Introduction to Queueing Theory

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

Lecture - 32
Closed Jackson Networks

Hi and hello, everyone; let us continue our discussion on Queueing Networks that what we have been doing so far. We have mainly confined ourselves to the Jackson network, which means that you have a Poisson process arrival, and you have exponentially distributed service time in each node; that is what you know we are restricting to. And, even in that the general Jackson network we have considered so far only up to open Jackson network. What we will do next in our analysis of queueing networks is the Closed Jackson Network. So, what is a closed Jackson network? In a general Jackson network, if we restrict the conditions that there are no external arrivals which means all $\gamma_i = 0$, and there is no departure from the system, which makes it as $r_i = 0 \; \forall i$, which means no customer can leave from any node. Then that particular Jackson network with $\gamma_i = 0$ and $r_i = 0 \; \forall i$ is what is called as a closed Jackson network, another commonly used terminology for such a network is the Gordon-Newell network. Because these are the people who have actually like worked on this closed network concepts; basically from the framework within that is Jackson network thing but in a closed network scenario.

So, that is why in many books, it will be simply referred to as the Gordon-Newell network; that means that it is basically a closed Jackson network. We have already mentioned in our earlier lecture that a simple example of a closed queueing network is the classical finite source queueing model or machine repairmen model, which is a closed network with two nodes, one representing the operating machines and the other being the repair facility. And, there is a total number of machines that is what we call as customers which are circulating within these two nodes; they will be either here in operating condition or in the repair facility. So, this particular thing is a very simple example of a closed queueing network.

So, here what we have is for $i = 1, 2$, $j = 0, 1, 2$ in the general Jackson network framework. If I take $r_{12} = r_{21} = 1$ and all other r_{ij}'s are equal to 0, then what you get is basically the version. We can write down this finite source queueing model as if it is a closed queueing network with these parameters. So, that is a typical simple example of a closed queueing network. Of course, it also has many other applications in a manufacturing production system or in multiprogramming systems or networks. So, basically, what you have here in a closed network, because there is no external arrival, and also there are no customers who leave the network to the outside world. So, there is a fixed number of customers who circulate between the different nodes continuously, moving from one node to the other and getting routed from one node to the other depending upon what is the routing scheme that we have. And, no new customers arrive, and also no customers leave the system. This is a single class, is what typically we might assume. Suppose, if it is a multiple class, then we define while defining the closed queueing network itself that if a network is closed, it has to be closed for all classes in the network. It is not just that if one class is closed for another class is open mean class means for one class of customers the what network acts as if it is a closed queueing network. In the other for other particular class, then if it acts as if it is open, it is really a mixed queueing network is what then you would have. But, here, closed means if you have multiple classes, then it means that for all the classes, the network is closed. So, such a network as we are seeing is basically equivalent to a finite source queueing system with N items, and the items continuously travel inside the network.

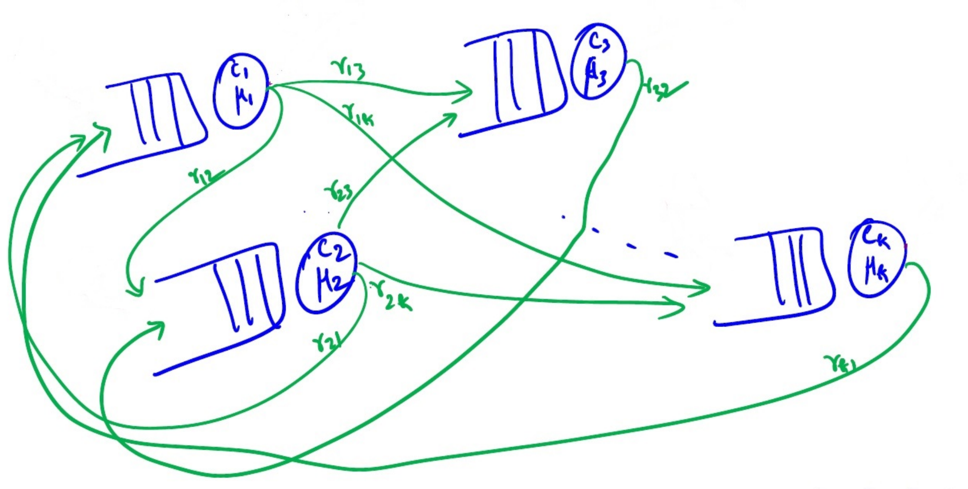
Anyway, maybe when we see one or two examples, you will understand more, but a typical simple example is there in front of you, which is the machine repairmen problem. So, what you have now is basically a setup for a closed Jackson network which is as follows.

- We now have a setup for closed Jackson networks as:
  - A network of k service nodes.
  - Service rate (exponential) at node i is μ_i, with c_i servers at node i.
  - Routing probability is $r_{ij},\ 1 \leq i, j \leq k$ (independent of the system state).

    Because r_{i0}, that extra thing that we had in case of an open network, is no longer applicable here. So, they will sum they will I mean for each j I sum overall j of r_{ij} then that must be equal to 1, that is what you will get here whereas, that may not be the case in the open network because there is one additional node which we call as an external node, which is 0. So, r_{i0} maybe some nonnegative quantity, a strictly positive quantity that is also possible.
  - There are N customers in total in the network. No new customers arrive to the network and no customer departs from the system.
  - No limit on queue capacity at any node (no blocking).

    If there is, then again, all these blocking issues will come, but anyway, we are not going to look at that. But, that is possible that you can easily extend these ideas to such scenarios as well, but again the complexity will increase.
- We have a Markovian system and the state of the system can be described via N_i's, where N_i is the random variable for number of customers (in queue and in service) at node i in steady-state.
  ► Note that $N_1 + N_2 + \cdots + N_k = N$.

Remember, there is a fixed number of customers who move from one node to the other as per the routing situation, and then they will be somewhere there in the system. So, that means N must be there somewhere. So, this sum of all these N_k's would then correspond to the quantity N; that is what you have here.



So, this is exactly the same diagram as what we had for an open network; what you do not have is, as you might see, there is no external arrival here, and there is no departure from here, and so on. So, it is exactly the same as the previous network diagram; there is no change I have made; I have made only thing changes that the external arrivals and departures I have removed. So, it becomes a closed queueing network.

So, now there is some number of customers. So, there maybe you can assume that there are a total of N customers. So, there could be a total of N customers in the network. So, they will be moving along these different networks; there are k nodes are in this network. So, they will move from as per the routing thing that you have; now, the only thing you have to notice is that the sum of $r_{11}, r_{12}, r_{13},$ and so on r_{1k}; this sum will be equal to 1. So, this will be the routing matrix for a closed queueing network is a proper stochastic matrix. Now, if I give you a routing matrix and ask whether this routing matrix can correspond to an open network or closed network, in case I have open, at least one node should be leaving out, so; that means there is at least one row which is not summing to 1. In the case of a closed network, that cannot happen; all the rows must be equal to 1. So, that is what you could think of it as a property of routing matrix as well anyway.

So this is what is a typical closed queueing network that you have here; there are customers who move from node 1 to node 2, or node 3, or node k, and from node k, they move to node 2, and from node 3 they move to node wherever it is, it is I think node 3 from node 2 it is coming and so on. So, this is what is a typical closed queueing network that you can visualize. So, it is exactly the same as open, no external arrivals, no departures, that is it.

- As usual, we want the joint distribution $P\{N_1 = n_1, \ldots, N_k = n_k\} = p_{n_1,n_2,\ldots,n_k}$ from which we can obtain other required quantities.
- The notation for the k-component vector is as follows:

State	Simplified Notation
$n_1, n_2, \ldots, n_i, \ldots, n_j, \ldots, n_k$	$\overline{n}$
$n_1, n_2, \ldots, n_i + 1, \ldots, n_j - 1, \ldots, n_k$	$\overline{n}; i^+ j^-$

- Assume for now that $c_i = 1, \forall i$ (i.e., single server at each node).

- The stochastic balance (global) equation for state $\bar{n}$ with $n_i \geq 1, \forall i$. if you recall the previous balance equation that we have written for an open network, it is the same except that if you put that $\gamma_i = 0$ and $r_{i0} = 0$, you will basically end up with this:

$$\sum_{j=1}^{k} \sum_{\substack{i=1 \\ (i \neq j)}}^{k} \mu_i r_{ij} p_{\bar{n};i^+j^-} \quad = \quad \sum_{i=1}^{k} \mu_i (1 - r_{ii}) p_{\bar{n}}$$

► The above will also hold for the case $n_i = 0$ if we set terms with negative subscripts and terms containing μ_i for which $n_i = 0$.

- Since this network is a special case of a general Jackson network, so we have a product-form solution

$$p_{\bar{n}} = C \rho_1^{n_1} \rho_2^{n_2} \ldots \rho_k^{n_k},$$

where $\rho_i = \lambda_i / \mu_i < 1$ must satisfy the balance equations for flow at each node i, so that the flows into and out of node i are equal.

- Therefore, the traffic equations, if I plug in that quantities, are $\gamma_i = 0, r_{i0} = 0$, then this will actually become

$$\lambda_i = \mu_i \rho_i = \sum_{j=1}^{k} \lambda_j r_{ji} = \sum_{j=1}^{k} \mu_j r_{ji} \rho_j, \quad i = 1, 2, \ldots, k.$$

- As earlier, we assume that the routing matrix R is irreducible and non-absorbing.

Because no node is completely absorbing; anyway, it does not have. Then, of course, the network there is nothing like studying; everything, after some time in a steady-state, will be absorbed into that particular state or node. So, to avoid the triviality like we will assume that this is non-absorbing, but now we know that the total sum is N and the total number of customers is N.

- But, one of the traffic equations is redundant (as $\sum_i \lambda_i$ is fixed). We can set one ρ_i equal to 1, and you can solve for the remaining one; this will get adjusted in the normalization constant. So, that will take care of this; there is nothing to worry about. So, one can set because of the linear dependent system you can, set one of them equal to 1 and solve $\mu_i \rho_i = \sum_{j=1}^{k} \mu_j r_{ji} \rho_j$ to get the solution.

Now, we said that $p_{\bar{n}} = C \rho_1^{n_1} \rho_2^{n_2} \ldots \rho_k^{n_k}$ is what is the solution, and this is also like in the open network case, it is called as Jackson's theorem, which gave that this as the result or the solution of the stochastic flow balance equations. But, here, this is also called as Gordon-Newell theorem because they only showed that, again, this form is of again the same form in this particular case.

- *Exercise: Verify that the product-form solution satisfies the balance equations.*

But in the open Jackson network, we also saw that the C also could be written as a product of terms, the product of $(1 - rho_i)$'s is what then we wrote, and hence, this whole thing actually broke it into the product of marginals.

- For this closed network, C is not a product of terms and must be evaluated from

$$\sum_{n_1+n_2+\cdots+n_k=N} C\rho_1^{n_1}\rho_2^{n_2}\cdots\rho_k^{n_k} = 1$$

$$\implies C = \left(\sum_{n_1+n_2+\cdots+n_k=N} \rho_1^{n_1}\rho_2^{n_2}\cdots\rho_k^{n_k}\right)^{-1} = C(N) \quad \text{(say)}$$

The constant C is written as $C(N)$ to denote the fact that is a function of N. It is also the case that the solution is often written in terms of $C^{-1}(N) = G(N)$ and therefore, the solution is

$$p_{n_1,n_2,\ldots,n_k} = \frac{1}{G(N)}\rho_1^{n_1}\rho_2^{n_2}\cdots\rho_k^{n_k},$$

where $G(N) = \sum_{n_1+n_2+\cdots+n_k=N} \rho_1^{n_1}\rho_2^{n_2}\cdots\rho_k^{n_k}$ is the normalization constant.

We are giving a name specifically because the crux of the problem or the emphasis on a major problem while dealing with closed networks are connected with this $G(N)$. So, we want to specifically give a name which is what is called is the normalization constant, in the case of a closed queueing network is what is $G(N)$ because this is $1/C(N)$ is $G(N) = \sum_{n_1+n_2+\cdots+n_k=N} \rho_1^{n_1}\rho_2^{n_2}\cdots\rho_k^{n_k}$.

So, $G(N)$ is basically $\sum_{n_1+n_2+\cdots+n_k=N} \rho_1^{n_1}\rho_2^{n_2}\cdots\rho_k^{n_k}$ here which is in the case of this single server at each node network, closed queueing network that is what you have. So, in this Gordon-Newell network, that is also G here. This sum where the sum is taken over all possibly n_1 to n_k such that they sum to N is what will give you the normalization condition. Here as we just said that $\frac{1}{G(N)}$ is not factoring into the product of terms involving n_i's or ρ's whatever, and hence

♦ Here, observe that the joint distribution is not a product of marginals.

So, that is why we wanted to have a relaxed or less restrictive definition of product form network. So that we can call $p_{n_1,n_2,\ldots,n_k} = \frac{1}{G(N)}\rho_1^{n_1}\rho_2^{n_2}\cdots\rho_k^{n_k}$, also a product form network, dealing is almost similar.

So, rather than a more restrictive definition where we want the product form network means the product of marginals, but we do not insist here, that is the thing that you can keep in mind. So, that is what this is a solution. So, this is a complete solution; the joint distribution $p_{n_1,n_2,\ldots,n_k}$ is given by $\frac{1}{G(N)}\rho_1^{n_1}\rho_2^{n_2}\cdots\rho_k^{n_k}$.

Now, once I have this, then I can get other quantities connected with this network. Suppose, if I want say what the probability that the node 1 is empty or some number of nodes are empty, or the whole network is empty, or certain probabilities. What is the probability that the number of customers at any node is more than 10 when the total number of customers in the network is, say, 100 and so on. Like anything probability expected measures or anything that you are normally interested to get, you can get from this joint distribution as we have done earlier in this particular case.

So, there is nothing peculiar about that. So, our main interest was to obtain $p_{n_1,n_2,\ldots,n_k} = \frac{1}{G(N)}\rho_1^{n_1}\rho_2^{n_2}\cdots\rho_k^{n_k}$, and you obtained for all n_1, n_2, n_k greater than or equal to 0 such that $n_1+n_2+\cdots+n_k = N$ is what is the scope or the space for the support for this N-dimensional random variable.

- The single channel at each node closed network can be extended to c_i-servers at node-i. The solution becomes

$$p_{n_1,n_2,\ldots,n_k} = \frac{1}{G(N)}\prod_{i=1}^{k}\frac{\rho_i^{n_i}}{a_i(n_i)},$$

$$\text{where} \quad a_i(n_i) = \begin{cases} n_i! & n_i < c_i \\ c_i!c_i^{n_i-c_i} & n_i \geq c_i \end{cases} \quad \text{and} \quad G(N) = \sum_{n_1+n_2\cdots+n_k=N}\prod_{i=1}^{k}\frac{\rho_i^{n_i}}{a_i(n_i)}.$$

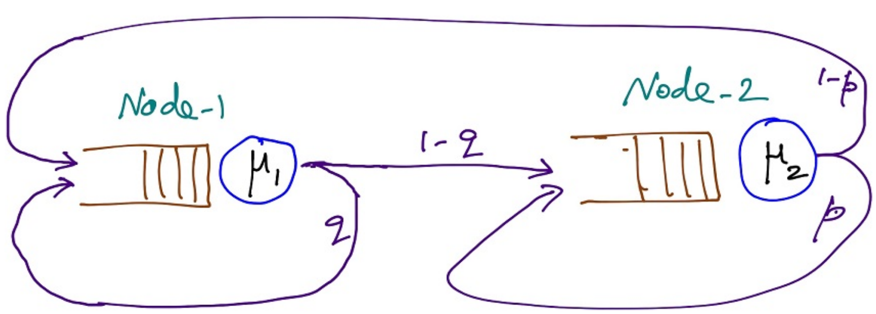
So, this extension to c_i-servers, I mean, if you want to directly write this c_i-servers case in general, the balance equation be a little complex and other things a little complex. For understanding, I think we can start with a single server, but it is very much true, the exact similar thing only thing this expression varies that is all. Then everything holds true for c_i-servers at node i; that is what we are writing it here.

So, for c_i, then again, this comes from your $M/M/c$ model. So, that p_0 factor, the normalization alone we do not do, but otherwise, this factors coming from this $M/M/c$ model that what you have here like the earlier case. So, this multi-server case one can easily handle; that is what you are seeing it here.

Now, let us look at examples.

Example. *[Two-Node Closed Queueing Network]*

- Consider the following two-node single-server-at-each-node closed queueing network with a total of M customers in the network*:*



let us consider this two-node network, node 1 and node 2, and there is a single server at each node. And this node, the service time here is exponential with parameter μ_1, and at node 2, the service time is exponential with parameter μ_2. And this is a closed queueing network, and there is a total of M customers in the network; there are totally M customers in the network.

So, some number of customers would be there in this node. So, maybe with the server as well as queueing and some number of customers will be there in this node maybe with the server and also at the queue. And, after completing the service at node 1, it can come back to node 1, again, which means immediate feedback with a probability of q; with a probability of q or with a probability of $1-q$, this will move to node 2 to get service in node 2, that is what happens after service completion at node 1.

And, after service completion at node 2, it can come back to node 2 again immediately with probability p, which means he will join at the tail end. So, it is basically FCFS is what we have in mind even for this situation or with the probability $1-p$, it can come to node 1 with probability $1-p$. So, in both the nodes, there is immediate feedback with probability q for node 1 and p for node 2, or with probability $1-q$ and $1-p$, it will move to the following node; you can think about this in the circular fashion as well. So, there are only two nodes very simple setup network; you have the infinite capacity here, you have the infinite capacity here for queueing, exponential servers; this is what you have here. So, what is our interest? The steady-state joint probability distribution is what we want to compute.

Example.

- The steady state joint probability distribution is given by

$$p_{M-m,m} = \frac{1}{G(M)}\rho_1^{M-m}\rho_2^m, \qquad m = 0, 1, 2, \ldots, M.$$

Because that is what pretty much you would have had if you had to write $p_{n_1,n_2,\ldots,n_k} = \dfrac{1}{G(N)\rho_1^{n_1}\rho_2^{n_2}\ldots\rho_k^{n_k}}$.

We must find ρ_i from $\mu_i\rho_i = \sum_{j=1}^{k}\mu_j r_{ji}\rho_j$. Here, the routing matrix R is $R = \begin{pmatrix} q & 1-q \\ 1-p & p \end{pmatrix}$ and hence the traffic equations becomes

$$\mu_1\rho_1 = \mu_1 q\rho_1 + \mu_2(1-p)\rho_2$$
$$\mu_2\rho_2 = \mu_1(1-q)\rho_1 + \mu_2 p\rho_2$$

Since these equations are linearly dependent, as we know already, we can set one of the ρ_i's to 1 and solve for the other.

Set $\rho_1 = 1$. Then, from the second equation, we get $\rho_2 = \dfrac{1-q}{1-p}\dfrac{\mu_1}{\mu_2}$.

Example. We thus have the steady state solution for the closed network as

$$p_{M-m,m} = \frac{1}{G(M)}\rho_2^m = \frac{1}{G(M)}\left(\frac{(1-q)\mu_1}{(1-p)\mu_2}\right)^m, \quad m = 0, 1, 2, \ldots, M,$$

where the normalizing constant $G(M)$ is given by $G(M) = \sum_{m=0}^{M}\rho_2^m = \dfrac{1-\rho_2^{M+1}}{1-\rho_2}$.

Performance Measures:

The probability that node-1 is busy equals $1 - p_{0,M} = 1 - \dfrac{1}{G(M)}\rho_2^M = \dfrac{G(M-1)}{G(M)}$.

Similarly, the probability that node-2 is busy equals $1 - p_{M,0} = 1 - \dfrac{1}{G(M)} = \rho_2\dfrac{G(M-1)}{G(M)}$.

The average number of customers in node-2 and node-1 are

$$L_2 = \sum_{m=0}^{M} \frac{m\rho_2^m}{G(M)} = \frac{\rho_2}{G(M)} \sum_{m=0}^{M} m\rho_2^{m-1} = \frac{\rho_2}{(1-\rho_2)^2 G(M)} \left[1 - \rho_2^{M+1} + (1-\rho_2)(M+1)\rho_2^M\right]$$

$$L_1 = M - L_2$$

So, this is L_1, L_2 likewise because the mean performance measures you can see and then you can obtain and with all those things, like in one can obtain other performance measures. But, point here to notice in a two-node network like this there may be some simplifications, simplified version that you will be able to arrive which can directly give rise to some nice expressions for certain performance measures, that you may look for; because here $G(M)$ is itself coming out to be in a nice form in an explicit form. Otherwise, the problem mainly lies with this $G(M)$ in the closed queueing network. So, in this particular case, it turns out that it is not that easy, you can obtain it very easily, and certain performance measures also another point that you notice certain performance measures are also can be written in terms of G. Of course, G will appear in I mean in the joint distribution, but in terms of the performance measures directly to express that is also certain nice things to have in many cases. So, this is one example that we are seeing; we will continue with more in the next lecture.

Thank you, bye.