Introduction to Queueing Theory

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

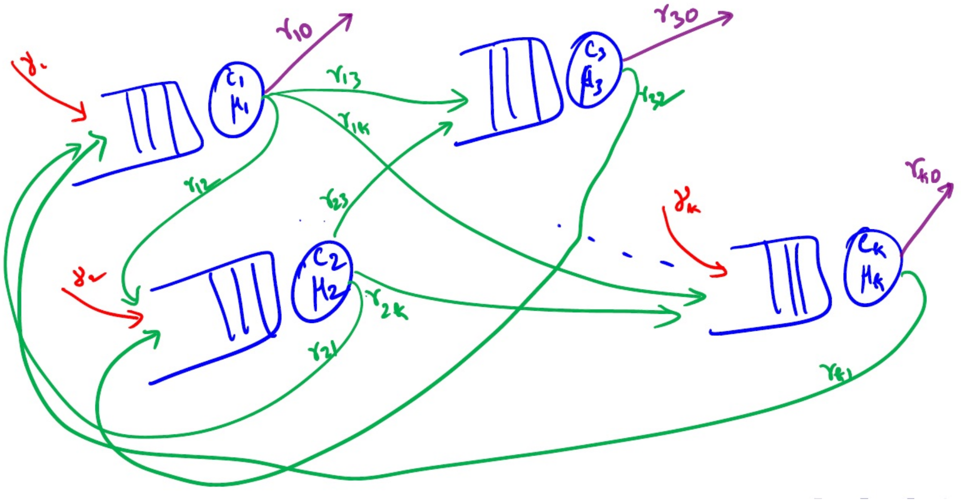
Lecture - 31

Waiting Times and Multiple Classes in Open Jackson Networks

Hi and hello, everyone; what we have been seeing was the idea of what is called an open Jackson network. Let us recall the description that

- A network of k service nodes.
- Arrival at node i according to a Poisson process with rate γ_i.
- Service rate (exponential) at node i is μ_i, with c_i servers at node i.
- Routing probability is r_{ij} (independent of the system state), with r_{i0} denoting the probability of exiting the network from node i.
- No limit on queue capacity at any node (no blocking).

And this is a general idea that you would have seen repeatedly we are talking about. The setup of the network is what something looks like



Then we derived I mean, or we stated the global balance equation. And then, we gave the solution to that as $p_{\bar{n}} = p_{n_1,n_2,\ldots,n_k} = (1-\rho_1)\rho_1^{n_1}(1-\rho_2)\rho_2^{n_2}\ldots(1-\rho_k)\rho_k^{n_k}$ in the case of a single server model. So, we saw that the network acts as if it is an independent $M/M/1$ queue, though really it is not really the case. So, we just verified the solution, and we have obtained the performance measures for this single server at each node open Jackson network, which is basically will be given by because of the form of distribution. Not just because they are $M/M/1$ in the real sense, but because of the form of distribution that you are getting. So, the L_i can be given by $\frac{\rho_i}{1-\rho_i}$, and W_i is by Little's law. Then the expected total number of customers in the network is $\sum_{i=1}^{k} L_i$. And the expected total wait in the network for any customer before it finally departs is basically $W = \dfrac{\sum_i L_i}{\sum_i \gamma_i}$, which is basically the Little's formula for the entire network.

There are other performance measures that one can define, but let us not worry about that too much; so, this is what is the performance measure that might work. One other performance measure that is sometimes used is the number of visits to a particular node because there is feedback and so on; so, how it is any way that is one can look at. Now, what we have seen is an open Jackson network, the restriction that we had was the single server network. Now, this particular case can be extended to the multi-server case without any difficulty.

- **Multi-Server Case:** The results of the single-server at each node can be generalized to open Jackson networks with c_i channels at node i. The joint distribution of number in the network will still be given in a product form as:

  $$p_{\bar{n}} = p_{n_1,n_2,\ldots,n_k} = \prod_{i=1}^{k} \frac{r_i^{n_i}}{a_i(n_i)} p_{0i} \qquad (r_i = \frac{\lambda_i}{\mu_i})$$

  where $a_i(n_i) = \begin{cases} n_i!, & n_i < c_i \\ c_i^{n_i - c_i} c_i!, & n_i \geq c_i \end{cases}$ and p_{0i} is determined such that $\displaystyle\sum_{n_i=0}^{\infty} p_{0i} \frac{r_i^{n_i}}{a_i(n_i)} = 1.$

  So, it is exactly the same one can prove it in such generality, but the thing is that in the network, the balance equation would be slightly complex up to c_i and so on; that is all. But, it is possible that one can show that this is also the case, just like for the single-server case, which was very easy to show also.

- The network again acts as if each node were an independent $M/M/c_i$.

- The performance measures can be obtained in a similar matter as that of the single-channel network, like it will have, the only thing is L_i, W_i, W expressions would vary.

  But otherwise, this remains the same even if there are c i servers in each node; that is what would be the case that one can easily generalize to the multi-server case; so, these are the number in the system.

Now, in any queuing system, we have also seen the other quantity, which we have already obtained through the sojourn times in a way. W gives you the sojourn time in the network; W_i is the sojourn time in the node i. We have already obtained the mean performance measures. Now, what about the waiting time distributions that you can think ok and related to that is the output processes.

- Though it is tempting to conclude that the waiting time distribution at a node should be the same as that of $M/M/c$, this is not necessarily true.

- Recall that in $M/M/c$, we relied on the fact $p_n = a_n$ which relied on Poisson input.

- Here the arrivals to nodes are not Poisson in general (because of feedbacks).

  Of course, in a series network, they will be Poisson, in again in a series network also like you would see like what would happen to the waiting time when you make more than two we will just discuss.
  But our feed-forward network they will be Poisson, but if there is feedback, then the input to each node is not a truly Poisson one. And it is very difficult then to talk about waiting time distribution because the waiting time distribution then will become dependent; it is not independent because suppose if there is a feedback, suppose we assume that there are two nodes the first time the customer comes. Now, if you look at the time customer arrives at the system and the time he will exit the system after getting the service, this will be much like the earlier one for this tagged customer. But, suppose if he comes back to this comes back again with feedback, suppose if he comes back again to that node, then what happens. Then the time how long he has to pass through this would, obviously, depend on his previous time, not just on his, but even other customers in front of him how much they have spent, how long he has been. In the first sojourn time, what was how long, because when the second time when he comes he would see in front of him some customers who have actually arrived from outside before him. And some customers who would have arrived, because of feedback they are in front of him or some other customers who arrived from outside after his arrival he would see both kinds of customers. Then what happens is how long he spent the first time when going through the system. So, that would determine how many would be these numbers. So, there is a dependency one can easily intuitively; also, you can visualize how it will happen. So, they are not going to be independent things are complex, and because of that, waiting time distributions will turn out to be difficult.

  Not just that, even in a series, suppose in this or a feed-forward network, suppose if it is not a single server, but multiple servers even in a series that two nodes assume series, but multiple servers are there. Then like, what would happen is that; say, the customer who arrives first will get into the service, the second customer he will get into the second server, and so on. Once all servers are busy in node one, then the other customers would wait.

  But, when they go to the second node, this sequencing may be broken; why? Because out of this customer, anyone can complete the service at the earliest. So, the customer who arrived later might complete the service earlier with the server, and then he moved ahead of another customer who actually arrived ahead of him in the first node. So, there is a bypassing this also poses a problem when you actually want to see how in order to determine the waiting time in this case. So, these are two issues; feedback and bypassing; both these it has been shown with even very simple examples, I mean as early as in the 70s itself. In the 1970s itself, like with simple examples, it has been shown that these are not really very simple. Even in a very simple situation, this bypassing in the case of multi-server systems, feedback in the case of a general Jackson network, can cause and actually cause complexity problems with trying to determine what is the sojourn time. Also, only for very simple systems, or feed-forward systems, with a single server, you can really talk in an easier way about the question about what is the waiting time distribution. But beyond that, things become very complex because of how this behaves; as you can see, it is very difficult to gauge, and hence this poses a problem. So, essentially what we are pretty much saying is that nothing much can be said about the waiting time distribution in a concrete way, even in these very simple settings. Now, I forgot about going beyond this, but mean value still holds because of Little's law. So, the mean value results are true; they satisfy Little's law; it can be obtained, but if you want the distribution of waiting times, things are not going to be that easy.

- One may also be interested in output processes from individual nodes.
  ► For series or feedforward networks, as we have seen, the flows between nodes and to the outside are truly Poisson.
  ► Feedback destroys Poisson flows, but Jackson's solution still holds.

But the feedback there is no Poisson, then you cannot really if the input itself is not Poisson then what is going to be the output it is very difficult to characterize. So, the characterization of this departure process that might be of also interest in some situations is again complex because of feedback. For waiting time, the culprit is feedback and bypassing, but here in the output process, it is really the feedback that is causing trouble. Otherwise, one can characterize in a way what is the output process or the departure process and how it happens.

Despite these two drawbacks with respect to waiting time distribution and output processes, this Jackson network is still quite popular and used. Because, as you have seen, the system size results are quite neat and clean, though you have a network, you can look at each node individually. Suppose the network satisfies certain properties, then you can look at each node individually, and you get the corresponding quantities and pull together these quantities to get the system-wide or network-wide measures that you are looking for. In that sense, this is quite useful, quite neat, clean, and very much useful; of course, with the assumption that we have these Poisson exponential assumptions, we cannot do away with this in the case of the Jackson network. So, with that; so, this is whether it is a computer system or communication system or inventory system, supplied networks, or production systems like these are quite useful in determining various quantities of interest in such situations. So, that is the part about the open Jackson network that we have seen. Now, let us look at a very simple three-node call center again taken from the text that you can easily see; what is the description of the system?

Example. *[Three-Node Call Center]*

- Calls arrive in a Poisson fashion with a mean rate of 35 per hour at a three-node telephone system of an insurance company.
- Upon calling, there are two options: 1-for claims and 2-for policy service. The caller's listening, decision and button-pressing time is exponential with a mean of 30 seconds.
- Only one call at a time can be processed and other calls wait in queue.
- Estimates suggest that 55% of the calls are related to claims and 45% are policy service calls.
- The claims processing node has 3 parallel servers and policy service node has 7 parallel servers, with both following exponential service time distributions with mean of 6 minutes and 20 minutes, respectively.
- All buffers in front of nodes can hold as many calls as come into the queues.

- About 2% of the customers finishing claims go on to the policy service and 1% vice-versa.

- What is the average queue sizes in front of each node and the total average time a customer spends in the system?

  So, here queue size means the system, not just the number in the queue; it is the number in the system that is what you look at; what is the total average time a customer spends in the system? So, that is what your interest is.

Example. *[contd. . .]*

- With 2 denoting the claims node and 3 the policy node, the routing matrix R is

$$R = \begin{pmatrix} 0 & 0.55 & 0.45 \\ 0 & 0 & 0.02 \\ 0 & 0.01 & 0 \end{pmatrix}$$

And, $\gamma_1 = 35/h,\ \gamma_2 = \gamma_3 = 0, \quad c_1 = 1,\ \mu_1 = 120/h, \quad c_2 = 3,\ \mu_2 = 10/h, \quad c_3 = 7,\ \mu_3 = 3/h$

- For solving the traffic equations, we have

$$(I - R)^{-1} = \begin{pmatrix} 1 & 0.5546 & 0.4611 \\ 0 & 1.0002 & 0.02 \\ 0 & 0.01 & 1.0002 \end{pmatrix} \text{ and hence } \boldsymbol{\lambda} = \boldsymbol{\gamma}(I - R)^{-1} = (35, 19.411, 16.138).$$

Then the offered loads are: $r_1 = \dfrac{35}{120} = 0.292, r_2 = \dfrac{19.411}{10} = 1.941, r_3 = \dfrac{16.132}{3} = 5.379.$

That means that the customer spends about 17 minutes in the network before he leaves the system; that is what you are getting.

Example. *[contd...]*

- Using the formulae from $M/M/c$, we have

$$L_{q1} = 0.120, \quad L_{q2} = 0.765, \quad L_{q3} = 1.402,$$
$$L_1 = 0.412, \quad L_2 = 2.706, \quad L_3 = 6.781.$$

  Thus, the total system L is

$$L = 0.412 + 2.706 + 6.781 = 9.899$$

and hence $W = \dfrac{9.899}{35} = 0.283h \approx 17$ *minutes.*

So, basically, what do you need to do? You need to figure out what are the nodes, how many servers, what are the arrival rates externally, what are the service rates internally, and the routing matrix. Once you have figured it out for any network that you can have here, then it is just then the routine calculations that you need to get the performance measures; so, other examples would also be similar. Again basically, these are the parameters that are what you need to determine from whatever description; if at all anything is there directly, parameters are there fine.

Now, this can also be extended in many ways, but we will just see this one extension which is very simple, which is basically when you have multiple customer classes. But here, we are assuming that all the customer classes have identical service time distribution and wait in the same FCFS queue. They do not have individual queues; they have this same FCFS queue, that is what we are assuming. Under that, this can be generalized to these multiple server classes with this same idea that you have here. So, what you have then

- A customer of one type has a different routing probability matrix than a customer of another type; that was the purpose of multiple classes, each one is designated. Now, you have to solve the traffic equation separately because each one has a different routing probability matrix.
- Solve the traffic equations separately for each customer type and then add the resulting λ's.
- Let $R^{(t)}$ be the routing probability matrix for a customer of type t $(t = 1, 2, \ldots, n)$.
- Solve $\boldsymbol{\lambda}^{(t)} = \boldsymbol{\gamma}^{(t)} + \boldsymbol{\lambda}^{(t)} R^{(t)}$ to get $\boldsymbol{\lambda}^{(t)}$ for each customer type t. Then, $\boldsymbol{\lambda} = \sum\limits_{t} \boldsymbol{\lambda}^{(t)}$.

  Since there is a single queue, you do not need to segregate the server to serve the individual classes because a single queue and all servers are identical. So, that kind of result still holds; so, it is still $M/M/c$ because of that.

- Noting that all customer types have the same average waiting time, since they have identical service time distributions and wait in the same FCFS queue, the average waiting time at each node can be obtained via Little's law. Similar is the case with average system sojourn time.
- We can also obtain the average system size for customer type t at node i as

$$L_i^{(t)} = \frac{\lambda_i^{(t)}}{\lambda_i^{(1)} + \lambda_i^{(2)} + \cdots + \lambda_i^{(n)}} \, L_i$$

So, let us see the same example that we have just talked about about the 3 node network of an insurance company's call center.

Example. *[Three-Node Network - revisited]*

- What we have done there is assume that 2 % of the calls after finishing the claims process go to service and 1% from service to claim.
- Recall the previous example in which it was implicit that a customer can revisit the previously visited nodes, and this is not realistic in the given scenario.

- The way around is: Customers who first go to claims are $type-1$ customers and customers who first go to policy service are $type-2$ customers. Then the two routing matrices are:

$$R^{(1)} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & .02 \\ 0 & 0 & 0 \end{pmatrix}, \qquad R^{(2)} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & .01 & 0 \end{pmatrix}$$

- Since 55% of the arrivals are $type-1$ (the rest $type-2$), we have $\gamma_1^{(1)} = 19.25$ and $\gamma_1^{(2)} = 15.75$.

- *Solving the traffic equations simultaneously, we get*

$$\lambda_1^{(1)} = 19.25, \quad \lambda_2^{(1)} = 19.25, \quad \lambda_3^{(1)} = 0.385$$
$$\lambda_1^{(2)} = 15.75, \quad \lambda_2^{(2)} = 0.1575, \quad \lambda_3^{(2)} = 15.75$$

So, you just use $L_i^{(t)} = \frac{\lambda_i^{(t)}}{\lambda_i^{(1)}+\lambda_i^{(2)}+\cdots+\lambda_i^{(n)}} L_i$ as a proportion because $\lambda_i^{(1)} + \lambda_i^{(2)} + \cdots + \lambda_i^{(n)}$ is the total mean flow rate in node i, and $\lambda_i^{(t)}$ is of the tth class. So, the proportion of this L_i would give you the mean number of customers in the system for the tth class in ith node that is precisely this; you can obtain it here. There are other generalizations that are possible, but things will become complex quickly. So, this is what we can cover with respect to the open Jackson network; again, the Markovian network, but with the open Jackson network is what we have covered so far. So, our discussion of this open Jackson network will end here.
Thank you, bye.