\} p_n}{\sum_n P\left\{\text{arrival about to occur} \mid N = n \text{ in system}\right\} p_n} \\ &= \lim_{\Delta t \to 0} \left\{ \frac{\left[(M-n)\lambda \Delta t + o(\Delta t)\right] p_n}{\sum_n \left[(M-n)\lambda \Delta t + o(\Delta t)\right] p_n} \right\} = \frac{(M-n)p_n}{\sum_n (M-n)p_n} \\ &= \frac{(M-n)p_n}{M-L} \end{aligned}$$

- An interesting fact for this model is that $a_n(M) = p_n(M-1)$ (not necessarily true for the case with spares).

- The waiting time distribution again turn out to be in terms of cumulative Poisson sums as

$$\begin{aligned}
F_{T_q}(t) &= P\left\{T_q \le t\right\} \\
&= F_{T_q}(0) + \sum_{n=c}^{M-1} P\left\{n-c+1 \text{ service completions in } \le t | \text{arrival finds } n \text{ in system}\right\} a_n \\
&= F_{T_q}(0) + \sum_{n=c}^{M-1} a_n \int_0^t \frac{c\mu(c\mu x)^{(n-c)}}{(n-c)!} e^{-c\mu x} dx \\
&= F_{T_q}(0) + \sum_{n=c}^{M-1} a_n \left\{1 - \int_t^\infty \frac{c\mu(c\mu x)^{(n-c)}}{(n-c)!} e^{-c\mu x} dx\right\} \\
&= 1 - \sum_{n=c}^{M-1} a_n \sum_{i=0}^{n-c} \frac{(c\mu t)^i e^{-c\mu t}}{i!}
\end{aligned}$$

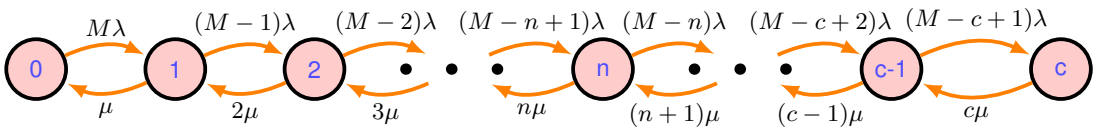
So, this is all about the finite source queueing model. Next, what we will see is again a loss system which is called as Engset loss system.

- This is a finite-population version of Erlang's loss system (named after T.O. Engset, a Norwegian mathematician and engineer).
- We still have c parallel servers with no waiting customers, but we now have a finite population M ($> c$) of arriving customers.
- The system state can be modelled by a BDP with rates:

$$\lambda_n = \begin{cases} (M-n)\lambda, & 0 \le n < c \\ 0, & n \ge c \end{cases} \quad \text{and} \quad \mu_n = \begin{cases} n\mu, & 1 \le n \le c \\ 0, & n > c \end{cases}$$

Both the loss systems together is what is very critical for the telegraphic theory or when you are trying to have a model. If it is an infinite source population, then one would use Erlang's loss formula, and when it is a finite source population version, one will appeal to this Engset loss system. So, that is why together with respect to the loss probabilities or blocking probabilities or anything loss system model that you consider both these models are very much important. So, this is the finite source population version.

- Transition rate diagram for the Engset loss system:



- The stationary system state probabilities are obtained as

$$p_n = \frac{(M-n+1)}{n}\frac{(M-n+2)}{n-1}\cdots\frac{(M-1)}{2}\frac{M}{1}\left(\frac{\lambda}{\mu}\right)^n p_0 = \binom{M}{n} r^n p_0, \quad n = 1, 2, \ldots, c; r = \frac{\lambda}{\mu}$$

Now, $\sum_{n=0}^{c} p_n = 1 \quad \Rightarrow \quad p_0 = \left[\sum_{i=0}^{c} \binom{M}{i} r^i\right]^{-1}$, giving us

$$p_n = \frac{\binom{M}{n} r^n}{\sum_{i=0}^{c} \binom{M}{i} r^i} = \frac{\binom{M}{n} \left(\frac{r}{1+r}\right)^n \left(\frac{1}{1+r}\right)^{M-n}}{\sum_{i=0}^{c} \binom{M}{i} \left(\frac{r}{1+r}\right)^i \left(\frac{1}{1+r}\right)^{M-i}}, \quad n = 0, 1, 2, \ldots, c.$$

which is a truncated binomial distribution.

Now, once we have this, of course, then you can see the usual performance measures which we can obtain. But we will concentrate only on one performance measure here, which is more important and which is different from the case of your Erlang loss model. So, which is the blocking probability concept. So, now, as we said earlier, in the case of Erlang's loss system itself, there is this notion of two types of blocking. One is time blocking; the other is call blocking.

- (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), because of stationary Markov process property.

  So, this time blocking probability remains exactly the same as the one that you would encounter in the case of an Erlang loss system, but the call blocking the notion would be varying now. Because there, the call blocking probability also was equal to p c in the Erlang loss model, but here the call blocking will not be equal to this. The reason is what we will see now.

- (Call blocking) In the Engset model, the call blocking probability (the probability that an arriving customer finds all c servers occupied) does not equal the time blocking probability p_c. This is because the arrival process is not Poissonian (because of state dependent arrival rate) and hence PASTA does not hold.

  Since PASTA is not true here, so this call blocking probability is not equal to the time blocking probability which remains as p_c. So, p_c remains as, in one sense, one kind of requirement, which is the time blocking probability, which gives you the probability that all c servers are occupied at an arbitrary point of time. This comes from the notion of the limiting distribution and the fraction of time all c servers are occupied comes from the notion of stationary distributions, and these two are one and the same in that particular case. So, that is also equal to the equilibrium distribution corresponding probability, which is p_c. No problem, the Erlang loss system Engset loss system both behaves in a similar way. But as far as call blocking is concerned, this is not equal because PASTA property is no longer true here. So, what is then the probability? So, for which you want to see what is a_n here

- Let a_n be the probability that the system is in state n upon an arrival. Then

$$a_n = \frac{(M-n)p_n}{\sum_{i=0}^{c}(M-i)p_i}, \quad n = 0, 1, \ldots, c$$

  Proof:

- Consider a long period of time T.
- During this time, the average time spent in state n is $p_n T$.
- During this time, the average number of arriving customers (who find the system in n) is $(M-n)\lambda p_n T$.
- The total number of arrivals in time T is on the average $T\sum_{i=0}^{c}(M-i)\lambda p_i$.
- Thus, the proportion of arrivals that find the system in state n is a_n.

- It can be shown (Prove this!) that $$a_n = \frac{\binom{M-1}{n} r^n}{\sum_{i=0}^{c} \binom{M-1}{i} r^i}, \quad n = 0, 1, 2, \ldots, c.$$

- Writing explicitly the dependence of these probabilities on the total number of customers, we get

$$a_n(M) = p_n(M-1), \quad n = 0, 1, \ldots, c,$$

i.e., an arriving customer sees a system where is there is one customer less (itself!) in equilibrium.

- We thus have the call blocking probability as

$$a_c(M) = p_c(M-1) = \frac{\binom{M-1}{c} r^c}{\sum_{i=0}^{c} \binom{M-1}{i} r^i}$$

which is known as **Engset's blocking formula**.

As opposed to Erlang's blocking formula, Engset's blocking formula refers to the call blocking probabilities because this is what is of interest. When a call comes, what is the probability that it will be blocked? So, $\frac{\binom{M-1}{c} r^c}{\sum_{i=0}^{c} \binom{M-1}{i} r^i}$ is what is relevant here, not exactly p_c, which is still a blocking probability, but it is a time blocking probability. But what is more important is the call blocking probability which is $a_c(M)$; an arriving customer finds all the servers busy, that probability is given by this, and hence this is the Engset blocking formula. So, what do we have now?

♦ For Engset model, the call blocking probability in a system with M customers equals the time blocking probability in a system with $M-1$ customers.

- One can compute other performance measures like L, etc, in the usual way after computing the effective arrival rate λ_{eff}.
- A typical application of the Engset model is on telephone traffic modelling in access network.
  - Assume that the potential users of a link is moderate (say, a closed group of office network)
  - Customer = call

- $\lambda =$ call arrival rate per idle user (calls per unit time)
- $1/\mu =$ average call holding time (time units)
- $M =$ number of potential users
- $c =$ link capacity (channels)

*** A call is lost if all c channels are occupied when the call arrives and the Engset blocking formula $a_c(M)$ gives the probability of such an event.

The objective could be to decide c such that this probability is bounded (say, at 1%).

So, this is what you might see like in a closed network; you want to determine what would be the link capacity and the number of channels to be put in the link. So, whenever a call is made out of this finite population of people, a call is made, that the blocking probability is not able to connect the probability of that would be bounded by a certain quantity; say 1%, 5%, and so on.

- Another application is a taxi service system with c taxis and M operators (drivers).

Like this, you can think about many other different uses for this. So, this is the Engset model, the Engset loss system model that you have seen, which is both the loss system together is what the loss systems which we have. The main difference is the blocking probability idea between these two and that this Engset formula is also very much used like Erlang's loss formula in reality. You can browse it, and you can find it out. So, what we see next is queues with state-dependent service. So, what is that?

- The mean service rate is dependent on the state of the system (number in the system).
- The simplest model is the one in which a single sever has two mean service rates, say, slow and fast (e.g., a machine with two speeds). Work is performed at the slow rate until there are k in the system, at which point there is a switch to the fast rate.
- With the usual assumptions, we then see that the mean rate μ_n is now explicitly depends on the system state n as

$$\mu_n = \begin{cases} \mu_1, & 1 \leq n < k \\ \mu, & n \geq k \end{cases}$$

- Assuming that the arrival process is Poisson with parameter λ, like a simple $M/M/1$ situation. We have $M/M/1$ situation, but with two service rates up to certain states, it is one service rate, and beyond that, it is another one. So, your p_n in the usual way you can obtain it to be

$$p_n = \begin{cases} \rho_1^n p_0, & 0 \leq n < k \\ \rho_1^{k-1}\rho^{n-k+1}p_0, & n \geq k \end{cases} \qquad \big[\text{Here, } \rho_1 = \lambda_1/\mu, \rho = \lambda/\mu < 1\big]$$

- Now, $\sum p_n = 1$ implies that

$$p_0 = \left(\sum_{n=0}^{k-1}\rho_1^n + \sum_{n=k}^{\infty}\rho_1^{k-1}\rho^{n-k+1}\right)^{-1} = \begin{cases} \left(\frac{1-\rho_1^k}{1-\rho_1} + \frac{\rho\rho_1^{k-1}}{1-\rho}\right)^{-1}, & \rho_1 \neq 1, \rho < 1 \\ \left(k + \frac{\rho}{1-\rho}\right)^{-1}, & \rho_1 = 1, \rho < 1 \end{cases}$$

- The expected queue size (with $\rho_1 \neq 1$):

$$\begin{aligned} L = \sum_{n=0}^{\infty} np_n &= p_0 \left(\sum_{n=0}^{k-1} n\rho_1^n + \sum_{n=k}^{\infty} n\rho_1^{k-1}\rho^{n-k+1} \right) \\ &= p_0 \left[\rho_1 \sum_{n=0}^{k-1} n\rho_1^{n-1} + \rho_1 \left(\frac{\rho_1}{\rho} \right)^{k-2} \sum_{n=k}^{\infty} n\rho^{n-1} \right] \\ &= p_0 \left(\frac{\rho_1 \left[1 + (k-1)\rho_1^k - k\rho_1^{k-1} \right]}{(1-\rho_1)^2} + \frac{\rho\rho_1^{k-1} \left[k - (k-1)\rho \right]}{(1-\rho)^2} \right) \quad \text{(on simplification)} \end{aligned}$$

- We can find L_q using the relationship:

$$L_q = L - (1 - p_0)$$

- W, W_q from Little's formula:

$$W = L/\lambda, \qquad \text{and} \qquad W_q = L_q/\lambda$$

- Note that the relationship $W = W_q + 1/\mu$ not true here, since μ is not constant but depends on the system-state switch point k. But, by combining the above equations, we see that

$$W = W_q + \frac{1 - p_0}{\lambda}$$

  $\implies$ expected service time is $(1 - p_0)/\lambda$.

So, here is what we have? We have a system with a state-dependent service. So, we have basically at one time switching of service rates.

- The simple model can be generalized to a "$c - server$" system with a rate switch at $k > c$.
  ■ Another generalization could be to have two rate switches (or even more if desired).

- Another possible type of model is the one where the mean service rate changes whenever the system size changes. For example, we can assume single server, Markovian state-dependent-service model with

$$\mu_n = n^{\alpha}\mu.$$

  This gives us $p_n = \dfrac{r^n}{(n!)^{\alpha}} p_0$ with $p_0 = \left(\displaystyle\sum_{n=0}^{\infty} \frac{r^n}{(n!)^{\alpha}} \right)^{-1}$, where the infinite series converges for any r as long as $\alpha > 0$ but has closed form solution only for $\alpha = 1$.
  ▶ One has to use some numerical procedures to compute p_0 and other performance measures of interest.

So, you see here the models even in a birth-death scenario if you have $\mu_n = n^{\alpha}\mu$ such rates if you are going to have then immediately the analytical tractability of the model might be lost; but still, you can employ the numerical procedures to get the solutions. So, this is a state-dependent service thing here that you have here. Next, what we will see is basically, of course, these are not the only service-dependent case. So, there could be many different ways in

which the μ_n's can depend on the state. Next what you will see is the queues with impatience, it is as important as the arrivals and services. Because for a system operator, system manager, this is very important, like what is the customer feels about his wait in the system so that he can act accordingly. So, that can be brought in only if you bring in some sort of this impatient behaviour.

- Customers are impatient if they tend to join the queue only when a short wait is expected and tend to remain in line if the wait has been sufficiently small.

  But the longer waits either at the time of joining or while waiting in the queue if they see they might be tempted to leave the system. So, that is the phenomenon that we are trying to see the impatient capture.

- Impatience is as important as arrivals and departures. Gives indication on what the customers feels so that corrective actions can be taken.

- We look at simple settings but applicable to more general settings, which we are not going to see any more from now onwards.
  Even when you look at the more general Markovian model or you know semi-Markovian model wherever it is because this itself becomes a very large class of models that, any model that you have you can always bring in an impatient component inside. You can see how that impatient component comes in that is all you can observe, and that is applicable across different models. That you have to always keep in mind because we are not going to revisit in an explicit manner the impatient behaviour in queues later on, even here also, we will see how one can handle that is all we will see. But we have ah three forms of three basic forms of impatience.

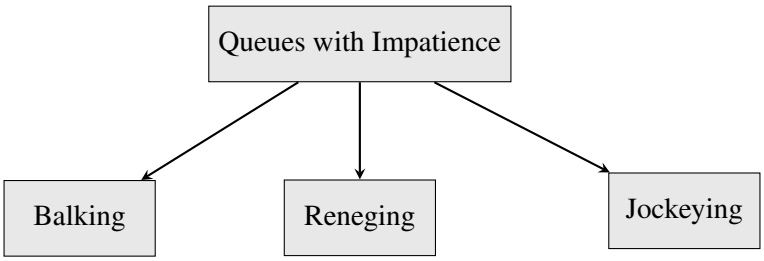


Figure 1: Three Impatience Forms

First, we will talk about **"balking"**. So, we let us take an $M/M/1$ model with balking. So, you have already specified the model $M/M/1$. So, it can be applicable to $M/M/c$; it can be applicable to other forms, finite server, so anything like you can make, you can always incorporate this balking. But what we are seeing is a simple scenario.

- The arrival become discouraged when the queue is long and do not wish to wait.

  So, basically, the balking here, we are looking at the queue size, and the queue size determines the balking nature of the arriving customer.

- Even $M/M/c/K$ can fall into this category (all customers have exactly the same discouragement limit – a rare phenomenon!)

- An approach to balking is to employ a series of monotonically decreasing function of the system size multiplying the average rate λ. Let b_n be this function so that

$$\lambda_n = b_n\lambda \quad \text{and } 0 \leq b_{n+1} \leq b_n \leq 1, n > 0, \quad b_0 = 1.$$

  ▶ When $c = 1$,

$$p_n = p_0 \prod_{i=1}^{n} \frac{\lambda_{i-1}}{\mu_i} = p_0 \left(\frac{\lambda}{\mu}\right)^n \prod_{i=1}^{n} b_{i-1}.$$

- Useful forms useful for the discouragement function b_n are $1/(n+1), 1/(n^2+1), e^{-\alpha n}$.

- The model with $b_n = 1/(n+1)$ is well-known as 'the discouraged arrivals' queue.
  ▶ The stationary distribution for this model is $p_n = \dfrac{e^{-r}r^n}{n!}, n = 0, 1, \ldots$.
  ▶ This distribution is the same as the one for an $M/M/\infty$ queue, but their transient behaviours are different (one of the reasons for studying transient behaviour!).

  And also though this $p_n = \dfrac{e^{-r}r^n}{n!}$ is the case, the other performance measures also would be different; you would see what will be L, L_q, W, W_q in each of these two models itself, though $p_n = \dfrac{e^{-r}r^n}{n!}$ remains the same. You would see that they might be different as well, that is what you know. And there is a queueing happening here whereas, in that M M infinity, there is no queueing happening does not queueing does not happen. So, you also have to remember the differences. So, this distribution alone does not tell you everything; the nature of the system also you have to keep that in mind.

- Customers may also be discouraged based on their estimated time of waiting. A plausible balking function based on expected waiting time could be

$$b_n = e^{-\alpha n/\mu}$$

  [n people in the system implies an estimate of average waiting time is n/μ, if we know μ]

  So, depending upon this one particular form which is basically based upon $\frac{n}{\mu}$, not just on n. You want to know μ because, see the number of customers in the system alone does not play a role, it could be the fact that what is the expected waiting time that may play a critical role. So, that is why b_n should also incorporate μ in some form. If you think so, then $b_n = e^{-\alpha n/\mu}$ could be one function, or you can think about many other functions.

  ▶ $M/M/1/K$ is a special case of balking with $b_n = 1$ for $0 \leq n \leq K-1$ and 0 otherwise.

The second form of impatience is **"reneging"**.

- Customers may join the queue but retain the right to renege if their estimate of their total wait is intolerable. So, after some time, they may decide to leave the system.

- Consider a single-channel birth-death model where both reneging and the simple balking (considered earlier) exist, which gives rise to a reneging function $r(n)$ defined by:

$$r(n) = \lim_{\Delta t \to 0} \frac{P\{\text{unit reneges during } \Delta t \text{ when there are } n \text{ customers present}\}}{\Delta t}, \text{ with } r(0) = r(1) = 0.$$

- This new process is still birth-death, but with the death rate $\mu_n = \mu + r(n)$. Thus

$$p_n = p_0 \prod_{i=1}^{n} \frac{\lambda_{i-1}}{\mu_i} = p_0 \lambda^n \prod_{i=1}^{n} \frac{b_{i-1}}{\mu + r(i)}, \qquad n \geq 1. \qquad \text{[And, } p_0 \text{ in usual manner]}$$

- A good possibility of reneging function $r(n)$ is $e^{\alpha n/\mu}, \quad n \geq 2$

So, this is reneging behaviour. So, you see, where we are tweaking the birth and death rates, if it is balking in the arrival rate, if it is reneging is in the death rate. Reneging is this particular concept that he is in the queue and then he leaves, balking at the time of arrival he decides to join the queue or not. The other phenomenon is this **"jockeying"**.

- Moving back and forth among the several subqueues before each of several multiple channels.
- This phenomenon is quite interesting and clearly applicable to numerous real-life situations, say in an airport or in a railway station or in a bank or in many places wherever you go or in supermarket wherever you go.
- But jockeying is analytically difficult to pursue very far, especially when we have more than two-channels, since the probability functions becomes too complicated and the general concept becomes hazy.

These are the three forms of impatience that we have here. Now, is there any more birth-death queueing models available for us to analyze?

- Remember that what we have seen so far is a very small quantity of queueing models that can be modelled by a BDP.
- Though we have not done, detailed analysis can be done with respect to each of the models considered, depending on the requirement.

  We see, like, sometimes we consider the blocking probability we did not bother about the number in the system in other particular system and so on. But even if you want to study elaborately, that is what is required; if you have a particular system in hand and you want a thorough understanding of the system, then you have to look at every aspect of it in a thorough manner; that is, you have to do. We just showed you how to derive the performance measures. Now once you have the performance measures in hand, each of these performance measures, you can analyze them with respect to each of these input parameters to see their impact on that particular performance measure; that is how the performance analysis needs to be done. But that depends upon the requirement, but it is similar; there is no difference.

- In general, such modelling framework is applicable whenever the arrivals and departures happen one at a time.

- We need to figure out what is the birth rate λ_n and the death rate μ_n and, once done, analysis can be carried out along similar lines.

  So, you have to observe from the real system what could be the functional form of these λ_n's and μ_n's, which would best represent the given system's arrival and departure. So, once you figure out that, then you have the model in hand, and you can do the analysis. So, depending upon each of these λ_n's and μ_n's, you have one model. So, the number of birth-death queueing models in that collection is very large. So, the rates could be a linear, quadratic, polynomial, exponential, or rational function; there could be many different ways we function that you can extract from the reality, which could model the real situation.

- More complex rates may not give us an analytical solution, but approximate solutions can be obtained or numerical/simulation procedures can be employed.

Numerical procedures are always possible because you have the standard numerical procedures for, you know, using these BDP's. So, you can always apply the numerical procedures no matter how complex the rates, lambda n's, and mu are mu n are even in the case of BDP's. So, to get the stationary distribution, mean number in this once you get stationary distribution then you get the mean number and all other parameters easily from that. So, one can employ numerical procedures or simulation techniques even in certain cases if it is required to get the solution. So, it is possible to handle it may not be analytically, but the models the class, the number of models is very large. We have given, based upon the birth-death process what are the type of queueing models that can we can handle in this system models that we have considered so far, which we call birth-death queueing models. The more general Markov and queueing models are what we might take it up later. So, we will stop the birth-death queueing models discussion at this rate. What is left is the transition analysis of such a thing which we will talk about in the next lectures.
Thank you, bye.