Introduction to Queueing Theory

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

Lecture - 35
Cyclic Queueing Networks, Extensions of Jackson Networks

Hi, and hello, everyone. What we have been seeing was this Closed Queueing Network or Closed Jackson Network, otherwise called as Gordon-Newell Network that is what we have seen so far. Now, what we will see next is; that you can consider that as a special class or a specialized version of the closed queueing network. So, what is that? So, this is called as Cyclic Queues or Cyclic Queueing Networks. This, what we are seeing is a special case of a closed queueing network, but this can also be put into an open version actually, but this is more in the case of a closed queueing network that it finds lots of applications. And in fact, it arose in a study by Taylor and Jackson in 1954, when they introduced the concept of this cyclic queue as a model for analyzing the flow of aircraft engines from operation to maintenance and again to operation. Operation, maintenance, operation, you can think of this again something similar to your machine repairmen similar kind. So, this is they use this as a cyclic view, and then they try to analyze and then try to get insights into the system. Of course, since then, like, there has been a lot of work which has been ah done on this. So, this is, what is that then is a cyclic queue or cyclic queueing network, is it is basically a sort of a series queue in a circle, where the input of the last node feeds back into the first node. So, if I want to put it diagrammatically, there is this node 1, which feeds into node 2 and so on, so there is this kth node. So, this comes from here; the arrival comes from here. So, this is the kth node. So, you can call this as I think μ_1, μ_2 and assume that you have μ_k. Then, what happens. So, this comes as an input to this; that is what it is. So, you have a certain fixed number of customers who are circulating in this network in this manner, one after the other that is what we see here. This, after this, they go to queue node 2 and so on up to node k and then, so you can call this as node 1, node 2, node k, it is a very simple setup. So, like series as a special case of the open network. So, this is basically a series in a queue. So, you can, in fact, draw this diagram in a circle like this so that the connection becomes very nice. Here we have to connect it from here to here, but if you draw in the circle like this, it will come naturally to that. So, it is also like called circulant network sometimes this kind of things are also called in that fashion. So, what is the setup?

- If we consider a closed network of k nodes such that

$$r_{ij} = \begin{cases} 1, & j = i+1,\ 1 \leq i \leq k-1 \\ 1, & i = k,\ j = 1 \\ 0, & \text{otherwise} \end{cases}$$

  then we have a cyclic queue.

- Since this is a special case of closed queueing network, the results of closed queueing networks apply here.

One can apply, but since it is a specialized one, so you can always get something more out of that than the general.

- Assume that there is a single server at each node.
- We have the joint system size distribution as

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

- The traffic equations $\mu_i\rho_i = \sum_{j=1}^{k}\mu_j r_{ji}\rho_j$ on substitution of r_{ij}'s give

$$\mu_i\rho_i = \begin{cases} \mu_{i-1}\rho_{i-1}, & i = 2,3,\ldots,k, \\ \mu_k\rho_k, & i = 1. \end{cases}$$

  Many a time, because we know that this is ρ_i is the relative utilization, then $\mu_i\rho_i$ is relative throughput which is relative λ_i's. So, you would find in many situations that people write $\mu_i\rho_i$ as a single quantity as we have done for the MVA algorithm kind of thing. So, that is $\mu_i\rho_i$ together, because whether you are determining ρ_i or $\mu_i\rho_i$ together, it is one and the same. So, that is why you can write it in that form also a traffic equation, but we will keep it in this form.

  Thus, we have

$$\rho_i = \begin{cases} \left(\frac{\mu_{i-1}}{\mu_i}\right)\rho_{i-1}, & i = 2,3,\ldots k \\ \frac{\mu_k}{\mu_1}\rho_k, & i = 1. \end{cases}$$

Now, once I know the rho is, I can put it here.

- That is,

$$\rho_2 = \frac{\mu_1}{\mu_2}\rho_1, \quad \rho_3 = \frac{\mu_2}{\mu_3}\rho_2 = \frac{\mu_1}{\mu_3}\rho_1, \quad \ldots, \quad \rho_{k-1} = \frac{\mu_1}{\mu_{k-1}}\rho_1, \quad \rho_k = \frac{\mu_1}{\mu_k}\rho_1.$$

  Now, if you substitute this here in $p_{n_1,n_2,\ldots,n_k} = C\rho_1^{n_1}\rho_2^{n_2}\ldots\rho_k^{n_k}$. So, to solve this, what we have to do is that we know that the system of equations is linearly dependent, and that is what you know you are coming out here, you see. Everything can be expressed in terms of one, but it cannot be independently determined.

- Since one ρ can be set equal to one, we set $\rho_1 = 1$.

  Then $\rho_2 = \frac{\mu_1}{\mu_2}$, $\rho_3 = \frac{\mu_1}{\mu_3}$, and so on, you will get. So, that is what you will get. Now, once you plug those quantities in $p_{n_1,n_2,\ldots,n_k} = C\rho_1^{n_1}\rho_2^{n_2}\ldots\rho_k^{n_k}$.

- Then the solution is

$$p_{n_1,n_2,\ldots,n_k} = \frac{1}{G(N)}\frac{\mu_1^{N-n_1}}{\mu_2^{n_2}\mu_3^{n_3}\ldots\mu_k^{n_k}},$$

  where $G(N)$ can be computed as earlier (naive or convolution algorithm).

- Multiple-server case can also be treated similarly without much difficulty when you are looking at this cyclic queue.

- As we already pointed out, the machine repairmen problem is really a two-node cyclic queue, one node for the operation, the other node for, as usual for the repair facilities, and then the items will move from one to the other in a cyclic fashion. So, that is basically a two-node cyclic queue.

  It is a very simple network, but many a time, the analysis would help us a lot depending upon the situation that you would here because there are many more specialized results that can be obtained for the cyclic queueing network, that is possible.

Example.

Consider a cyclic queue with two nodes and $N = K$ circulating jobs.

The first node (node 1) has an exponential server with rate λ, and the second node (node 2) has an exponential server with rate μ.

This can be viewed as a special case of the two-node closed queueing network considered earlier.

So, there are two nodes, and after each node, there could be feedback with a certain probability, and it could go to the other node with a certain probability that is what we had there. So, now, if you remove that feedback 1, then what you are going to get is exactly this problem or this example that we are looking at.

So, there in that model, if you are keeping in mind if you substitute these parameters, ($\mu_1 = \lambda, \mu_2 = \mu, p = q = 0, M = K$) there, we consider M jobs, so now, we are putting it as K jobs. This p and q, for self-feedback immediate feedback, which was if you make it 0, it will go to only the subsequent nodes, and with rates as lambda mu in the two nodes, if you substitute then, you are going to get this back. Nevertheless, even without going there, so you can consider these as it is, and you can write the traffic equation.

The traffic equations becomes

$$\lambda\rho_1 = \mu\rho_2$$
$$\mu\rho_2 = \lambda\rho_1$$

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 $\rho_2 = \dfrac{\lambda}{\mu}$.

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

$$p_{K-m,m} = \frac{1}{G(K)}\rho_2^m = \frac{1}{G(K)}\left(\frac{\lambda}{\mu}\right)^m, \quad m = 0, 1, 2, \ldots, K,$$

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

Now, if you substitute $G(K)$ in $p_{K-m,m}$, then I will get a complete expression, but remember here we have assumed ρ_2 is a generic λ/μ, but now we can generalize, be a little bit more specific for two cases; one is where $\lambda \neq \mu$ and when $\lambda = \mu$, ρ_2 is also 1. And hence $\left(\lambda/\mu\right)^m$ is also 1, and hence like you have to appropriate, $(\rho_2)^m$ is 1. So, you will simply get the required quantity $\sum_{m=0}^{K}(1)^m = K + 1$. That is what you are going to get.

Example.

Taking into account both the cases of $\lambda = \mu$ and $\lambda \neq \mu$, we get the complete steady state solution as

$$p_{K-m,m} = \begin{cases} \dfrac{(1-\rho_2)\rho_2^m}{1-\rho_2^{K+1}}, & \rho_2 \neq 1, \\ \dfrac{1}{K+1}, & \rho = 1, \end{cases} \qquad m = 0, 1, 2, \ldots, K.$$

Recall that the steady state system size distribution for an $M/M/1/K$ system is also given by the above expression.

Thus, a two-node cyclic queue may be considered as equivalent to an $M/M/1/K$ system.

If you have an $M/M/1/K$ system with the parameter λ and μ, if you obtained the steady-state number in the system in that $M/M/1/K$ system, then you saw that you obtained exactly $\begin{cases} \dfrac{(1-\rho_2)\rho_2^m}{1-\rho_2^{K+1}}, & \rho_2 \neq 1, \\ \dfrac{1}{K+1}, & \rho = 1, \end{cases}$ $m = 0, 1, 2, \ldots, K,$ where $\rho_2 = \lambda/\mu$.

So here, this one also you are obtaining it to be exactly the same. So that means what? There is an equivalence. These kinds of things you know we have seen already. Say, for example, in the case of Erlang and is connected with the bulk. Erlang n models to the bulk, queues we have made the equivalence. Similarly, for the cyclic ones, the finite space model, one can make equivalence so that the properties can be studied nicely. And again, I would say that you can, if you are interested, you can look into the literature on how these kinds of equivalence have been utilized to study a queueing network or the other way around, whichever way so because these are in a way in some way equivalent system. So, this two-node cyclic queue, as we described here, may be considered as an equivalent to $M/M/1/K$ system. And this might be of some help, as we said when we are trying to analyze the model in some detail in this case. So, that is the example that we have; again, we said that this is considered a closed queueing network. So, you can also think about an open network scenario, but more application-oriented stuff happens only in the closed network context. So, that is why we restrict here.

Again one need not study this separately because you can study as a special as a part of a closed queueing network. Similarly, the open one you can study as part of an open Jackson network, no problem. As we always say, it may be convenient; it may be useful that you study these special systems in more detail to which that end like that might be of some help; so, that is what is about cyclic queues that we are seeing it. Now, we will try to end this discussion on queueing network on what was further happening and the directions in which this has gone ahead. Because what we started when we started we said that we are going to look at within the Markovian framework only, Markovian network, we are still within that. Even within that, there are a lot of extensions, and there are beyond that also there are. Now, if you look at the Jackson network itself, we will see certain extensions of the Jackson network. We are not going into detail, but we will just highlight what are the extensions and so on.

- There have been many directions in which Jackson networks have been generalized (like state-dependent exogenous arrivals, state dependent service, travel times between nodes, etc.)

  It may not be the case that the customer who leaves the previous node; suppose the following node is empty in a way. So, then his service starts the moment he leaves the previous node. So, in that case, what we are

looking is, looking at, suppose if he is going and waiting in the queue, then it does not matter but provided if that server is free, then we are taking it in our model things. As you can understand, if his service, the second service, for the second at the subsequent node, the service starts immediately at the moment his service is completed in the previous one. But that means that you are assuming that there are no travel times between nodes. But in reality, there are some travel times. Now, how can one do? One way is you think the travel time itself corresponds to some kind of node. So, the travel time between one. That you see is a node with multi-server, ample server, infinite server. So, everyone keeps moving without any hindrance, but if there is a hindrance, then again, you have to model appropriately. But without if there is no difficulty if they are moving, then you can think that is corresponding to m infinite server model, and then one can put a certain time duration that will be the travel time for that node. But otherwise, one can explicitly also incorporate the travel times. These are all certain extensions that have happened with respect to Jackson. Jackson himself did say, for example, these state-dependent exogenous arrivals and so on, a few years later been after he gave the first one.

- The most significant and useful form of extension of Jackson network was to the case of multiple classes of customers, considered by Baskett, Chandy, Muntz, and Palacios (1975). These are called BCMP networks.

This is the standard network because it has lots of features, and it can quite very well incorporate the thing you want to do so; if you see in use today, the BCMP network would be the most useful network that you can see. So what is this?

- A BCMP network considers different service disciplines as well as a number $Q(\geq 1)$ of classes of customers. The setup is:
  - k nodes in a closed network with $Q(\geq 1)$ classes of customers.
  - Customers circulate in the network and may change class as they move from one node to another.
  - A customer of class r completing service at node i next goes to node j as a class s customer is denoted by $p_{ir,js}$. Hence, the routing matrix $R = (p_{ir,js}),\ 1 \leq i, j \leq k,\ 1 \leq r, s \leq Q$.
  - The service rate of a customer of class r at node i is μ_{ir}.

- A product-form solution is obtained in a BCMP network for the following cases of queueing disciplines:
  − Processor sharing, which means that the available service capacity is distributed equally among the customers waiting in the system at any point of time. Like internet.
  − ample service, which means any number of servers infinite server case.
  − LCFS with preemptive-resume servicing
  The exogenous Poisson input can be state dependent and the service distributions can be phase-type.

It does not just need to be exponential; that is why we just talked about the rate; we did not talk about the distribution for which this is the rate. It could be even phase-type distribution.

- A product-form solution is also obtained in a BCMP network for c-server FCFS nodes, but with service times of all classes must be IID exponential (i.e., all customer types look alike).

So, it is as if only in that case, they obtain the product-form solution. Of course, there were generalizations again, as happens in any area from that point onwards along these lines. This is one of the most useful, and of course, one can easily write down these assumptions like the traffic equations, obtain a solution and write down the product-form and the normalization constant you can obtain it much like the other queueing networks that we have considered.

So, is not much difficulty there; essentially, once we specify the assumption, anyway, we are not going into that. So, I am just highlighting that there is some such thing called the BCMP network, which is what is the most useful form of queueing network as far as applications are concerned that is what you know you have to take care of.

- Kelly (1975, 1976,1979) considered some other significant generalizations of Jackson networks and that included extensions to most general queueing disciplines. He obtained product form solution with Erlang service distributions too and conjectured for more general distributions (which was proved later).

So, this BCMP network and Kelly's extensions, in fact, Kelly's book reversibility and network; it is available freely online and where he had given lots of work on whatever he has done along with up to that point of time. It is a very good resource if you are looking at something on the network and need more details. So, that is the; these are theoretical developments.

At the same time, as these theoretical developments are happening or the theory develops when the product form solution. We always look for a product-form solution because that makes your life easier. If you have to keep it as it is in the full generality, like what happens in the case of a queueing network with blocking, then the complexity is exponentially very high ratio. So, it is very difficult to handle, but even so, you want to keep a still in product form; even then, you have a lot of issues with computationally when you want to compute, but when you do not have product form, then it is very difficult to do that of course. So, that is why you keep looking for under what generalization I can make so that I am well within the product form solution format.

While this was happening, there was also attention towards efficient procedures or algorithms for obtaining the computational results. Again, in each case, you have to look at the normalization constant, for example, whether it is an open or closed network; whichever form you know you are looking at it, you still have to compute that.

Because the product form, it is even in an open case, we need not assume that it is basically factored into the product of marginals. It is just that its product form. So, still, you will have a normalization constant that you need to compute. Obviously, in a closed queueing network, that is an inevitable thing, but in the open, also like in within that form only like we are calling it as product form network. So, within that, then again, you will have a normalization constant. Now, how you will compute in an efficient way, all that aspects also have to be looked at. Otherwise, theoretically, you are developing, but applicability will be limited if you are not having a procedure for that. So, side-by-side, one has to look at that aspect as well, and that aspect is going on until today. And you can say that the various sections keep coming and then depending upon the necessity to model many real-life phenomena as they evolve.

Like, the more complex situations that come, the more complex model will become, and then you try to analyze that, then you try to see how we can implement it, then computation aspects you always look at. So, this is a never-ending

process. So, it is always on, like in any other field. There is nothing peculiar about queueing networks or queues alone that such kinds of scenarios are happening. So, it is across the board for any research to develop. So, we will end here. Of course, one can also talk about the non-Jackson network as an extension and some other something called as the loss network. See here; we have assumed that all $M/M/1$ type or $M/M/c$ type in each node.
Suppose, if you have instead of this model if you are keeping a loss model, say, for example, the Erlang's loss model; suppose if you are keeping it in as part of nodes, then such a network is called loss networks. Again, it has applications in computer and communication fields such as networks. So, one can talk about this from different aspects and angles, beyond even Jackson's network, but that is for you to explore further.

So, we will stop our discussion here on this Queueing Network. We have just restricted ourselves to the Markovian framework, which considered open and closed queueing networks, basically open Jackson and closed Jackson network, and how one can handle that situation. That is it.
Thank you. Bye.