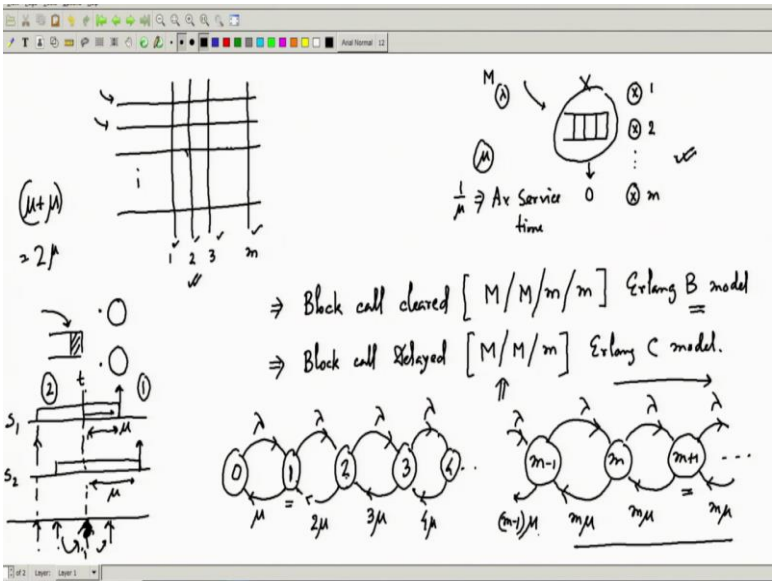
Communication Networks
Prof. Goutam Das
G. S. Sanyal School of Telecommunication
Indian Institute of Technology, Kharagpur

Module - 06
Erlang Formulas
Lecture - 27
M/M/m and M/M/m/m System

Ok so, so far we have discussed M M 1 queue ok. Now, we are in the position to launch towards means trunk switch analysis things which are actually known as Erlang Formulas. So, we will try to see some means some more complicated system Markovian system and then try to see how they are linked to trunk switching ok. So, that is what we will be doing in this class. Let us try to see.

(Refer Slide Time: 00:57)



So, what will do; we will now try another system where we have a queue and we have now multiple servers, not a single server ok. What happens? There is a queueing place where customers arrive, of course, a Poisson arrival. So, Markovian arrival with parameter lambda they have service time all servers are identical.

So, basically, they do not have every customer whichever server it goes they are identically treated and they get actually a service time that is independently and identically distributed exponentially and its parameter is mu; that means, 1 by mu is the

average service time. So, this is something we know. So, this is the system now we wish to analyze. So, this kind of system we want to analyze.

Now, let us try to first see why this system is identical to trunk switching. So, let us try to understand what happens in trunk switching. So, trunk switch if you remember we had let us say multiple lines ok 1 2 3 up to m and there are multiple inputs ok. So, over here it might be finite input we can also assume that it is actually too many inputs. So, it is almost infinite input. So, that should be our assumption otherwise it will be a finite customer. So, we can do the same analysis with finite customers. So, that is a different kind of analysis that we will have to do.

But right now the assumption will be that probably local subscribers too many customers are getting connected ok. So, that is almost infinite and their aggregated arrival rate is lambda ok. So, and they are Poisson because independently they are making arrival. So, they are independent of each other and it makes a Poisson arrival ok.

So, arrival means call arrival. So, somebody is requesting a call. So, some customers are requesting call some customer is requesting calls. At the time every customer requests a call you give a server just like over here, The only thing is that if the queue is not present ok? So, we can also think that the queue is not present; that means, the buffering place is 0 which we can assume, but we can also have some buffering place that might be also true.

So, the buffering place is there means it is like these earlier days if you used to call then they used to say that all lines in this route are busy and please wait. You can actually wait over there whenever the next connection is free or; that means, the next trunk server is free it will connect you.

And all the customers there might be multiple customers like you are waiting over your telephone there might be multiple customers who are waiting. So, that is actually a queueing place where everybody is waiting whenever the means, and whenever they are waiting they also keep a record that who has come 1st and who has come 2nd. So, that is how virtually they create a queue ok?

So, you are waiting there might be some other guys also who have taken their telephone and started ringing they are waiting, but basically, the system knows when you started.

So, they know those starting times accordingly they have arranged the queue and then whenever the trunk switch one of them is free; that means, one of the customers who is already talking has disconnected the phone. So, immediately the server becomes free.

So, suppose all these are occupied already, now suddenly whoever you have given the second connection has disconnected it; that means, this has now become available. Whenever it is available the 1st one standing in the queue or 1st one who has arrived who are waiting will be given the server.

So, it is completely a queueing system as you can see they are making Poisson's arrival. If the servers are there they are giving the server. How much time they are taking for service? The call duration which is 1 by mu, is exponentially distributed no problem with that all customers are equivalent which is also true, so, equivalent customers because they will be all making voice calls are equivalent kinds of customers we can assume that way ok.

So, if they are equivalent kinds of customers then all servers are equivalent because they give one connectivity one telephonic connectivity of 64 kbps or TDM connections. So, if this is the case then it is actually this kind of multi-server system ok. So, all we have to now do is analyze this kind of multi-server system.

So, in this multi-server system, it might be two scenarios might occur. So, in one scenario you might be waiting. So, whenever you are basically all servers are busy this particular thing you hear immediately your call might be dropped; that means, you try to make the connection because all servers are busy all trunks are busy. So, your call is dropped. So, this is called block call cleared.

So, you have been blocked because all the servers are busy. So, your call has been cleared. So, that scenario is called. So, what associated system what that system be? The system will have no queueing place. So, it will just have a server everybody arrives occupies a server, if all servers are occupied there is no place for keeping them in the queue. So, they will be blocked.

So, associated system if you see that will be Markovian arrival of course, Markovian service of course. How many servers are there? m number of servers are there. And how many total places is there? Only m server no queueing place. So, this is also m. So, this

is called the M M m m queue. Of course, the last column is infinite because you have already assumed that an infinite customer base or infinite population is potentially or they can potentially make arrivals.

So, this is that system and it is generally being characterized by the Erlang B model we will see ok what is that Erlang B model. The other scenario is where you might be waiting in a queue, so that is called block call delay. What does that mean? That means if you are blocked if all servers are occupied then you can wait in the queue whenever the next server is free you will be given connectivity; that means, you will have to now wait. So, there is a waiting time ok.

And what will be the waiting time that depends on the kind of calls that are going on and all those things all kinds of system parameters, all kinds of arrival characteristics. So, everything will determine what kind of waiting or waiting statistics will be happening average waiting time that will be happening. So, those are the things we will have to analyze.

So, in block call delayed system what is a system associated system? So, you have Markovian arrival again Markovian service m number of servers, but after that, I do not have to specify anything because there are infinite places for waiting. As many customers are there we have already said infinite number of customers they all can potentially wait they can all make calls only m of them will be occupied means m of them will be given connections all the rest can be waiting.

So, therefore, the waiting place can be infinite. So, therefore, the number of customers that that system can hold also goes to infinity. So, I can just say M M m queue rest of the things are infinite and this corresponding formula is called the Erlang C model or Erlang C formula ok. These are the two systems that now we will be interested in analyzing. So, these two systems we will be now trying to analyze.

So, let us first try with this one M M m system. Let us try to see how I characterize the Markov chain associated Markov chain because if I can construct the Markov chain, if I can see the transition rate transition matrix then the rest of the things are done we just have to solve some set of linear equations like we did for M M 1 queue. So, that is all we will have to do if we have to do it ok?

So, let us see M M m queue this Erlang C model or block call delayed where you can actually wait. So, for that how do I construct a model? Because it is Markovian arrival and Markovian service so basically the state description nothing changes. How many customers are there? There will be always again some integer number of customers and nothing else is required.

So, Markov state will be similar 0 1 2 3 it just counts how many customers are there in the system ok m minus 1, m, m plus 1 and so on it just goes up to infinity. This is also because I have an infinite buffer. So, it also goes up to infinity it is not a finite thing. Now, let us try to see the rate matrix ok or the transition rate. Over here arrival is happening with parameter lambda. So, lambda is the arrival rate from 0 it will be the arrival rate if there are 0 customers; that means, nobody is there no circuit has been created. So, nobody is present in the trunk switch.

So, therefore, the first customer arrives with arrival rate lambda. From 1 what will be the arrival rate of the next customer? That also is lambda because with lambda arrival rate only the arrival happens irrespective of how many customers are there inside the system because there are infinite customer base outside. From there 1 or 2 customers you take inside it does not matter there are still an infinite number of customer base they will be again populating with rate lambda only.

So, it happens to be all lambda. What changes are over here? If there is 1 server let us try to now understand the situation with a time diagram. Let us take just two servers ok. So, what is happening is the 1st customer arrives he goes into this ok, and the 2nd customer arrives he also goes into 2nd server because whenever the server is free customer will be given the server ok only the 3rd customer arrives he waits over here ok.

So, timeline if I see I have server 1, I have server 2 and this is the queue. 1st customer arrives immediately goes to server 1 and gets his service this is the service time probably. The 2nd customer arrives so this goes into server 1, 2nd customer goes because server 1 is busy. So, he can go to server 2.

Now, if the 3rd customer arrives. So, for him both the servers at this instance are busy. So, therefore, he has to wait. While waiting now let us try to see how many systems are there in this and how many customers are there in the system. So, overall 1 2 3 customers are there both the servers are occupied.

Now, when he will be actually going into service, whenever one departure is happening either this departure or that departure whoever is earlier; so, how do I say when the departure will be happening that is the rate with which one departure will be happening; that means, the state will be coming down.

So, how do I characterize what is the departure rate? So, over here there is an exponential distribution with parameter mu we know that this residual times. So, he has arrived over here from there he will be counting this is again exponentially distributed with parameter mu, this is also exponentially distributed with parameter mu.

So, it is actually a minima of two exponential distributions with parameter mu both the one has having parameter mu. So, therefore, the minimum of this distribution we have already characterized; will be the addition of those two. If there are 3 then it will be the addition of those three. So, therefore, the rate will be mu plus mu it will be 2 mu. If there are 3 it will be mu plus mu plus mu 3 mu.

So, now let us try to see when there is one server busy. So, one means what? One server is occupied. So, what will be the corresponding rate? It will be mu. Over here when he is in state 2 how many servers are busy now? Two servers are busy because he is in 2 states means the state 2 there are m servers he is in state 2 means two servers should be busy. So, what should be the departure or departure rate or rate of going out from the system? That should be now associated with 2 mu because two servers are occupied over here it should be 3 mu and so on.

So, basically, even if this customer does not come from the observer's point of view at this time instance I try to see what will be the rate of departure. So, right now two customers are there. So, my state is 2. So, from 2 when it will become 1, that is what I am trying to account for over here from 2 it will go to 1; that means, one of them will be departing.

What is the associated rate? So, that is minima of these two exponential distribution. So, that should be 2 mu. If there are 3 server 3 are occupied then minima of 3 exponential distribution that is associated with 3 mu. From this time instance so suppose I am observing what is the possibility that one arrival will happen before a departure. So, basically that is coming with lambda. So, that is why it is it is going with lambda. So,

with lambda rate it is actually going to upper state and with 2 mu or 3 mu depending on where he is he is coming down.

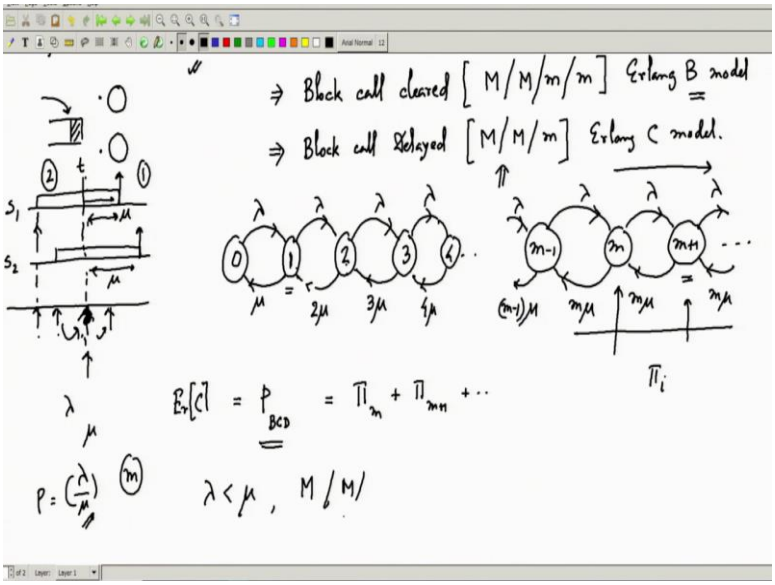
So, now everywhere let us say if it is 4 then also it will be 4 mu because there are 4 servers up to m this will be happening this will be continuing m mu. When it goes to m plus 1 what happens let us try to understand that scenario that was the scenario I was depicting over here. So, there are two servers already occupied 3rd customer has come for him how do I know? So, 3rd customer has come already with lambda rate. So, it has gone to state probably 3.

But from there I now try to see what is the departure rate. So, at that point only for two server I will be accounting. So, therefore, at m plus 1 also only the server I will have to account for because the m server was busy m into mu rate I can see that one server will become free. So, therefore, this will still remain m mu and all the higher ones will still remain m mu because there are only m number of servers. At max with m mu rate he can serve. So, therefore, the departure rate will be at max m mu when all m are all m are occupied.

So, all these state from m onwards all m server must be occupied because if m plus 1, 1 must be in queue all m servers are occupied, m plus 2, 2 must be in queue and all m servers are occupied. So, therefore, all these places' service rates will be mmu only from here onwards it will be reduced. So, as you can see with this simple description of understanding of the queueing we can actually construct this whole Markov chain.

Now, once I have constructed this Markov chain now I just have to write down all the equation governing equation and then solve them recursively. Whenever I solve I get my whatever things I want to characterize. So, over here I am interested in this what is my blocking probability because overall blocking probability or probability that I will be delayed? So, that probability I want to calculate. Let us try to see how do I that is the Erlang C formula is and how I characterize it. So, when what is the probability that I will be delayed?

(Refer Slide Time: 18:18)



If I am in state m means I am an arriving customer if I see the system in state m like this 3rd customer. So, there were two servers already occupied if I arrive I will be blocked. So, basically while arriving I see the system in state m or state as many servers are there in that many states. So, that or more than that then always I will be blocked, all these state probabilities if I add them must be the blocking probability or call delay probability. That is the probability that my arriving customers will be delayed.

So, therefore, this pi m plus pi m plus 1 up to infinity is my probability that the block call will be delayed or this is actually called the Erlang C formula. So, all I have to do is now evaluate this pi i's like we did for M M 1 queue and then sum them from m to infinity and that should give me the overall blocking probability ok.

So, basically what I can now see that I will get that should be definitely dependent on the traffic parameter which is lambda and mu or I can say lambda by mu this rho this should be associated with this and it will also depend on this value m the characteristics of my trunk switch how many trunks I have given this m.

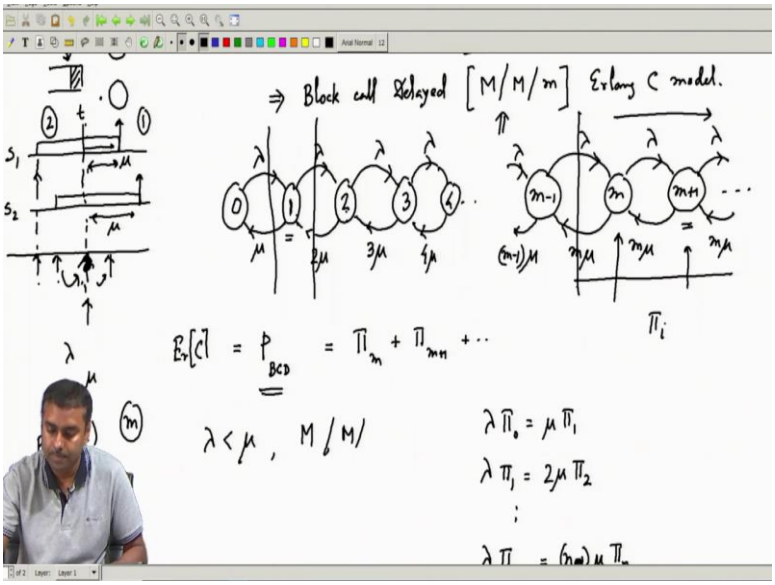
What will be happening I will be getting a blocking probability with respect to this rho and m. Now, m is my design parameter. When I try to design a trunk switch I have to give how many output links I will be giving because that is a costlier more number of links I give will have more cost. So, what I try to design is some targeted probability of

this BCD or block call delayed probability that call will be delayed. So, that probability for a particular arrival and service rate I will be designing a m.

So, this is the design that we have targeted earlier. In the last graph, we were targeting this kind of design also, but that was done in a more average analysis over here we are actually doing an exact analysis ok? So, this is a very exact analysis because we have developed the whole theory and these particular probabilities are actually the overall probability that will be happening that we will see in the if as long as the Markovian assumptions all the assumptions that we have taken are true.

That means lambda is below mu for stability condition then it's Markovian arrival and Markovian service these two things are happening and the amount of customers they are coming they are almost going towards infinity. So, if these assumptions are true then these formulas are accurate stochastically, and the average value that will be getting will see that in a real system whatever is happening the average values also will be accounted for. So, let us try to see how I analyze that.

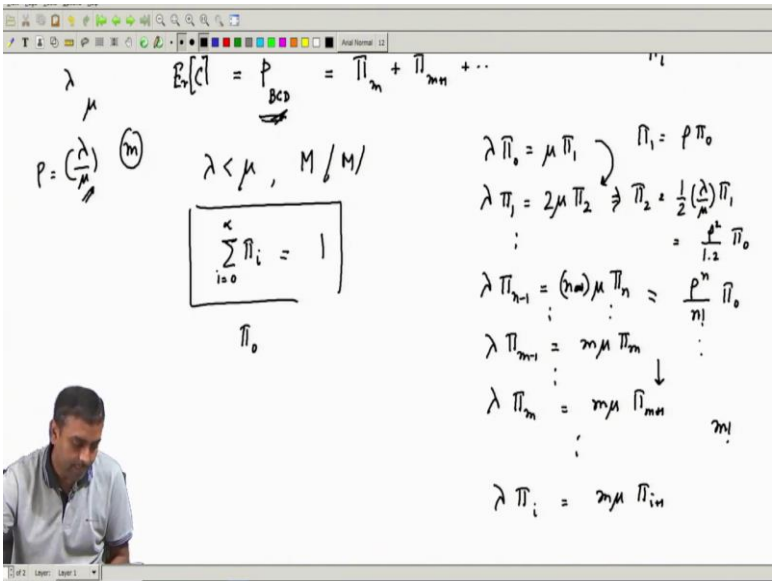
(Refer Slide Time: 21:42)



So, over here I can again do the same thing. So, from this 0 state if I consider what is the outgoing rate and what is the incoming rate. So, I can write lambda into pi 0 is equal to mu into pi 1 the 1st equation remains the same only the 2nd equation changes. Now, what happens lambda into pi 1 becomes 2 mu into pi 2 and so on. So, basically the nth one what will happen? Lambda into pi n minus 1 that should be n minus 1 into mu sorry

n into mu pi n ok remember this happens up to m. So, this keeps on going up to m this is true. So, over here this is true.

(Refer Slide Time: 22:50)



So, if I just put it lambda into pi m minus 1, but beyond this the formulas gets changed little bit. Beyond this if I just put the formula of as you can see this will still remain m mu. Earlier this coefficient of mu was increasing, but this is where it gets fixed and so on it just goes on. So, any lambda into pi I will become still m mu into pi I plus 1 or something like that ok?

So, that is the set of equations that is being constructed and it will keep on growing ok. So, now, if I just keep substituting then we will be getting our associated formula ok. So, if I just do those things and then write a summation this pi I; i goes 0 to infinity if you do this summation that should be 1 I put that everybody I will be able to represent with respect to pi 0 right? So, from here pi 0 I can solve and put the pi 0 over here and then I get my solution.

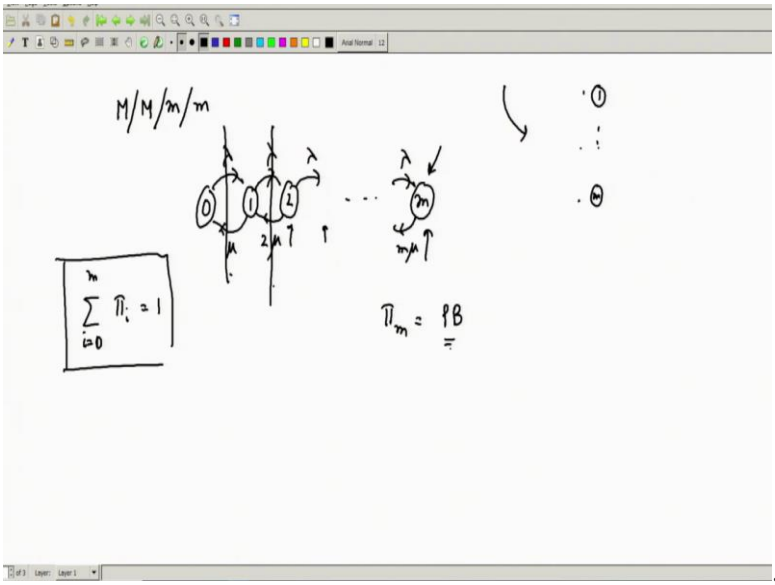
So, just I will not do that you can take that as homework or your assignment will have this. So, all that will be happening you can see over here is pi 1 becomes lambda by mu that is rho into pi 0, pi 2 happens to be 1 by 2 lambda by mu again into pi 1 or I can write this again this is rho another rho. So, basically, the rho square is divided by 1 into 2 pi 0.

Similarly, if you do this will be rho to the power n divided by all this 1 2 3 4 factorial will be coming. So, n factorial pi 0. So, this goes on ok. Only thing is that from here onwards it will be slightly changed ok. So, rho to the power n will be happening, but this will no longer be factorial m ok. So, this will not be factorial m there will be a factorial m part and then this m to the power rest of the things ok i minus n that will be happening.

So, you can construct this very nicely then you write this then you get a summation term in pi 0 you solve that you get your all the corresponding pi 1, pi 2, pi 3 every value and then all you have to do P BCD will be from pi m to infinity you take the summation replace it will be represented with respect to pi 0; pi 0 from the normalization equation you will be getting.

If you substitute that you get your Erlang C formula whatever that formula is you can take that from the textbook ok that will be there in your study material also. So, you can derive that formula that will be also coming in the exercise. So, you will be able to do it. So, only thing that is left let us try to see if I can do the Erlang B formula now.

(Refer Slide Time: 26:28)



What is that system Erlang C? Only servers are there m servers are there 1 to m nothing else no queue all arrival happens they are just getting distributed to the server. Corresponding Markov state and Markov chain what will be happening? Now, this has no queueing place. So, it is a M M small m and how many customers can be total maximum taken in the inside the system, only small m because only small m places are

there where customers in the server they can be attached to and get service to no queueing place.

So, basically, the system is now truncated the Markov chain is no longer an infinite chain. It goes from 0 the state can go up to m. So, 0, 1, 2 up to m only that is it; it does not go beyond this. So, it is a finite Markov chain. Arrivals are again lambda accordingly. What is this departure rate? That should be when the same logic if one server is occupied that should be mu, two servers occupied 2 mu, m server occupied m mu that is it; it is as simple as that.

This Markov chain I will have to analyze this again I can put the total outgoing and total incoming rate of probability balanced I can put that equation can solve them similarly and I get my all-the-state equation. Remember over here the normalization equation will be finite because it has finite state. So, I will go from 0 to m only pi i which must be 1. So, this normalization equation will be finite normally means finite summation will be there ok.

So, from there you will get relationship means all this equation you put you will be getting relationship with respect to pi i to pi 0 and then put it over here pi 0 will be solved and then you get all the state equation. Now, let us try to see what is the blocking probability from here how do I calculate the blocking probability?

Blocking probability is very easy over here as you can see any other state you are there if you actually arriving customer comes to any other state they are not blocked because if he arrives when two servers are occupied he will just be the customer will take it to the 3rd server and he will be already getting the clock. So, he means server. So, he is not getting blocked only when m are occupied. So, whenever all m servers are occupied at that time only your blocking will be happening.

So, this pi m is the blocking probability. So, and that is if you solve this calculate the probability of mth state that pi m is your blocking probability and that is called the Erlang B formula; that means, block called are now means they are cleared. You do not actually whenever call is blocked you do not give them chance they are cleared they can do arrival later on next time, but at this time they will just say sorry ok all trunks are busy.

So, again if your system is like that you do not allow them to wait then this is the blocking probability and accordingly you can actually calculate the blocking probability it will be again related to your m and the rho. So, from the knowledge of rho because that should be traffic that should be characterized from the knowledge of rho and a targeted blocking probability you will be always able to decide m. So, this is famous Erlang B formula. Again you can derive it I am not deriving it; it is just algebraic manipulation. So, we will be giving assignment for that ok.

So, you can do that and you can always check those formulas those are available anyway. So, Erlang B and Erlang C formulas these two formulas are heavily used for trunk designing and we have seen that how queueing theory can be applied to derive these two equation and further design your trunk switch. So, with this we will end our queueing theory discussion right now. We will go into now basically this has given us all the analysis of the corresponding circuit switch network.

Now, we will launch ourselves towards a packet switch network which has happened a paradigm shift happened. So, we will see that little bit of initial introduction how that shift has happened, what was the transient period, where this transition was happening. So, those things we will discuss and then we will again maybe come back theory of queueing and the analysis of that for those packet switch analysis as well later on ok.

Thank you.