Introduction to Queueing Theory

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

Lecture - 02
Characteristics of Queueing Systems, Kendall's Notation

Hello everyone. As you see, we are looking at the quantitative aspects of queueing systems. A quantitative evaluation of such a queueing system requires mathematical characterization of the various underlying processes that drive or constitute that system. So, in many cases, you would notice these five things that we are describing here, first, the arrival pattern of customers. So, if you have a system now, what is the pattern of arrivals of customers to that system. Again I recall that customer is a generic term, it could be human beings, or it could be cars, or it could be messages, it could be emails, or it could be telephone calls or anything of that sort. Then the service pattern of servers, again, a server is a generic one. It could mean you know various things depending upon the context, what we mean by servers. It could mean a link, or it could be a counter where a human person is serving them, or whatever is the case. Then the number of servers and service channels, system capacity, and queue discipline. So, in many cases like this, you would see that this provides an adequate description, but again, this is only the very basic element that you know one would look for in every system. Now, depending upon the system, there will be more than these five elements you need to describe a complete system. So, out of these, the first one that we look at is the arrival pattern. When the arrivals are deterministic, which means that, for example, every two minutes, one arrival happens. It is a nonrandom one, it is a very deterministic one, but generally, that does not hold true in reality. So, arrivals are stochastic. So, when they are stochastic, you need to describe them through their inter-arrival times, like the time between the arrival of two customers and two consecutive customers. So, that is what we call inter-arrival time. So, that inter-arrival time is required to be described in terms of its stochastic nature or random nature; for example, it could be a Poisson process. We will see what it is, but as of now, like currently, few terms you take as it is, but we will know in the course of time what is a Poisson process. So, for example, this is one. So, basically, we are saying that inter-arrival times, behave, or you make certain assumptions on that. And the main assumption here is that the inter-arrival times, a time between two arrivals, two consecutive arrivals is exponentially distributed. When you use that as you will in a Poisson process, that is what happens actually. So, that is the randomness, and again arrivals could be single arrivals or bulk or batch arrivals. So, it could mean like one by one people can come or in case of a bus or train arriving at a station, you know you will see that the arrivals are happening in bulk. So, in that case, you know you need to describe the bulk size, so that is basically the size of the batch.

Whether it is a constant one or is a random one, and if so, what is the randomness of that again. So, that is the second component that you will look at; that is, this is to see whether the arrivals are bulk arrivals or single arrivals, bulk or batch arrivals. And again, arrival pattern over time, whether it is stationary or nonstationary, of course, we will consider only the stationary case in this course, but you know what do we mean by that stationary and nonstationary it is with respect to time. Whether at all times you observe the same pattern or at different time points, you observe different patterns, meaning, for example, if you take a restaurant. What might happen is that morning breakfast time, or during lunchtime or during dinner time, the arrivals could be more in number, the pattern, and the rest of the time, you will have a very less number of arrivals. So, if you assume that it has the same pattern throughout the day, meaning that you know any time is like any other time in some sense, then you know you have a stationary situation. So, the Poisson arrival, simple Poisson process arrival, is basically we were in the stationary pattern; it is basically with respect to time we are talking about. Nonstationary- suppose if it is not the case and it is dependent on time, so it could mean that at various time points, for example, of the day you have different arrival patterns which are described through the rate of arrivals per unit of time no matter, whatever unit that you are taking it up. So, it could be nonstationary. So, in reality, you would see that nonstationary is what is happening, but then again, what is the exact pattern of this nonstationarity, again, one has to do a lot of studies on that. But to get a first-hand understanding of the system, as you know, one might always take the average rates to be the rate of arrivals for the whole time period, a duration that you are considering.

So, in that case, one could assume stationary, and that is how you can assume that stationary is what the arrival pattern that you will have. Of course, in this course like, we are considering only the stationary arrival process, but these are, or the other word could be an appropriate word could be like time-independent and time-dependent one in this particular case. That is the third component in the arrival pattern that you are looking at. Then you have the customer behaviour upon arrival to the system. The customer could behave differently when he arrives in the system to get the service. As soon as he comes, he observes that the queue and he sees that there is a long queue waiting and he says oh this is not going to work for me like, I am going to quit right away and then I am going to I am deciding not to enter the queue. So, that phenomenon is what is called as balking. Balking in queueing means, or in general, means that you decide not to enter the queue, so this is one phenomenon. Now, what do you do, you join the queue, for some reason, you decide you join the queue, and you wait for some time. And after waiting for some time, you lose your patience, and then you see, now I cannot wait anymore. I am going to leave the queue; I am going to exit the queue at this stage. So, that phenomenon is what is called as reneging. Reneging means that you join the queue, and after spending some time in the queue then, you decide before you get the service. Of course, in the meantime, if you get the service you know, you will get your service done, and you will leave. But your service if not yet started, so then you said, fine, I cannot wait anymore; I may have something else to do. So, you just decide to leave after joining the queue, so that phenomenon is called reneging. Then there is a third phenomenon like this, called jockeying. What is that? So, suppose if you go to, say, the airport or the supermarket, you see multiple queues against or in front of the

multiple servers. Then you will see that you first join one queue; then, you see that the other queue is moving faster, so you move to the other queue. So, after spending some time in one particular queue, this is in a multi-server scenario only; this can happen. In a single server, of course, there is no jockeying that can happen. So, multi-server after spending some time in the queue, then you decide that fine that particular queue is moving faster, so I am going to switch to the other queue. So, that is called jockeying; that is, switching between different queues.

All these phenomena broadly can be grouped under customer impatience. But in the first two cases, he is actually abandoning, which could also be called queues with abandonments. The third one is that now he is not really abandoning the system but abandoning a particular queue. So, that is a little bit finer if you want to look at it, but all three of them fall under the case of impatience. Then there could be a dependency between arrival and dependency on queue length. So, there might be between the first arrival and the second arrival right, there may be some relationship, or between the first arrival at the tenth arrival, there may be some relationships. Whether these are completely independent or otherwise, one can have that kind of scenario. For example, if you take the internet traffic, it is known to be exhibiting some sort of dependency, and likewise, you know in many situations you know you would find. But the easier thing to model is to assume that independence, but if you want to incorporate the dependency, then one can do that, so that is the dependency between arrivals. Maybe one arrival might require something more of a similar kind of arrival. So, if you are looking for some disease, being detected and then you consider that as an arrival; now like the moment you see a particular person is infected and you have found, and then you will see that there might be in that particular location you might find more number of such cases might be depending upon the disease. So, there could be dependency in a simple scenario. Similarly, dependency on the queue length, there may be some mechanism by which the arrivals are controlled in some way or the customer himself or the system itself operates in such a way that you know, the queue length is also playing a role on the arrival pattern. So, these are some simple descriptions, but there could be many more, even with respect to the arrival pattern itself, which can describe the arrival pattern. So, one needs to completely describe the exact nature of an arrival.

So, because like any mathematical model, a queueing model also relies on certain assumptions, it is exact to the assumptions. So, the moment the assumptions are violated, the model is no longer valid. One has to completely describe the nature of arrivals after observing the phenomena in systems that you would see. Now, the second component here is the service pattern. So, pretty much whatever you know we have seen with respect to the arrival pattern also holds true for the service pattern. Again service times are if they are deterministic or; that means it is constant. That means there is nothing much to analyze, so things are very well predictable in some sense. So, in general, the service times are stochastic, and again they are described through service times. So, the simplest one is assuming that to be an exponential distribution, for example. So, service time, you mean duration of the service time, duration of the service that your customer is getting from the server, that duration that time is what is service times. So, pretty much all these inter-arrival times and service times, any nonnegative random variable, like would, can

come in place, but the simplest ones are exponential. So, there are inter-arrival times, and here are service times for this case. So, there are service systems though I also mean that most of them are stochastic; obviously, these service times are described through stochastic random variables or phenomena. Now, again the service could be in bulk or single. So, bulk here would mean, for example, the same example that you can pick it up. Like when the train leaves or when the bus departs or a jungle safari like when the safari leaves, so the customer, for example, in case of a train or jungle safari people might come one by one singly, or it may be smaller batches as well, but any bigger batch size it leaves. So, which means that the service is done, probably a ferry which is crossing the river from one side to the other, so, this is all like you could see that the service is happening in bulk. So, now, again, the bulk, what is the batch size? What is the group size? So, that is described through the size of the batch. So, that can happen and also service pattern over time whether the time the service again the rate of service if you look at it. So, the service could be stationary or nonstationary, which means it could depend on time or be independent of time. So, we consider again, in this particular case, the stationary case only. As opposed to that, there could also be a dependency on the number of customers waiting for service; this is typically what these are all called state-dependent. Again, the arrivals and services could be state-dependent; as we said, the state means the state of the process or the state of the system. So, dependency on the number of customers waiting for service could either expedite, or the server may start serving at a faster rate, or it can happen the other way around like you could find an example.

So, all these things like it is called state-dependent services . So, maybe like as the experience is gained by serving few customers in the beginning, the rate of service might become more efficient. So, it becomes more number of customers can be served or after seeing the queue you are increasing, or you get some help from someone to increase the service rate or and so on. Like there could be many ways that you could work . So, again this is all broadly the service pattern; whatever we will lo for that into this case. Again with respect to server, something might happen. Of course, those are all further things, but whatever happens with respect to the service part, . So, if you describe it completely, then only like you are you are described completely about the pattern of service that you have. Then the third ingredient is that the number of servers. So, this could be an infinite number of servers which ah is basically representing a self-service kind of systems or there could be a finite number which could be even one server, or it could be some ten servers or some finite number of servers, . So, that is the number of servers, ; this is the another description that you need to make. So, this is basically, as, like, this is number of servers depends on a fundamental trade-off which is cost to the business versus reduced delay. You have more servers in place; you have less waiting time for the customer; you have less number, you will have more. Now like what you do in such a case? Like if it is a pure economical situation a decision, then you can such simply make a trade-off. But in some cases, again, you will not be able to increase the number of servers , then one has to lo at other I mean other aspects of it. But if it is possible if it is a business manager's decision to decide the number of then one can have this trade-off between that, then one can decide the number of customers in to this case. Another description with respect to this number of servers is the how the waiting lines or queues are configured in the system.

So, all these things it is called state-dependent services. So, maybe as the experience is gained by serving a few customers initially, the rate of service might become more efficient. So, it becomes more number of customers can be served or after seeing the queue you are increasing, or you get some help from someone to increase the service rate or and so on. Like there could be many ways that you could work. So, again this is all broadly the service pattern; whatever we will look for that in this case. Again with respect to the server, something might happen. Of course, those are all further things, but whatever happens with respect to the service part. So, if you describe it completely, then only like you are described completely about the pattern of service that you have. Then the third ingredient is the number of servers. So, this could be an infinite number of servers which is basically representing a self-service kind of system, or there could be a finite number that could be even one server, or some ten servers, or some finite number of servers. So, that is the number of servers; this is another description that you need to make. So, this is basically, as, like, this is the number of servers depends on a fundamental trade-off which is a cost to the business versus reduced delay. You have more servers in place; you have a less waiting time for the customer; you have less number, you will have more. Now like, what do you do in such a case? If it is a purely economical decision, you can simply make a trade-off. But in some cases, again, you will not be able to increase the number of servers, then one has to look at other, I mean other aspects of it. But if it is possible if it is a business manager's decision to decide the number of then one can have this trade-off between that, then one can decide the number of customers in this case. Another description with respect to this number of servers is how the waiting lines or queues are configured in the system. Say, for example, in a multi-server system, whenever we, one more thing that we want to say before we go further is the number of servers; when you mean, we always mean parallel servers in general; it goes without saying throughout unless otherwise mentioned that number of servers we always mean parallel servers, not sequential ones. So, they all have equal opportunity whenever a customer comes unless it is done otherwise.

So, in a multi-server system, when you have this, all these parallel servers are there whether they are fed by a single queue or each server is fed by its own queue or a hybrid model. So, this is how the configuration of the waiting lines or queues is also part of this part of the description. Say, for example, when all servers are fed by a single queue, you can think about a salon where you will be waiting; many people will be waiting. And then suppose five hairstylists are there in that. So, then you can think that any one of them is a parallel server, which means you are saying that they are all, in some sense, a homogeneous group of servers. This means everyone can do all the work that is supposed to be done in that salon. So, these are all further assumptions; if you want to go deeper and deeper into how you are assuming. Now, if that is the part that you have here. So, then, like in such a case, the common waiting line is what you will be having. As opposed to that, suppose if it is a grocery store or something, then there might be each server fed by it is own queue; it can happen in other places as well or a hybrid situation where for example, if you are in an airport like in if you are now going out of the country and if you are looking at immigration check. Then you will see that there is a common queue, and again you can even

think in a domestic level like you could have the security check path where you have there is a common queue, and you are being fed into another smaller queue for your checking. So, you could have a hybrid situation; these kinds of things can happen. So, this is how like the part of this description. So, whenever we say those parallel servers are there like, we will assume in many cases that a single queue will feed it, or if it is otherwise, then we will state explicitly that each server is fed by it is own queue or if this is the case then hybrid nature and so on. So, this is the third component of the description of the queueing system. The fourth one is the system capacity. The system capacity means how many customers can be there in the system at any point in time. When we talk about the system, as we already pointed out, it consists of the queue plus the ones in the service. The ones waiting for the service and the ones in the service, this is the whole system capacity; this is what we call the whole system. And whether there is an infinite capacity for customers to wait. Again, in reality, there is nothing like an infinite capacity, but a very large one can be approximated easily by an infinite capacity, and as you will see, the analysis also might become easier. But again, it is an approximation in some sense. So, what you are assuming is that this is to be infinite capacity for customers to wait, or there is a finite capacity. Now, whenever there is a finite capacity business, you have a scenario that is called blocking. So, common in say computer and communication networks or data networks or mobile networks and so on.

Because there is a capacity limit, so, the link between two nodes in a network, the nodes could be anything like it has a finite capacity which is it has a finite bandwidth, through which only up to which only like things can be transmitted. So, when you try to make a call, all these lines in this root are busy, meaning that it has reached that capacity, and you cannot transmit anymore. So, that is the finite capacity system. So, that is the part that you can look at basically as the system capacity; whether it is infinite or finite, though, in reality, things will be finite; it will be more realistic if you assume that finite. And study the behaviour of this one, the blockedness, because what is the probability in such a situation that these phenomena suppose what is the probability that if a customer were to arrive at a particular time, what is the probability that he would have been blocked. So, your system capacity should be such that this is bounded by some very small number. Of course, all this is what we call quality of service, so there are regulators who describe this quality of service parameters in the sense that they need to meet this requirement; that is what you would call a good service in some sense. So, to do that, you have to arrive at that. So, in that case, you have to assume that to be a finite server and obtain this probability and then see whether it meets the requirement. But you would assume simple analysis kind of thing you can assume that to be infinite as well. So, this is the fourth component rather. Now the fifth component is what we call the queue discipline. So, this describes the manner in which the customers from a queue are selected for service. So, the most common one that you would see across various applications or various systems is what you call "first come first served" or, in short, "FCFS"; first come first served or "first in first out," "FIFO." So, this is what is the most common that you would see, and you might feel that this is the fairest system that you might have when you have a queuing system or when you have to queue when you have to wait. But there could be other kinds of queue discipline depending on the application you are trying to analyze. The other one could be, for example, the "last come

first served." Suppose it is an inventory; for example, you are making some products, and then it is an inventory item. It does not matter which one and which item you are picking it up. So, when stocks are piled, probably the last one is what you would pick for further processing. So, it is "last come first served," and there could be a random selection depending upon the scenario. And there could be, so basically, in the random selection, it does not happen. So, if there are n customers, ten customers are waiting, so all ten could have an equal probability of being selected. And processor sharing is so common in computer and communication networks. So, it is called processor sharing, in which the server processes all jobs or all customers or serves all customers or processes all jobs. In computer communication or network parlance, you would call these as jobs rather than customers; it is so common, but we will use customers rather.

So, all customers or jobs are handled by the server simultaneously. So, this is common now whenever you are working on 2 customers or when you are working on 2 jobs and whenever you are working on 10 jobs; obviously, the rates are not going to be the same. So, the existing capacity is being distributed to 10 different customers. For example, if you have an internet bandwidth that links that, if there are only 5 customers who are accessing that, you are going to get a higher speed, and if there are 100 customers who are accessing the same link. Then obviously, the existing capacity is being distributed across this 100 people, so; obviously, the rate or the speed that you realize from your end; is obviously, slow. So, so you have such a scenario which is what is called processor sharing phenomena or queueing discipline. Then there are these polling or round-robin systems. So, you have systems like this; there is a queue here, there is a queue here, there is a queue here, there is a queue here. So, what happens here like the server he rotates, rotates between these different customers, and then serves. It may be one customer after serving one customer, he may move to the one customer here, and then he may move to the other customer here, then one customer here, then come back to this again, or it could be a group of customers. That group either by group size, you could decide or time duration you will decide. Say, for example, if you are looking at a traffic junction. So, typically this is what you see here. So, this is what is the scenario which is called polling or round-robin. Within that queue, of course, obviously, then you will have the system again that whether first come first served, or you stand here, and someone comes from behind, you just move in front, and then he gets service first.

Things happen in very complex situations; if you cannot, I mean, describe everything completely, you need to have certain basic major features done. So, if this is the case, this is what is called polling or round-robin systems that you would have. And then you have this priority. So, there could be different classes of customers again in the service arrival pattern we could have it is said that actually, there could be different classes of customers; who are acquiring a different kind of service requirement based upon that. In some sense, that basically imposes the priority, so some customers may be higher in value or whatever is the reason. So, they do get priority, and you do see like in a hospital situation, priority customers are those who need immediate care. So, that is the case. So, priority is prevalent, and priority customer priority as a discipline also one needs to keep that in mind. Now, when this priority happens, there could be two things one is preemptive and non-preemptive. The first word is preemptive, and the

second thing is preemptive, which means what happens to the customer who is of lower priority and who is currently getting served. What happens to this particular customer when a higher priority customer arrives. Suppose this particular lower priority customer service is interrupted to start the service for the higher priority customer; that is the scenario that we call preemptive. And if that does not happen even though he is a higher priority customer, he waits till the customer of lower priority who is currently undergoing service, till his service is complete; if he waits, then that system is called nonpreemptive. Now again, within preemptive, then there could be preemptive resume and preemptive non-resume. So, what happens to the service that has already been done. Whether the service starts from that point onwards for the remaining service, then it is a preemptive resume, or he has to be served all over again preemptive non-resume. So, it means his service starts afresh that is the situation that you will have.

So, this is about the priority and preemptiveness, and there could be like many other things that you might observe, but the shortest job first, so, a requirement of the service requirement or the longest job first. So, there could be many different kinds of things that you need. But, as we said, the most common one was this first come first serve that is what you would see, but there are others depending upon the application that you need to serve. This is the fifth component that is used to describe a queuing system. And for basic systems like this might do, but there may be some other features that also need to be described to describe the queueing system completely. So, and again I caution this is not an exhaustive list, but some of them which are also the most common that you would see. So, for example, multiple stages of service. If you apply for a passport, you will see that you will go through multiple stages; you could consider that as a stage or think about what you call. In a hospital, suppose if you want to go, you want to get this physical examination done, or a full-body check-up is done, then what are the things that you need to go through where you will go. So, this you could call multiple stages of service. Then the most important one, which is most practically it is relevant, is this particular network of queues. This is what you would see in most real-life scenarios and in most applications because if you do not have the network, then we are talking about a single system. But in most places, it is not just a single system; it is a network like after getting after you you are getting served from one place, you move to the other place other service centers, you get service one after the other before your complete service requirement is met.

So, then you have a scenario where you have networks of queues. So, that is why the common word communication and computer networks are all again networks because nothing is common. Because suppose if you are talking about a mobile network, there is a base station in this place, and after some distance, there is a base station there. Like you are calling from a range of this base station to capture the signal which will go to this base station and from here transfer to the other one and from here and so on or whatever is the phenomena that you have it here. So, you have a network or wherever things have to pass. So, in a production network, an item needs to go through different stages before it comes to its final stage. So, networks of queues again each one would be the network each one is a queueing system and when they are interconnected when the customer is moving from one to the other according to some rule. It is not that everyone will go through all that is the different phenomena that is the special case, but of course, in general,

also you could have when we study this you will see this one. So, this network is so common in reality to study this as a network of queues. We have to study the single system first before we move to the network of queues. Then there is recycling or feedback. So, you are manufacturing some product, and you do the quality check at the end of the production, and if you see there is some defect, you send it for reprocessing. So you would have recycling phenomena or feedback phenomena that you would have. Then you will have a retrial phenomenon of blocked customers. So, whenever the system is full to its capacity, a customer is said to be blocked. Now, what happened to these blocked customers. So, that is another phenomenon that is again common in communication networks and systems where this is the phenomenon that we call retrial. Retrial means that you let customers; you are assuming in some sense the customers are in some orbit, and after a certain point of time, the customer will try again to do that.

He is not gone out; he is not making a new entry already; he has come to the system, but he is not actually in the queue, but he is in some other queue virtually, which is basically called an orbit. And from there, he makes a retrial after a random amount of time; that is what you keep. So, then there could be server vacations. The server vacation is such a common word- server vacation. Vacation is such a common word that it could mean differently. For server maintenance, you down it for some time, or when there is nothing that getting, there is no customer. Basically, you try to incorporate such behaviour through this "vacations." It is a common term, but it could mean many different things depending upon the context. So, the server may take a vacation when it does not have anyone to serve. So, that is the meaning. Now, what happens after one vacation. So, it could be a single vacation, and he comes back, wait for customers to come, or he may go on whenever he sees the system is free like he can go for a vacation. So, it is called multiple vacations. So, again there could be different variations in that also. And there could be variations, for example, something called working vacation, meaning that in the normal rate in a working vacation situation, it may serve at a lower rate. For energy efficiency like in the military and other applications for energy-efficient systems like something operating on a battery, you will not make the server and a full battery available. So, you reduce at a lower capacity which means you down your battery level a bit; that is what it would mean in that way. Another phenomenon could be that service can only start when there are a certain number of customers in the queue, maybe because the server needs some setup time, a setup cost is involved. I mean, you are starting to start the server. So, you want some customers to accumulate before you start the service for the customers who are already waiting; there may be some customers waiting. This could be either based on the number of customers means until unless there are five customers, I am not going to start serving or there is a time, maybe after I will wait for half an hour and then until whoever comes like or it could be hybrid of both and so on. And there could be catastrophic events there could be a link failure. So, whatever packets that have been sent up to that point of time it has reached that particular link, they are all dropped. So, these are catastrophic events. So, again there is catastrophic again, it could be complete one total catastrophic, or there could be mild catastrophic, meaning that it will not kill all the packet that has been waiting, but it might kill some portion. How it is decided, again, there is a variation like these are all some other features.

It is not just all; there could be many different features that we need to describe. So, it all boils down to like observing the system and describing whatever is required to describe that system. The main essential features you need to describe to characterize the queuing system. And based upon that, then the analysis has to begin. So, that is what it is. Now, to describe these kinds of queuing systems, I mean in queueing theory, we use typical notations. Because when you use these notations, then the system whatever you are trying to say is basically understood. Again there are only very basic sets. A more complex system needs to be described in explicit terms. But the notations would describe to some amount like the general description of the system. So this is called Kendall's notation because he is the one who developed this first. So, this is; basically, we describe through typically by these five elements $A/B/C/D/E$. So, what are the characteristics? A denotes the inter-arrival time distribution, and B denotes the service time distribution. So, if the inter-arrival time distribution is exponential, then this A and B both are A and B like is denoted by the symbol M. M means exponential; ideally, it could have been E, but then that would confuse with this E. So, it is M. So, M means Markovian in nature because, in Markov systems, the natural distribution that arises there the holding time is exponential, so that is M. So, suppose if I have this A as M or B as M, that means that I have this is the one exponential in mind. D represents deterministic, E_k represents Erlang type k. It is in some sense related to the gamma. Of course, we have already, I mean, we will describe it in the probability part that what is the structure of this distribution that you can see. Again H_k is a mixture of k exponential.

And then PH is a more general one called phase-type distributions you will also see later in the course. G means it is a general. So, even this G whenever you are putting it could mean anything any of these also it can represent in theory, but when you are not able to represent by the usually accepted conventional symbols then you use this G, and this could mean general we will also take this to be general and independent. So, in some books or some places, you would find the notation GI to denote this; typically, whenever you need to distinguish between general and independent, general and dependent, then G and GI are used. But, since we generally have only this independent system, so for us, this G would work to mean that it is general and independent in this case. And this the third ingredient here in this symbol is the number of servers; it could be one server, it could be two servers, it could be some finite number of servers, or it could be an infinite number of servers. D represents the system capacity again; this could be system capacity could be one, and if there is a single server and there is one, that means that there is no waiting space per se in that place. There are situations like that, which are typically called loss systems. And there could be more capacity, and the system capacity could be ∞ as well, which means any number of people can queue it. Then there is a queue discipline which is what is E, which will be represented either by FCFS or LCFS or RS random selection or processor sharing or whatever the description that might come here. When we are writing these notations, the defaults that we are assuming here are that the defaults are here, or we will assume D to be ∞, which means the system capacity is infinity and $E = FCFS$. So, whenever ∞ and $FCFS$ are there, these two things are typically omitted from this D and E. Even without specifying if you write something, that means, Defaults: $D = \infty$ and $E = FCFS$ is assumed that the system capacity is ∞ and the queue discipline $FCFS$. If otherwise, it

will be specified explicitly. Say, for example, if I have a description of $M/D/2$. $M/D/2$ means that the inter-arrival times are exponential, the service time is constant, there are 2 servers, and the system capacity is ∞, and the queue discipline is $FCFS$. Remember here one thing that we need to look at here itself again this service and the inter-arrival processes these two also we generally assume that they are all independent; which means an arrival does not influence the service in some sense or arrival happening does not get influenced by this and vice versa.

So, these two processes are independent; there is no relationship; that is what we are seeing. If you assume that they are dependent, there will be more complexity; you will not have that thing, at least in this course. Now similarly, for example, $M/M/c/N$. So, in this particular case, there is an exponentially distributed inter-arrival times and an exponentially distributed service time, and there are c servers, and the system capacity is N. Similarly, $G/G/1$ in a similar way, $M/PH/\infty$ which means now the number of servers is ∞ the service time distribution is PH and M. Now we have seen something like a vacation, retrial, network and so many other phenomena like. Again they could be represented using this as to the extent possible, but it is not that completely they can be described by using notations themselves. Because some might use it, but that is not common, that is not standard notation for everyone to understand.

So, everyone will understand with these five points only. Additional notations are sometimes used if it is common within that community, but it is not common, but maybe it is better to describe it explicitly. At least when you start, so that you need to do. So, additional notations are used whenever necessary, but they may not be uniform or commonly accepted as far as that is concerned here. So, this solved the characteristics of the queuing system, which you need to describe in order to describe the system completely. Once this description has to be there to model a system, now depending upon the description of the system, you try to extract a model that the queueing model for further analysis in this case. So, to pick any model, for example, as with any mathematical model, a clear understanding of the characteristic of queuing systems is essential for the successful application of any queueing model that you have here. That is typical of any mathematical model, and in the queueing model scenario also, that is true. So, one needs to understand; you have to observe the system; one needs to understand the system to extract the essential features to make a model. The more and more assumptions you make, the more the model will become; it will go away from reality, but the essential features should not be destroyed. But then the result that you get out of the analysis will be meaningless. So, you need to keep a balance of that in trade off it. Say, for example, if you look at it, that is with respect to any mathematical model per se, but applicable here as well. Now, the observation is important because you would see that, for example, if you look at a supermarket, you have multiple counters, which is the most common model that you can employ there if you look at them. Whether it is a common queue or an individual queue, by the look of it, if you go and observe like, there is one queue for each of these checkout counters that you would see. So, you would see that these are multiple parallel servers, and suppose there are 10 of them and there are 10 individual queues is the model. But if you observe, if you look at the system carefully and if you observe the system a little bit more time, you will see that; suppose against 9 counters

you have queues and the 10th one which may be the 5th one, for example, he served all his customers. Now, is he going to be not doing any service and he will be simply waiting for the other customers to come directly from inside the shop, not from any of these queues. You would observe that immediately from the other queue, someone will come to this queue, and then he will start serving that particular customer. So, it is essentially though it would mean that multiple queues with jockeying is what is happening, it is not really the case. So, this could as well be analyzed because no server is going to be free if at least customers are waiting in the queue at the other counters. So, this could as well be that there are 10 parallel servers, and with a single common queue, you could as well model this phenomenon. So, you have to see like which one so which one is more efficient. Now, you will see later that having these individual queues or a common queue with anyone can reach depending upon the availability that any of the servers it can reach, which system is more efficient you can use intuitively also you can think; but you can also show it through analysis that which one is better. Now the single queue common queue is what is the better one it will turn out to be intuitively also you can feel so. Now, in this particular case like, if you analyze that they have 10 different queues, then things will be not correct because in that scenario, you have an individual queue, and when there are customers waiting in the other queue, this server will be free he will be idle. And that scenario is not going to happen at these checkout counters. So, then that model is not really the appropriate one. So, the common queue is the appropriate model that you would see. So, this is what we mean. At first sight, you would see that there are 10 different queues against 10 different servers, but it is not really the case. Really the real case is that you have a common queue with 10 parallel servers; that is what you have here. So, this kind of observation, like anything that, you have to observe the system and then see what is the best one happening there. That is how this model can be effectively utilized or used in practice. So, we will stop here, and then we will continue later. Thank you bye.