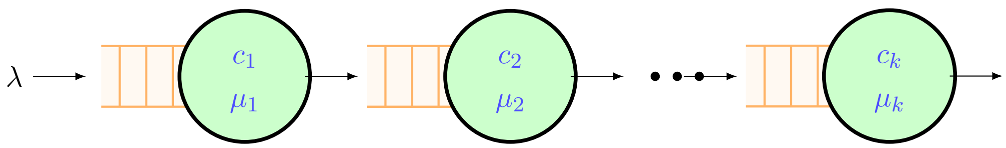
Introduction to Queueing Theory

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

Lecture - 30
Queueing Networks with Blocking, Open Jackson Networks

Hi and hello, everyone. What we have seen in the previous lecture is a series network or a tandem network wherein our quantity of interest which is basically the number in the system at each node was the joint distribution of this, was given by the product of their marginal distributions. So, that is what we have seen. Say, for example, we had this



kind of network that we had in mind for which it was happening that the joint distribution is given in terms of the marginal distribution. So, $p_{n_1,n_2,\ldots,n_k} = p_{n_1}\, p_{n_2}\, \cdots\, p_{n_k}$ is really a truly product form solution even in any sense that you look at it.

So, this a similar thing is possible like we could understand that how this is resulting in the analysis is from the knowledge of Burke's theorem. And it is possible that a similar kind of analysis can be taken forward to a feed-forward network, where also similar things can easily be followed without much difficulty; that is what we have seen. Now, in the situation, we have assumed that all the nodes have infinite capacity, meaning that there is ample space for queuing in front of every node. So, whenever it is passing one node, if it wants to go to another node, in the destination node, there is it can queue if it is not getting the service immediately. So, there is always the ample capacity to hold any number of customers in waiting; in any particular node, that is what we have assumed, and that is what we are going to assume later on also. But if it is not the case, in reality, nothing is infinite; it is always the case that there is a finite amount of space only available between the nodes. So, if that happens, that is what is called a blocking effect. So, we have seen in a single queue also like it could be a blocking effect, which means that if the capacity limitation is there with respect to nodes, then blocking occurs. Of course, there are general notions of blocking, but even otherwise also, blocking can occur; but for us, blocking means that a customer, after having completed the service at, say, node i, he wants to go to node j. But node j has only a finite capacity meaning that it can only a finite amount of customers can queue in front of it to get the service, and that limit is reached, which means that capacity is full.

So, from this, after service completion, now he cannot move to the destination node immediately. So, such an effect is what is called blocking. In a single queue, what we have done is that we have assumed that the customers who are coming when the system is full or last to the system. Then we looked at the parameters of that, like what would be the probability of that and how to minimize or how to create capacity in such a way that you want to be connected. So, you want to study all those things. But in a network scenario, such a loss in between, if it is happening at the arrival point, it is a different matter. Like one can treat it as if it is a single queue, but if it happens when starting from one node to the other node, if it has to happen, that may not be a realistic situation; it can happen; it is not that it cannot happen. But then, it is not a realistic situation in many different situations.

So, it is not a practical option; you keep only that as one of the only options that are not practical for networks of queues. So, but then how you will treat that particular scenario, how you will make the system react to how the system will react to such a scenario depends on what kind of policy you are going to adopt when blocking happens. So, these blocking characteristics could be of different types; we see a few of them, for example, rejection blocking. This is basically what we could; we are seeing it in a single queue which is basically once the customer is blocked, he is forced to leave the system. And this forced to leave thing can happen only in open networks; obviously, in queues in closed networks, since no customer can leave or no customer can enter, that will not happen. So, this kind of blocking, if that is the policy, then this is similar to what you are observing in the case of a single queue; this is called "rejection blocking," which means you reject when the customer is blocked. The other is "transfer blocking": this is what, typically normally, one assumes. So, the blocked units wait in the current queue after having completed their service. Now, it cannot release the server; the server is also like cannot go to the other customer sooner because the current customer has not yet left the system. And this current customer, who has already completed the service, cannot move to the next node because there is blocking there. So, this is what is called as transfer blocking. So, he will wait with the current server as long as he is able to move to the destination node. So this is called transfer blocking and then "repetitive service blocking;" the blocked job goes to another service at the current node once he cannot move to the destination node. And this cycle can also repeat if necessary; right this is, there are even other different policies like how one can do. So, rejection blocking, transfer blocking, and repetitive service blocking is some of the common types that occur depending upon the scenario, but the analysis of such queuing networks where blocking occurs is much more complex, and it may not have the product form solution. Recall in a series network what we have done? Though the network is in series, the joint probability of number in node 1, number in node 2, number in node 3, and so on, the number in node k is given by the product of the individual distribution, which means that what you can read particular node in isolation. You analyze it; you get the quantity, then you can get the quantity for the whole network that was the scenario, which is the advantage of product form solution. But here, it may not have a product form solution as you would see now with a simple example.

Say how things get complex; we want to highlight that you are not going to go very much in detail for which we consider a two-node series queue, but with blocking.

- Consider a very simple two station, single-server at each station series network, where no queue is allowed to form at any of the station.

  ► Arrivals follow $PP(\lambda)$ and service times at the two stations are $Exp(\mu_1)$ and $Exp(\mu_2)$, respectively.

  ► The blocking policy is 'transfer blocking,' which means the customer will wait with the current node until he can move to the next node; that is what is the scenario.

  ► The system can be represented as $M/M/1/1 \rightarrow \bullet/M/1/1$

  So, this is what we have.

- Possible system states are: $(0,0), (1,0), (0,1), (1,1)$ and $(b,1)$

♦ While the first four describes the number in each node, the state $(b, 1)$ describes the situation where a customer finished service at node 1 is waiting for the server at node 2 to become free.

- If p_{n_1,n_2} denotes the steady state probability of the system, then they satisfy

$$
\begin{aligned}
\lambda p_{0,0} &= \mu_2 p_{0,1} \\
\mu_1 p_{1,0} &= \mu_2 p_{1,1} + \lambda p_{0,0} \\
(\lambda + \mu_2) p_{0,1} &= \mu_1 p_{1,0} + \mu_2 p_{b,1} \\
(\mu_1 + \mu_2) p_{1,1} &= \lambda p_{0,1} \\
\mu_2 p_{b,1} &= \mu_1 p_{1,1}
\end{aligned}
$$

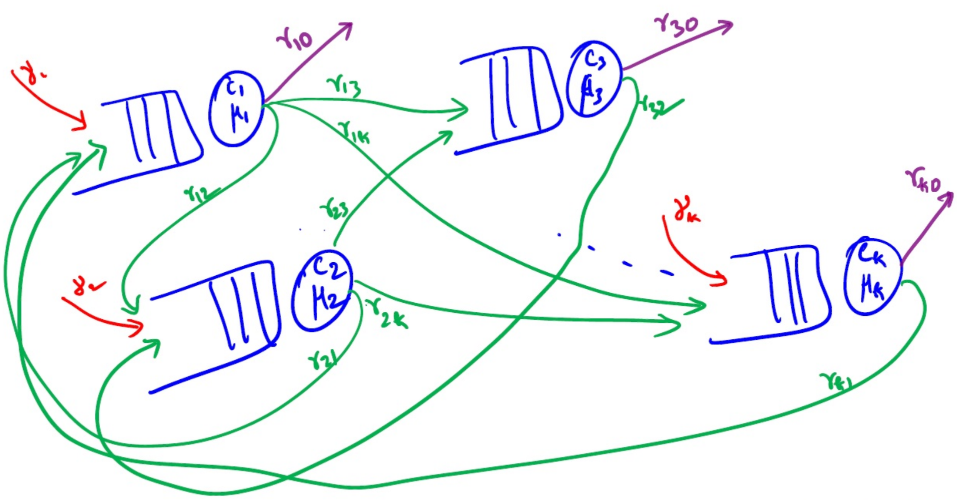
Then you can obtain the performance measure. So, given now, for example, even in this particular case, given certain parameter values, I will get a set of equations that you can easily solve to get the quantities with no issue. But the complexity is, it is very easy to see from this set of equations itself. Like if you start allowing suppose instead of 1, there are 2 customers here, then one more state will come. Like this, it will expand, but the point here is that if you allow positive but finite queue capacity in front of each node, this will expand the, but the complexity is that you have to write the balance equation for each possible state. Conceptually it is fine; you can write down whether it is even 1 million; you can think that conceptually you can write down, but practically that is not a feasible option. Compare this with respect to the non-blocking network that we have considered earlier series network. In this case, you have to treat a node in isolation, but here you have to keep it together only, and if the nodes are large, and if the capacities are large enough, then you can see the imagination that you have to handle the whole thing together because it may not have product form solution.

So, for a large but finite set, one can still use numerical, but otherwise, it is very difficult to analyze such; that is why we say that this is much more complex, but blocking is relevant and with blocking effect has to be analyzed. So, one has to handle it at some level, but things are complex, which, in our normal scenario like we would not like to venture into at this point of time. Those who are interested can look into that; of course, there are books on networks with blocking themselves because of their relevance. So, that is one can look into that. This is about blocking; now, we will not come back to this blocking aspect later on, but we will be considering much simpler ones. Now, we generalize now blocking we have stopped; we have series we have done. Now, in the series context itself, we have now given the idea of the effect of blocking in a queueing network and what it can have. Now, it will consider truly a full general open Jackson network. So, what is that?

- We now consider the general open Jackson networks as described earlier.
  - 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).

- 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.
- As usual, we want the joint distribution $P\{N_1 = n_1, \dots, N_k = n_k\} = p_{n_1, n_2, \dots, n_k}$ from which we can obtain other required quantities.

Now, once we have this, we know once we have the joint distribution. Now, suppose if I want the marginal distribution of n_1 alone, then I can sum over the remaining indices to get this marginal distribution, and I can obtain. So, any number of joint marginals anything I can obtain, once I have this full joint distribution, that is what our objective is to see; like in the series case, we saw that this breaks into the product of the marginal. So, in a much more open general Jackson network, again, we were interested to see how this can be obtained.



Recall in the previous class, as we have in the previous lecture, we have drawn this just to depict one typical open Jackson network, where there are k nodes. In each node, there are c_i number of servers with the rate μ_i; capacity is infinite under each node. These red colour arrows depict the arrivals from outside, and these purple colour arrows depict the departures from any particular node to the outside world, and these green arrows depict the movement or routing of customers from one node to the other according to certain probabilities, that is what we have depicted here. For example, from here, either it can go out, or it can go to node 2, node 3, and node k, possibly. There could be n number of nodes in between; it does not matter; this is a typical example that we have depicted. So, just recall this is what is the much general open Jackson network, open because there is at least one node in which customers can come in and at least one node in which customers can go out apart from movements between the nodes which is given by these routing probabilities from each node. This is what you keep in mind. Now, like whatever we are going to analyze, this particular network is what, in this generality. Now, suppose this has feedback; now, if you want a feed-forward network only, then we have to eliminate the feedback from here, then you will get the feedback feed-forward network. It can still be in the open Jackson framework, but without feedback, but this is much more general is what then we have in mind.

- The notation for the k-component vector is as follows:

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

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

  For now, we will generalize this to later c server just for understanding purposes; otherwise, things become a little complex. So, we will assume that there is a single server in each node; though this generality is c servers, we will come back to the c servers. For now, we will assume that this is a single server at each node; it is what is the scenario that we are assuming.

- The stochastic balance (global) equation for state $\bar{n}$ with $n_i \geq 1, \forall i$ is:

$$\sum_{i=1}^{k} \gamma_i p_{\bar{n};i^-} + \sum_{j=1}^{k} \sum_{\substack{i=1 \\ (i \neq j)}}^{k} \mu_i r_{ij} p_{\bar{n};i^+j^-} + \sum_{i=1}^{k} \mu_i r_{i0} p_{\bar{n};i^+} \quad = \quad \sum_{i=1}^{k} \mu_i (1 - r_{ii}) p_{\bar{n}} + \sum_{i=1}^{k} \gamma_i p_{\bar{n}}$$

  ► 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 $n_i = 0$ to zero.

So, this is what is the global balance equations. Now, we need to solve this to get $p_{\bar{n}}$.

- Jackson (1957, 1963) showed that the solution to the balance equations is in a '$product form'$':

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

  ► So, this definition of product form is less restrictive in the sense that we do not require in this form of definition that the C need not separate into an actual product.

  But if it so happens that it is still a product, like in a series network case, it did happen that the C itself was separated into a product of the individual quantities, so that was very nice. So, that is what is product form, some people call that as a product form, but this definition is less restrictive in the sense that C need not factor into products.

- We will give Jackson's solution and then show that it satisfies the balance equations. And this is also called as Jackson's result or Jackson's theorem for an open network.

  Many people would like to call this as simply as Jackson's theorem for an open network, which means that the solution to the joint distribution is what is given by Jackson's theorem, and that is what we are calling the solution.

  Now, to do that, what we have to have is that we need to find out these λ_i's. What are they?

- λ_i be the total mean flow rate into node i (external and rerouted). Given γ_i's and r_{ij}'s, to satisfy equilibrium flow balance at each node, we have the '*traffic equations*' given by

$$\lambda_i = \gamma_i + \sum_{j=1}^{k} \lambda_j r_{ji}$$

or, in vector-matrix form $\quad \boldsymbol{\lambda} = \boldsymbol{\gamma} + \boldsymbol{\lambda} R. \qquad (R : \text{Routing Matrix})$

The solution to the traffic equations is $\boldsymbol{\lambda} = \boldsymbol{\gamma}(I - R)^{-1}$. The inverse of $I - R$ exists as long as there is at least one node for exit and no node is totally absorbing.

So, as long as $\boldsymbol{\lambda} = \boldsymbol{\gamma}(I - R)^{-1}$. is there so $I - R$ inverse would exist. So this means that under that condition, $\lambda_i = \gamma_i + \sum_{j=1}^{k} \lambda_j r_{ji}$ system of equations becomes linearly independent, and you can find a solution to this system.

- Define $\rho_i = \frac{\lambda_i}{\mu_i}$. The network will be at equilibrium if each of these nodes are at equilibrium which can happen only if $\rho_i < 1$ for $i = 1, 2, \ldots, k$.

- The steady state solution to the balance equations is

$$p_{\bar{n}} = p_{n_1,n_2,\ldots,n_k} = (1-\rho_1)\rho_1^{n_1}(1-\rho_2)\rho_2^{n_2}\ldots(1-\rho_k)\rho_k^{n_k} \qquad n_i \geq 0,\ i = 1, 2, \ldots, k$$

Now, you can see it here. So, this is as if it is $p_{\bar{n}} = p_{n_1,n_2,\ldots,n_k} = \left[(1-\rho_1)\rho_1^{n_1}\right]\left[(1-\rho_2)\rho_2^{n_2}\right]\ldots\left[(1-\rho_k)\rho_k^{n_k}\right]$; you are seeing it here. So, which is a true product of the marginal distributions.

So, the C times $\rho_1^{n_1}$ and so on is what we have written. But that C actually factors into the product of $1 - \rho_i$'s, and hence this is actually factors into this. So, which is also similar to what you had seen in the case of a series network. So, this is what is Jackson's solution or Jackson's theorem gives you the solution to this open Jackson network as this. In the case of a single server, that is why we are getting. We have taken a single server just to take some simpler forms of solutions like this to exhibit. So, the balance equation is writing it easy, and then this is also easy to see here.

Now, what are the implications of this result. So, once the traffic equations have been solved and you have obtained λ_i's, then this is as if you can see that

$$p_{\bar{n}} = p_{n_1,n_2,\ldots,n_k} = \left[(1-\rho_1)\rho_1^{n_1}\right]\left[(1-\rho_2)\rho_2^{n_2}\right]\ldots\left[(1-\rho_k)\rho_k^{n_k}\right] = p_{n_1}p_{n_2}\ldots p_{n_k},$$

it is a product of the marginals what you are actually seeing it as if it is happening here. So, this is what we say is a true product of the marginal distribution. So, because of $p_{n_1}p_{n_2}\ldots p_{n_k}$, it is not just the product form the way we define, but because of the product form of the marginals; in terms of marginals, what this implies is that the individual nodes may be considered in isolation just like we have done it in the series tandem network case.

And then putting together, you can get the joint distribution, it is a very powerful result in that because such a complex network you are seeing. But the analysis what ultimately you see, if the network satisfies this property, then what you are seeing is that individually I can treat them as if it is a separate entity and then I can put together and do, it is a great thing too.

But here, there are certain points that you need to observe.

- The network acts as if each node could be as an independent $M/M/1$ queue, even though that is really not the case.

  Whereas, in series networks, they are really independent, but here they can be viewed as an independent, and the joint distribution can be written as a product of marginal distribution, as it turns out here. The flow into each node behaves as if it is, I mean, it is Poisson, even though they may not really be Poisson in nature; especially this is especially the case when there is feedback in the network. When there is feedback in the network, it has been shown in the literature, or it can be shown with a little bit of effort that the actually the internally when the flow if you look at into any particular node they need not be Poisson. One can split that further and go deeper into that; we are not going to do that; what you what is that relevant here is in a series network when we have only feed for or in the general feed-forward network, they will be really independent. But, whenever in an open Jackson network, when the feedback feature is there, the internal flows are not really Poisson. Like this has been shown long back, it is a new feature that we are talking about here. But for us to understand that is a bit difficult, but it is actually the case that the internal flows are not really Poisson in general, at least whenever there is feedback; if there is no feedback; obviously, it can be shown that they are truly Poisson, but otherwise it is not. But, even then, the advantage and the plus point of this Jackson network, even then $p_{\bar{n}} = p_{n_1,n_2,\ldots,n_k} = \left[(1-\rho_1)\rho_1^{n_1}\right]\left[(1-\rho_2)\rho_2^{n_2}\right]\ldots\left[(1-\rho_k)\rho_k^{n_k}\right] = p_{n_1}p_{n_2}\ldots p_{n_k}$ holds, even if there is a feedback that is what is the power, or the significance of this particular result that you get.

- The flow into each node behave as if it is Poisson, even though they may not be really Poisson in nature (i.e., if there is a feedback in the network).

- To verify that the solution, we first show that $p_{\bar{n}} = C\rho_1^{n_1}\rho_2^{n_2}\cdots\rho_k^{n_k}$ satisfies the balance equations and then $C = \prod_{i=1}^{k}(1-\rho_i)$. Plugging this into the balance equations gives us

$$\sum_{i=1}^{k}\frac{\gamma_i\mu_i}{\lambda_i} + \sum_{j=1}^{k}\sum_{\substack{i=1\\(i\neq j)}}^{k}\mu_i r_{ij}\frac{\lambda_i\mu_j}{\lambda_j\mu_i} + \sum_{i=1}^{k}\mu_i r_{i0}\frac{\lambda_i}{\mu_i} \quad =? \quad \sum_{i=1}^{k}(\mu_i - \mu_i r_{ii} + \gamma_i)$$

  From the traffic equations, we have

$$\lambda_j = \gamma_j + \sum_{\substack{i=1\\(i\neq j)}}^{k} r_{ij}\lambda_i + r_{jj}\lambda_j \quad \Rightarrow \quad \sum_{\substack{i=1\\(i\neq j)}}^{k} r_{ij}\lambda_i = \lambda_j - \gamma_j - r_{jj}\lambda_j.$$

  Substituting this into the above, we get

$$\sum_{i=1}^{k}\frac{\gamma_i\mu_i}{\lambda_i} + \sum_{j=1}^{k}\frac{\mu_j}{\lambda_j}(\lambda_j - \gamma_j - r_{jj}\lambda_j) + \sum_{i=1}^{k}\mu_i r_{i0}\frac{\lambda_i}{\mu_i} \quad =? \quad \sum_{i=1}^{k}(\mu_i - \mu_i r_{ii} + \gamma_i)$$

$$\Rightarrow \quad \sum_{i=1}^{k}\left(\frac{\gamma_i\mu_i}{\lambda_i} + \frac{\mu_i}{\lambda_i}(\lambda_i - \gamma_i - r_{ii}\lambda_i) + \lambda_i r_{i0}\right) \quad =? \quad \sum_{i=1}^{k}(\mu_i - \mu_i r_{ii} + \gamma_i)$$

- Finally, we get

$$\sum_{i=1}^{k} \lambda_i r_{i0} \quad =? \quad \sum_{i=1}^{k} \gamma_i.$$

  Now, if this is true, that means the solution satisfies the balance equation. Now, you see what this means is. $\sum_{i=1}^{k} \gamma_i$ is the sum of flow into the network from all nodes, and $\sum_{i=1}^{k} \lambda_i r_{i0}$ is the sum of flow out of the network from all nodes. So, from λ_i ith node with probability r_{i0}, it goes out. So, $\lambda_i r_{i0}$ is from the ith node it is the flow out, the sum over all the nodes, and this is the sum of and for equilibrium to hold means that this must be equal right and hence this is true. So, this is true, and hence, the form $p_{\bar{n}} = C\rho_1^{n_1}\rho_2^{n_2}\ldots\rho_k^{n_k}$ satisfies the balance equation.

- Now, C can be obtained from

$$\sum_{n_k=0}^{\infty} \cdots \sum_{n_2=0}^{\infty} \sum_{n_1=0}^{\infty} C\rho_1^{n_1}\rho_2^{n_2}\cdots\rho_k^{n_k} = 1$$

$$\implies C = \prod_{i=1}^{k}(1-\rho_i) \qquad \rho_i < 1, \quad i = 1, 2, \ldots, k$$

  Thus, the solution is verified.

So, we have verified, now the solution to this, or we have, in fact, seen the Jackson theorem essentially for single server thing. We will generalize this to the next one. Now, once we have this, then again, as usual, the performance measures can be obtained in a routine way.

- For single-channel node open Jackson network considered here, we have

$$L_i = \frac{\rho_i}{1-\rho_i} \quad \text{and} \quad W_i = \frac{L_i}{\lambda_i} \qquad \text{for node } i$$

  ⧫ This is because of the product form of the joint distribution, and does not imply that the nodes are truly $M/M/1$.

  It is not because of that it is coming; it is because of the joint distribution factors into the product of certain marginal distribution and from there we are getting this. That is how you are obtaining this quantity.

- The expected total number of customers in the network is $\sum_{i=1}^{k} L_i$.

- The expected total wait (sojourn time) in the network for any customer before its final departure is

$$W = \frac{\sum_i L_i}{\sum_i \gamma_i} \quad \text{(Little's formula for the entire network)}$$

  So, these are straight away the performance measures that you can obtain for ith node, and from there then, you can write for the entire network. So, we will stop here, and then more we will see in the next lecture.
  Thank you bye.