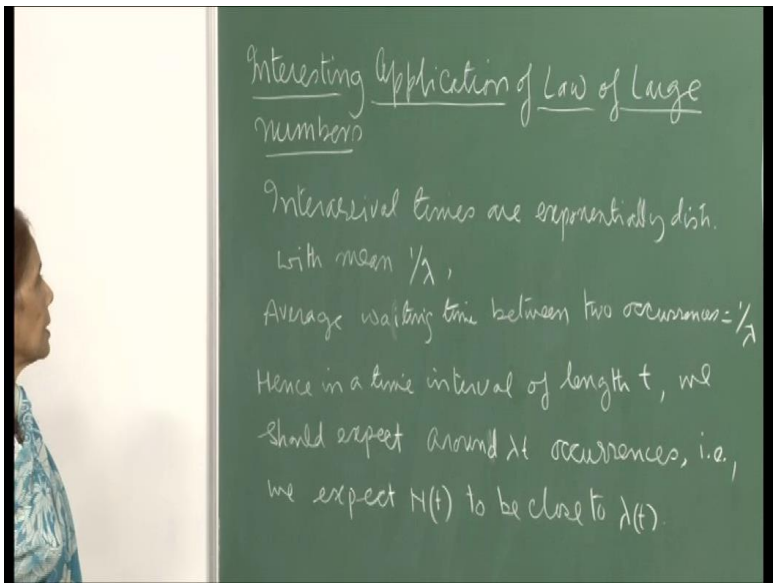
Introduction to Probability Theory and its Applications
Prof. Prabha Sharma
Department of Mathematics and Statistics
Indian Institute of Technology, Kanpur

Lecture - 34
Queuing Models M/M/I Birth & Death Process Little's Formulae

(Refer Slide Time: 00:15)



So, I had to wait for this example, you know to discuss this example, because I had not worked on the when we talked of weak law of large numbers and strong law of large numbers, I have not, you know talked about the Poisson process. So, I waited till I had, you know introduce the topic of Poisson process to give you this example. In fact even when we are talking of law of large numbers I had shown you some examples. So, this is also one of them; one of interesting example.

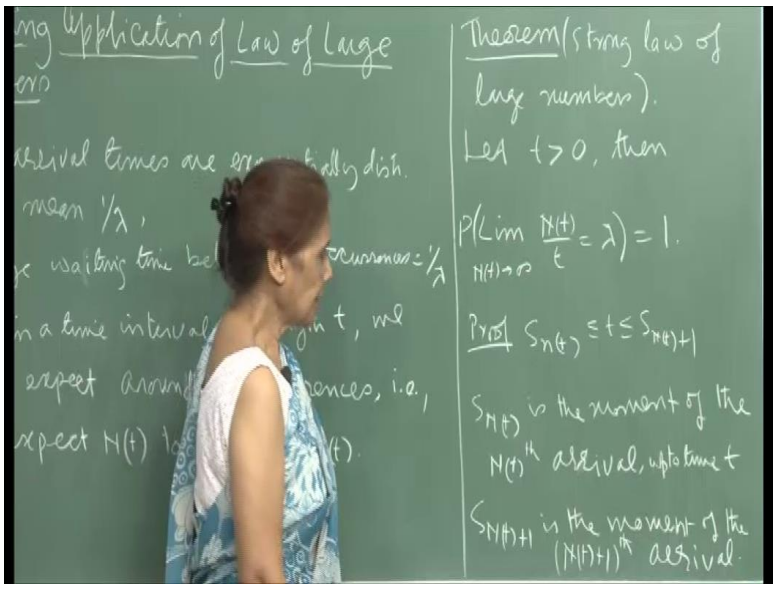
Now, here you see inter arrival times are exponentially distributed with mean 1 by lambda, right; because the arrival rate is lambda. So, we have shown that the inter arrival times will be exponentially distributed; and the mean time would be, that means, inter arrival time would be 1 by lambda. So, that means average waiting time between 2 occurrences is 1 by lambda; and so the number of arrival, mean arrival weight is lambda, right.

So, hence in a time interval of length t we should expect around lambda t occurrences, right; if lambda is a mean arrival rate, so therefore per unit times. So, therefore, time interval t; you were expect on the average lambda t occurrences, lambda t arrivals, right.

So, then and since our notation for the Poisson process for the number of arrivals up to time t is N (t). So, therefore, we expect that N (t) and lambda (t) should be close, and this is what the weak law of large numbers and strong law of large numbers is all about.

So, let us just look at this, and we will show that, yes, the ratio of N (t) by, this will be N (t) by t would be close to lambda, right; because if you want lambda (t) to be close to, N (t) to be close to lambda (t) then N (t) upon t should be close to lambda; this is the whole idea.

(Refer Slide Time: 02:27)



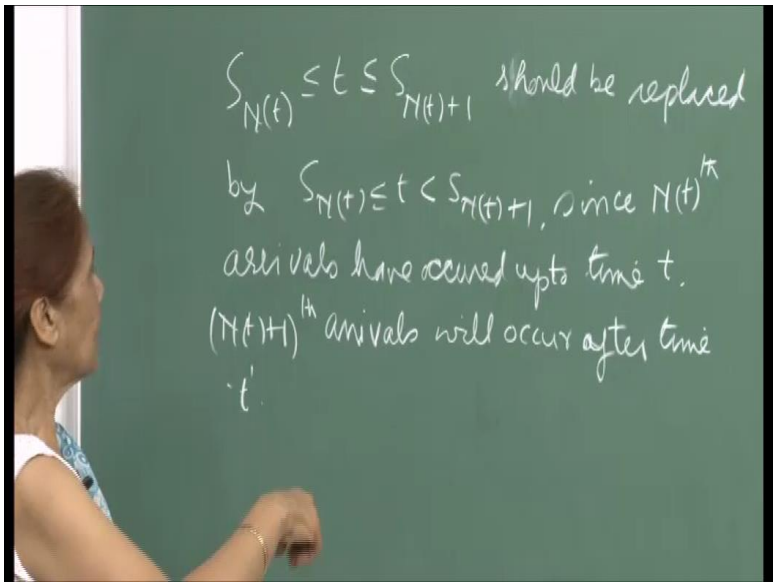
So, let us look at the proof, interesting. Now here, I should have written the word proof here. Now, let t be some positive time, right; and then we want to show, I said it is the, we want to show that the limit of N (t) upon t, N (t) goes to infinity is equal to lambda. So, this probability is 1; that means, this is the certain event. So, as you take the limit and your law N (t) to grow large, then your ratio N (t) by t would be lambda. Now let us look at the proof.

So, see, this is the thing; now for the Poisson process when we are looking at the arrival process, and so on, then my S n (t) is the movement of the N (t) th arrival upto time t. This is what we have been denoting. And later on when I discuss the death birth and death process at that time S n (t) was the, you know, time arrival, because remember I was looking at the waiting distribution for the waiting time in the q; and then I also used the symbol S n (t), but that time it was the waiting time for the n plus 1 th arrival; that

means, S n (t) denoted the time at which the N (t) th service got completed, right. So, here it is S n (t) is the movement of the N (t) th arrival upto time t.

So, I hope the reference to the context the things will be clear because here we are only talking of the, and I have not introduced the waiting time and so on, upto this point. So, therefore, it should be ok. So, S n (t) is the movement of the N (t) th arrival upto time t and therefore, S n (t plus 1) will be the movement of the N (t plus 1).

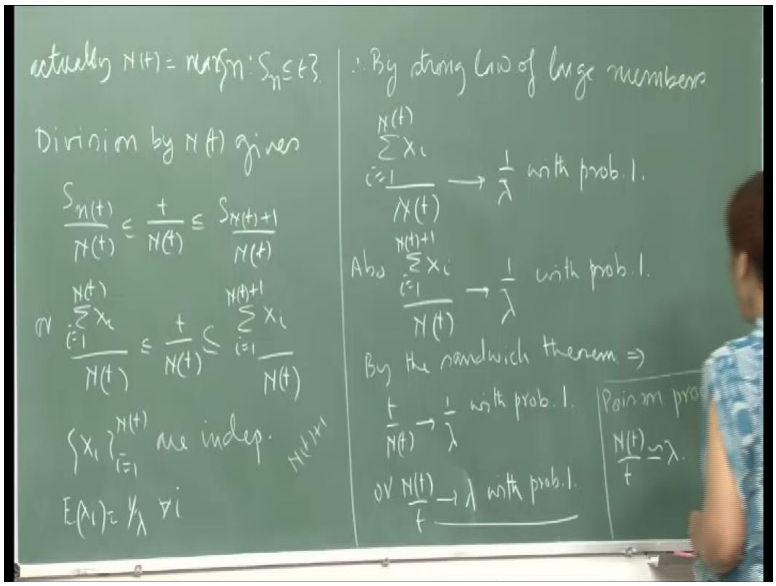
(Refer Slide Time: 04:17)



So, the inequality that we are writing here, S n (t) less than or equal t less than or equal to S n (t) plus 1, should be actually replaced by S n (t) less than or equal to t and strictly less than S n (t) plus 1. Since N (t) th arrivals have occurred upto time t, and n S t is the time of the N (t) th arrival, and so therefore, when the S n (t), when the N (t) plus 1 th arrival occurs that will be the time n S (t) plus 1. So, that will have to be bigger than t.

So, want to make that clear, and that is why this should be replaced by the strict inequality here; that is because we say that S n (t) is the time at which the N (t) th arrival is occurred, and upto time t; and so S n (t) plus 1 the time arrival for the N (t) plus 1 th arrival will take, will be more than t, right after t.

(Refer Slide Time: 05:21)



In that arrival; in fact, I would understanding is that, you know, N (t) is the max of n so that S n is less than or equal to t; so that means, upto time t we do not expect, there no more arrival then N (t); and therefore, this inequality is valid; that means, t is greater than are equal to S n (t), but if there is one more arrival then certainly that time will x e t; this is the whole idea, right.

So, upto time t, S N (t) is the number of, the movement of the N (t) th arrival this should be N, sorry; should write here is this is N; so that means, the time of the N (t) th arrival has to be less than or equal to t, but N (t) plus 1 th arrival will exceed the time t. So, this is the understanding, right.

So, with this understanding you now divide the both the inequalities by N (t), and therefore, you get S n (t) by N t less than or equal to t upon N (t), this is less than or equal to S N (t) plus 1 upon N (t), right. Or, remember the x size are the inter arrival times. So, therefore, you, S n (t) will also be equal to x 1 plus, x 2 plus, x n t. So, when you add up the inter arrival times they will all add up your S n t.

So, sigma i varying from 1 to N (t) sigma x i divided by N (t), and this is less than or equal to t upon N (t), then this is, this, here the summation will go upto N (t) plus 1. So, you will add up the inter arrival time for the N (t) plus 1 of the arrival, upto this thing. So, N (t) to N (t) plus 1 th, the arrival this will be this, right.

Now, sigma x i s are independent, remember we have shown this; we get the poisson process, Poisson arrival process; then the inter arrival times would be exponential distributed; and they are independent identically distributed, each has the mean 1 by lambda. So, then the conditions for your law of large numbers is, are satisfied. And therefore, by the strong law of large numbers, sigma x i vary from 1 to N (t) divided by N (t) will converge to 1 by lambda with probability 1. This is our strong law of large numbers.

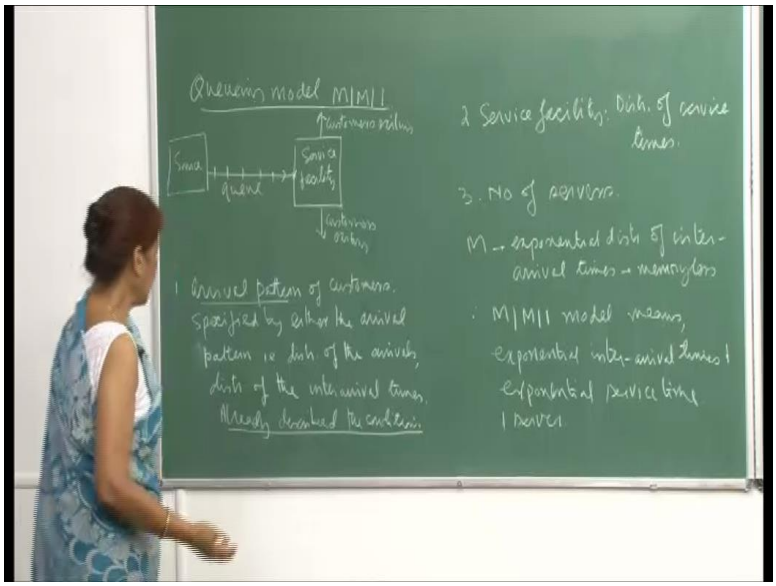
And since, this is also the same series, you know, N (t) plus 1, but you are allowing this go to infinity. You are allowing N (t) to go to infinity; so therefore, this and this have the same limit which is 1 by lambda, right, the mean of x i with probability 1. And so by the sandwich theorem, because here, you see this is converging to 1 by lambda, so t by N (t) has no choice, it has to converge to 1 by lambda.

And so therefore, by the sandwich theorem, limits t upon N (t) will converge to 1 by lambda with probability 1; or N (t) upon t will, because we have taken t to be positive, so N (t) by t will converge to lambda with probability 1. So, therefore, you know, for the Poisson process; so now, again as I told you that the situation for the law of the large numbers is basically used to estimate the mean of the population.

And so you go on making observations, and we say that the observations are independent identically distributed because they are coming from the same population. So, then the average, we expect the average to converge to the mean of the population. So, here also the same thing for the Poisson process; what we have shown is because N (t) is the number of arrivals in time t, so this ratio will converge to lambda, the mean arrival rate.

And therefore, you can go on observing the values, number of arrivals in a particular time, and then upto time interval t, and the ratio will converge to lambda. So, if in case your lambda is large then you will have to make the observations for a longer time period because your N (t) will have to be sufficiently large; and therefore, that of course, make sense. So, therefore, this gives you a good way of wanting, trying to estimate the value of lambda.

(Refer Slide Time: 09:40)



So, the Queuing model I am going to talk about today is M M 1, it is called M M 1, and I will explain in a while why we call it M M 1. So, here the whole idea is that you have a source from which your customers are coming to some service facility; there is a queue. So, these indicate the customers who are waiting in the queue for to be serviced; you have a service facility. And then again it will depend on what kind of service facility you have. And so the customer; so one by one a customer comes here, get serviced, then exists from the system after his service, his or her services is completed.

Then the next one in the queue comes to be; so this is I am describing the situation when there is only one service facility or one clerk at a counter or something; if there are more than one, then of course, the movement one of the clerks is free the, a person waiting in the queue will go and get serviced, right. So, whatever it is, the service facility, the customers come, they get serviced; and once their service is complete we expect that they leave the system; so they are out of the system.

Now, here, in order to discuss the model we first, so we need to specify the arrival pattern of the customers, right. So, this can be either specified by the arrival rate and the distribution of the arrivals because remember the whole situation, the thing, that the scenario is that the events are unpredictable. So, we do not know when a customer will walk in. Also we have no idea, the service times are also unpredictable.

So, therefore, everything has to be modeled through these probability distributions. And so we will either specify the distribution of the arrivals, just as in the Poisson process we

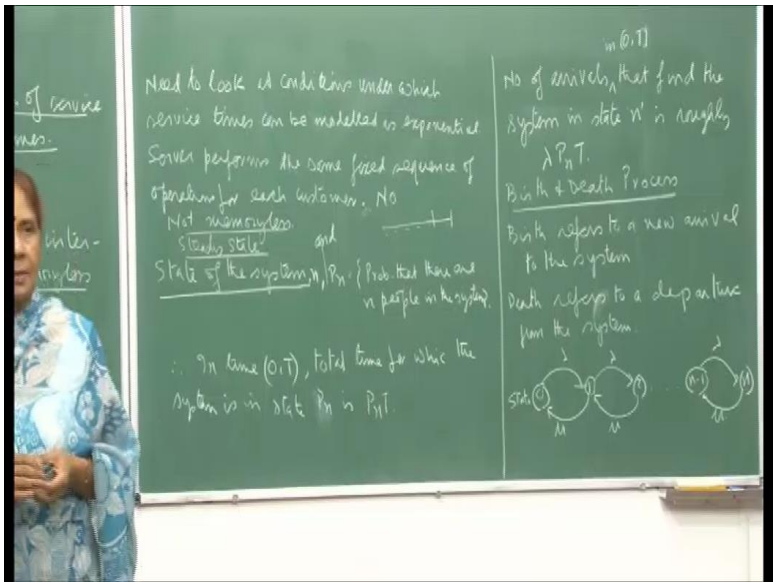
say, that the arrivals are coming at a rate whatever the rate is lambda, and then they are being modeled by the Poisson distribution; and or we give the specify the distribution of the inter arrival times which we saw that if the arrivals are following a Poisson distribution then the inter arrival times will be exponentially distributed.

And then we had, in the last few lectures we have talked about in detail what, under what conditions we can say that the arrival pattern can be modeled by the Poisson distribution, right; so stationary increments. And then independent increments and so on; and then that the probability of arrival in a small interval would be only lambda times the length of that interval, and so on. So, there are huge lot of conditions under which we said that we can then model the arrival pattern by the Poisson distribution, right.

Then you have the service facility; and here of course, you can specify the distribution of the service times. As I said that it is not fixed operation each customer may take different amount of time, and so on. So, we have to; and then of course, you need to specify the number of servers. So, basically if you have these 3 things specified then your queuing model is there; and the M denotes the exponential distribution of inter arrival times, memory less.

This is the property or markovean which we will again when we later discuss markov processes, we will see that Morkovean process is also have memory less property, right. So, inter arrival times are exponentially distributed; and x service is also, service times are also exponentially distributed. So, therefore; and then one server. So, first we will discuss the case when the, this is only one server at the server facility and latter on we will try to generalize the, now s is to more than one server.

(Refer Slide Time: 13:58)



Now, just tells the specified the conditions under which we can model the arrival pattern by the Poisson process, we need to look at conditions under which service times can be modeled as exponential distribution, by an exponential distribution. Now, if your server is performing fixed, some fixed sequence of operations for each customer, then certainly this is not memory less.

Because, if the customer, if these clerk has to perform 10 operations, sequence of 10 operations for every customer then he is up come up to the this thing, task, then we will know that he is going to finish after next 2. So, the sort of, one can assess the time taken for example, if a server has come up to this point, I mean this task is performed, then we know that he will finish these 2, and so the time at which this service will end depends on how far he has been already with the customer; how far he has been servicing the customer.

So, therefore, they are certainly not a case for modeling by exponential distribution, right; because this is some sort of a fix sequence of operations. So, therefore, here also we will have to be, basically it will have to be the memory less property. If you can somehow justify that the service facility of the situation that you are modeling has this property then it will be, you know, safe to say that yes we can model the service times by the exponential distribution, and so on, right.

So, then and the state of the system we will always specify by the number of people in the system; and then P n will be the probability that there are n people in the system. So,

you can see that it is people coming for service, after being service they leave the system. So, it is, you know, our constant state of changing, because people come and go. So, P n is a probability that there are n people in the system. So, therefore, in time, (0, T); yeah, here I should I have underlined this, but I sort of missed it right now. So, the whole thing is being discussed under steady state situation.

So, now, what we are saying is that suppose there is a new restaurant that is opened in the locality then you know, the number of arrivals would vary from day to day, and there will be no set pattern for some time till people sort of get used to that restaurant or they make it a habit of whatever it is; and there are steady number of customers who comes to the place to the restaurant.

So, therefore, what we are saying here is that the we are discussing all this, when the system has settled down the tabulations are all over, and it is only steady state; that means, the probabilities have also settled down, and so on. So, under steady state we are discussing the modeling of this, modeling the queuing the situation, ok.

So, therefore if P n is the probability that the n people in the system, but in time (0, T). So, this total time for which the system is in; see, you can also look upon P n as the proportion of time for which the, for a unit time, proportion of a unit time in which this system is in state n; that means, the n people remember because this is the probability, and so this is the fraction, and therefore, we are saying the fraction of time that people, the n people in the system, right.

And therefore, over the time interval, (0, T), we will say that the total time for which the system is in state n, sorry, is not p n; is in state n, it is P n T, right. So, approximately we will say that proportion of time that there are, in this time interval proportion of time for which there are n people in the system is P n T. And therefore, and number of arrivals in (0, T) that find the system in state n is roughly, lambda P n T.

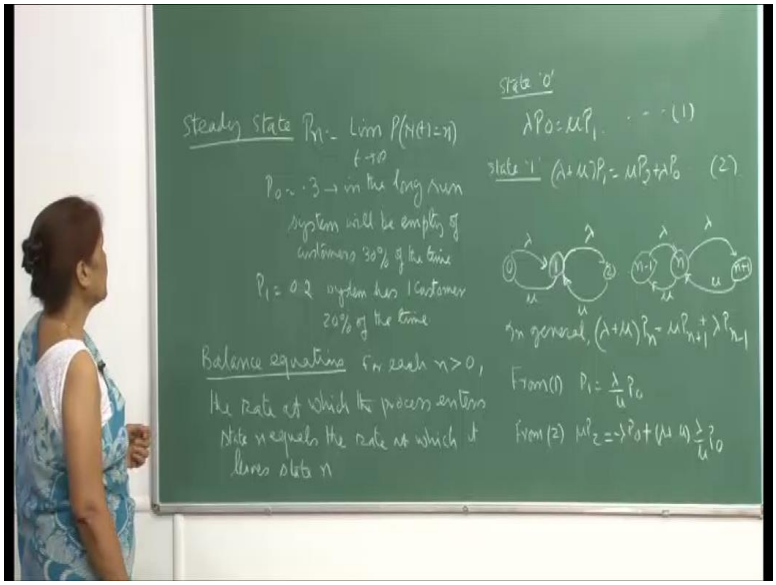
So, we are talking in approximations and right, because the arrival rate is lambda. So, the number of arrivals in (0, T), that find the system in state n would be, lambda P n T because they are lambda arrivals in a unit of time. And so I mean the arrival rate is lambda. So, and this is also called a birth and death process because birth refers to a new arrival to the system and death refers to a departure of from the system. So, therefore, each the departure is treated as a death and each arrival, new arrival is treated as a birth. So, this is also called as a birth and death process. And so this birth and death process

under the assumption that your arrival rate is a, process is Poisson, and the service time the exponentially distributed, there is one server.

So, if you, diagrammatically you can describe the situation here, of the birth and death process. You begin with state 0, no people in the system, then 1 arrival takes place; should be lambda. And so you go to state 1. But from state 1, you can reward back to state 0 if the departure, and that is for this. So, we are saying that the exponential, the service time is exponentially distributed with rate mu, right.

And then again when you are in state 1, it can go to state 2 by arrival and can reward back to state 1 because if there is a departure, and so on. So, that means, at each state, n minus 1, for example, you can go to the next state, and from this you can reward back to the old state; and so therefore, this make sense that you will, this proportion of time you will be in state n, right; because the situation keeps changing. So, let us further analyze the, you know, arrival pattern, mean arrival, average arrival time, average waiting time, and so on.

(Refer Slide Time: 20:17)



You see, when I made the statement that we are considering the system, the queuing system in steady state, so we actually I just remain that this limit probability of N (t) equal to n as t goes to infinity is P n. So, this is steadying down. And of course, t going to infinity is the analytical we are saying it, but essentially for a large time this system has operated and then it is settled down to steady state, that is what you mean. So, these are the limiting probabilities essentially of the system.

So, and then for example, if you say that P 0 is 0.3, then in the long run, system will be empty of customers 30 percent of the time, right. Again, you know, because these are all probabilistic statements; what we are saying is that if your P 0 is 0.3, then long run if you observe the system then you will find that 30 percent of the time the system is empty. And that is what we meant when I said that P n T is the proportion of time.

So, this is again in the long run when you, P n T will be, approximately we proportion of time for which the system has n people in the system, the system has n people, n customers and users whatever it is. So, this is the idea. So, and similarly if P 1 is 0.2 then the system has 1 customer, 20 percent of the time, even if you observe it for a long. And so approximately for time t, we can approximately say that this is the proportion of time that the system will have n customers, fine.

Now, we want to start getting some more, you know, we want to get some, make some computations regarding this queuing system. And so we will use this concept of balance equations. What we mean is therefore, each n greater than 0, the rate at which the process enters the state n, equals the rate at which it leaves state n. So, this is also part of the system that, condition under which we are modeling the situation or the process.

So, what we are saying is that the balance is maintained. In other words, what we are saying is that if you are state 1, you see, then you are leaving it here, by, because 1 more arrival has come; or you are leaving state 1 because 1 person has been serviced. So, this is how state 1 changes; either 1 more arrival, or 1 death or 1 percent leaving the system. And then again the way state 1 is reached is also from state 0 when there is a one arrival at this, so then you come to state 1.

And here again, you come from state 2 when there is a departure here at this point. So, this is the idea. So, at each state of the system you have; so for example, when you are at state 0 then this is the rate at which lambda P 0. So, this is the rate at which the system leaves the state 0, right; because it is state 0 then lambda arrival; I mean the mean arrival this is, I should not say mean; the rate, arrival rate is lambda; therefore, lambda into P 0. This should be, this is rate at which it will leave the system.

See, the system right now is in state 0, so it will leave, the system leaves that state at the rate, lambda P 0. And from P 1 it arrives to state 0 at the rate of mu P 1. And therefore, the 2 must balance. So, the rate at which it changes its state from 0 and arrive at 1, and the rate at which it leave state 1 and arrives at 0 is mu P 1. So, the 2 must equal.

Now, if you are in state 1 arrival describing you, then you see it is the 2 ways it can leave, either 1 more arrival or 1 departure. So, therefore, lambda plus mu into P 1 because remember that 2 processes we assumed are independent; service process and the arrival processes are both independent. So, therefore, I can add up the these rates. So, and this will be lambda plus mu into P 1. This is how it will leave the system P 1, the state P 1.

And, if it is P 2 then it can again come back to P 1 at the rate of mu P 2, and here from P 0 it can come to P 1 at the rate of lambda P 0. So, therefore, we departure from state 1, the arrival to state 1, the rate at which these 2 things happen must be the same, right. And so in general again same thing, that n for example, you are leaving it again because this arrival and you are leaving it because there is a departure.

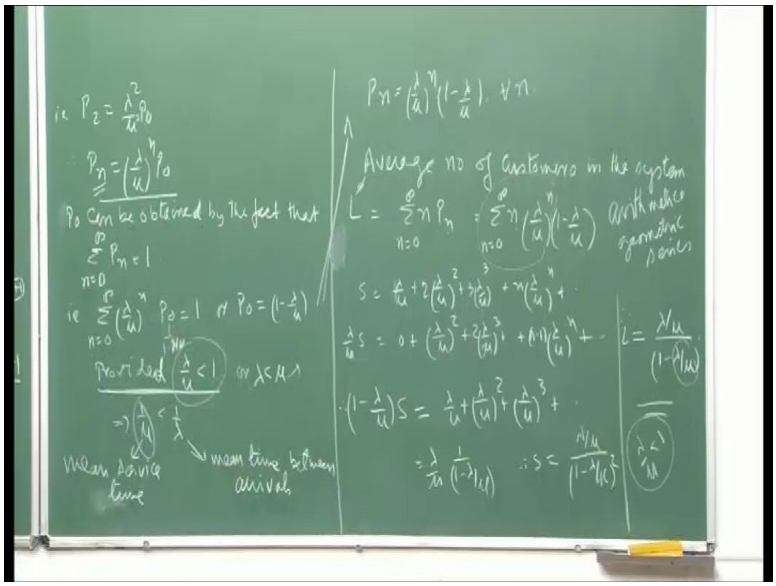
So, therefore, these 2, lambda plus mu; and then you are coming to state n through n minus 1 at the rate of lambda P n; and then you are, sorry, P n minus 1, lambda P n minus 1. And then here you are coming from n plus 1 at the rate of mu P n plus 1. So, therefore, in general you can write this.

Now, here of course, if I am only considering a very simple form here because you can also have a situation where your lambdas are also dependent on the people in the system. But, we are, see, because that can happen some places where it is not a very essential service if the places crowded. For example, if a restaurant, people may not want to wait and they will leave the place because you can go elsewhere to eat, right.

So, then you are, lambdas would be the arrival rates, would also be dependent on what state the system is in. And similarly the mu s can also depend on your, the number of people there are in, the customer they are in the system. So, these can be different for different states of the system, but I am right now considering the most basic case where all the lambdas; so these are not dependent on the number of people in the system; similarly the mu is not dependent on the number of people in the system. So, the service rate continues at the same P's.

So, now, if you solve these equations; see here, immediately you get the P 1 is lambda by mu P naught. So, let us get all these P's in terms of P naught. And then similarly from here if you substitute for P 1 from here then mu P 2 would be lambda plus, mu into, lambda by, mu P naught minus, lambda P naught, this goes this here.

(Refer Slide Time: 27:45)



And then if you simplify you get the P 2 as lambda by mu whole square into P naught. So, in general your solution to these equations, these balance equations is P n is equal to lambda by mu raise to n times, P naught. So, all, for all n this will be the formula; that means, this just lambda by mu raise to n P naught, right. Now, we can obtain P naught by using the fact that all these probabilities must add up to 1, right.

The system must be in one of the states from 0 to infinity. And therefore, when you add up this you get this as a geometric series; P naught is outside with common ratio lambda by mu. And so P naught is 1 minus lambda by mu, because this sums to 1 upon, this series sums to 1 upon 1 minus lambda by mu, right; so therefore, P naught for the 1 minus lambda by mu.

Now, first this is valid only, this series converges provided your lambda by mu is less than 1 because otherwise it will explode. And you can also see, of course, mathematically you know that this series will converge only if lambda by mu is less than 1. If lambda by mu is not, is greater than 1 or even equal to 1, then this will not converge. So, the sum will explode. And so what does it mean?

See here, when you say that lambda is less than mu, that means, lambda is less than, sorry; lambda by mu less than 1 that it implies that lambda is less than mu, right. And so this is the service rate and this is the arrival rate. So, obviously, you expect that otherwise people will go on collecting in the system if you are service is lower than the rate at which people are coming.

Or, in other words, the better way to look at it is, that 1 by mu is less than 1 by lambda. So, mean service time is 1 by mu, remember, because it is exponential mu. So, therefore, the mean time is 1 by mu. So, mean service time is 1 by mu. Now, this should be less than; and 1 by lambda is the mean time between arrivals, remember, because if the arrival process is Poisson with rate lambda then the inter arrival times are exponential with rate with parameter lambda.

And therefore, the mean time between 2 arrivals will be 1 by lambda. So, in general you expect that 1 by mu should be, that means the service, mean service time should be less than the mean time between 2 arrivals, right. So, therefore, then only you expect this system not to explode; that means, the queue will not explode and you will be able to process the customers faster than they come; I mean in lose way you saying that it will not happen, right.

So, therefore, this makes senses that, and so once you get your P naught as 1 minus lambda by mu, from here you get that your P n as lambda by mu raise to n into 1 minus lambda by mu for all n. So, therefore, nice way we have been able to compute the probabilities for the different states of the system, right, under these assumptions.

And, many more ways of explaining this, but basically the whole idea is that they should be; even otherwise from here you see, you can just see that P naught being finite must be because it is a probability of empty system then if lambda is greater than mu and this will go on becoming larger and larger.

So, here they will be a positive probability for, you know, n being, well, this is yes, yeah, because P naught will take with P n cannot be, but what I am saying is there will be a positive probability of the system becoming, you know, number of people increasing in the system because, lambda by mu raise to n, if lambda is greater than mu, then that will be become, this start becoming a big number, fine.

Now, if you want to find out the average number of customers in the system, so therefore, you want to know that at any point of time, what is the average number of people. And mostly when you design a facility, you base it on the average number of customers in the system because you should at least be able to cater to the average number of people in the system; and then of course, there can be variations.

So, that means, L here is, we will define. So, L is the average number of people in the system; and so this will be sigma n P n vary from 0 to infinity because the probability of

there being n people in the system is P n; so n into P n. The expectation of this P n, I mean of the variable denoting the, or we can say that may be L n is the, this thing random variable whose probability is n. So, or we have been, sorry; you can, we have been denoting it by, but that was N (t); so does not matter. Let us just keep it that, this way.

So, sigma n varying from 0 to infinity n P n, will give you the average number of people in the system. So, here substitute for P n. And since this is not, this is independent of n, I will just concentrate on this. So, let me call this series, sigma n lambda by mu raise to n, 0 to infinity; let me call this s. So, I just write it out, you know, like this.

Then I multiply this by lambda by mu s; and I just, because, see, the infinite term in the series I can start writing this from here, does not matter; this sum I can just, you know, slip 1 position, and so I start writing it from here. And again, both the thing are x turning to infinity. Now, when you subtract this from here, it will be 1 minus lambda by mu s. And here you see, this is 0; so this is lambda by mu; then this is 2 lambda by mu square; and this is lambda mu by, lambda by mu whole square.

So, therefore, the difference is again this. And this is, you know, anyway, from those of you who are familiar, known that this is an arithmetic co geometric series, yeah. So, the terms are, the first term is changing as an arithmetic progression, and the second term is changing as a geometric progression or coming from a geometric progressions; so arithmetic co geometric series, fine.

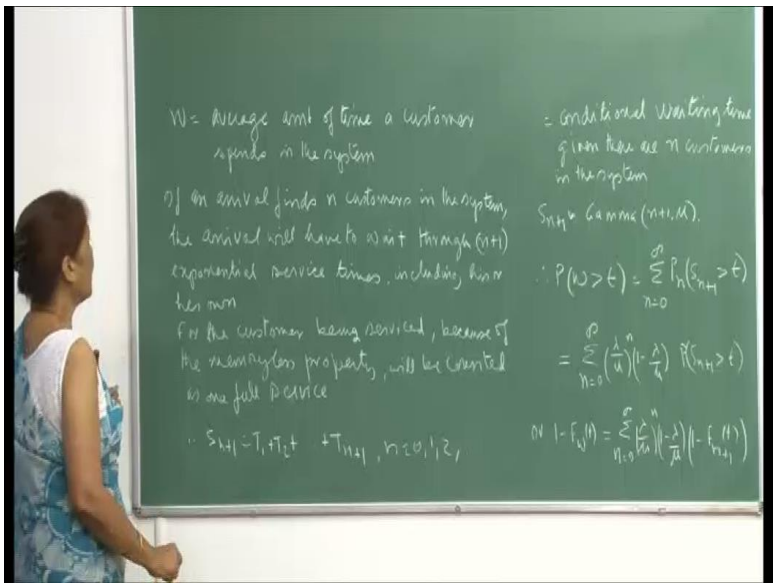
So, the way to sum up such a series is that you write down s and then you write down lambda by mu s, just slip to writing the terms from, you know, second position for this you write start writing from second position, and then you get, the difference comes out to be geometric series. And therefore, here the first term is lambda by mu, so I will write lambda by mu into 1 upon 1 minus lambda by mu, right. So, if your s is lambda by mu into, because this is this, so 1 minus lambda by mu whole square, right.

And so your L, because L had a, 1 minus lambda by mu here. So, then that will get canceled. Therefore, the average number of people in the system is lambda by mu upon 1 minus lambda by mu. And so here you see that even if this is large, atleast if this is close to 1; lager distance of course, because we cannot come to this expression if lambda is greater than mu; then this does not, I mean we cannot even talk about the average number of people in the system because a system would be explored it.

So, lambda by mu less than 1; if it is close to 1 then you say this number is small; and so 1 upon this will be very large; and therefore, again the number of people in the system we will be very large. So, this definitely gives you the idea is to how, you know, the lambda by mu has to be small for efficient service. And if you try, if you not able to keep lambda by mu much much smaller than 1, it will certainly will be, there will be times there will be chaos because this is only talking about the average number of people in the system, right.

So, therefore, this gives you an idea that if lambda by mu is reasonably small then this number will also be reasonably small; and so most of the time, I mean on the average you will expect, that there will be not too many people waiting to be serviced.

(Refer Slide Time: 37:09)



So, now the other characteristic of a queuing, of a good queuing model is that the amount of time a customer spends in the system should not be very high. So, therefore, we want to now estimate the average amount of time a customer spends in the system. So, again that will depend that will be a function of lambda and mu; your arrival rate and the service rates, right. So, let us find out this.

Now, if an arrival finds n customers in the system, the arrival will have to wait through (n plus 1) exponential service times because n people are already in the system, and he or she is the (n plus 1) th arrival in the system. So, there is, then before the (n plus 1) th arrival leaves a system, that means, (n plus 1) services have been completed, right.

Now, the thing is that there is already 1 customer being serviced because there are (n minus 1) people in the queue, and there is 1 person who is being serviced. But, because of this memorialized property we cannot say that, you know, this service, how long he has been at the counter, and therefore, how long he will takes more; we cannot say anything about it; that is as much unpredictable quantity as when he started the service.

So, therefore, because of this memorialized property I have to count that also as 1 full service; and therefore, we have saying that they will be, (n plus 1) service is to be completed before this arrival who finds there are n customers in the system, finally leaves the system, right; so therefore, your, S n plus 1, will be T 1 plus, T 2 plus, T n plus 1, and varying from 0, 1, 2, and so on.

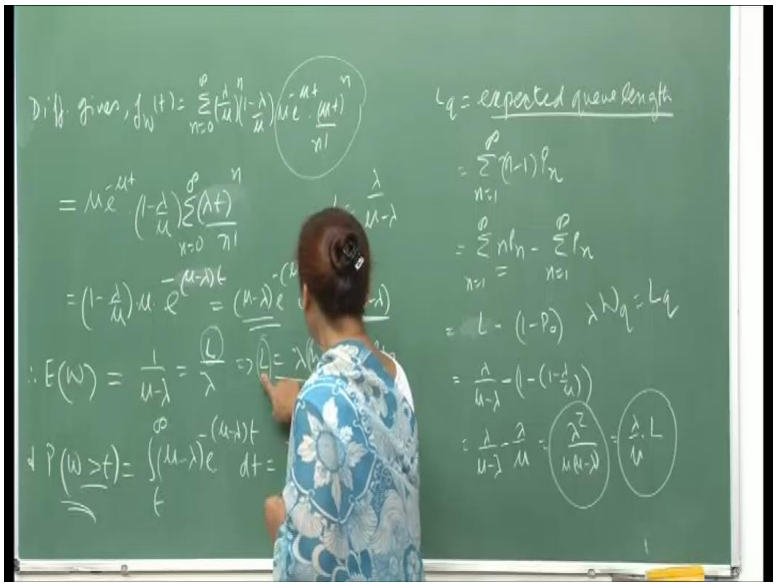
So, this is the, and this is the conditional waiting time given there are n customers in the system, right; because, S n plus 1, means your conditional waiting time given there are n customers in the system, and therefore, (n plus 1), services have to be completed. Now, we know, since the service times are exponentially distributed we know that some of these (n plus 1) exponential identically independently distributed exponential random variables will be gamma (n plus 1, mu).

So, the same parameter, but since they are n 1 of them. So, this becomes a gamma (n plus 1, mu); here s n plus is this. And so when you want to compute the probability that the average waiting time is greater than t, or that is the expected value, expected waiting time then this is sigma n varying from 0 to infinity, t n. So, conditional probability; remember, this is conditional. So, P n into S n plus 1 greater than t.

So, you will write this as, this is probability that S n plus 1 is greater than t. So, you are services the, n plus 1, services take more than t time to be completed. And the probability that the n people in the system then only, n plus 1, services have to be completed. So, this is sigma n varying from 0 to infinity.

So, substitute for P n, then you will get, lambda by mu raise to n, 1 minus lambda by mu into, probability S n plus 1 greater than t. So, this you can write as, 1 minus F w t because this is if I am saying that F w is the distribution function of w; and similarly, F n plus 1 t, I am denoting as the distribution function for S n plus 1. So, therefore, this is what I can write. Now, I can just differentiate both sides. So, this of course, is 0; I get the; so the minus sign, minus sign will cancel out, because so this is not a function of t.

(Refer Slide Time: 41:09)



So, here this will be minus and minus that will cancel out; and what you will get is that F w t is equal to this whole thing, and this is your gamma (n plus 1) mu p d f, right; mu e raise to minus mu t, then mu t raise to n upon n factorial. And now let us just simplify. So, what I will do is, this is independent of n; this is independent of n. So, the only quantity you see is mu in the denominator here, mu raise to n, and this is a mu raise to n in the numerator.

So, the two will cancel out, and therefore, simply we left to the lambda t raise to n, upon n factorial; the other things can be all taken out. So, lambda t raise to n upon n factorial, you sum up this from n 0 to infinity. And now, this is very familiar series for us; and so this will be, e raise to lambda t. So, therefore, I can combine it with this. So, therefore, e raise to minus mu minus lambda t; remember, mu is greater than lambda.

And so if you simplify this expression, mu minus lambda upon mu, cancels with mu. So, it is, mu minus lambda e raise to minus mu minus lambda t; and this is exponential, mu minus lambda, right. And therefore, you immediately know that the expected value of w is 1 upon mu minus lambda, right. And this if you remember the expression for L was, I will not see what was the expression for L, that was lambda mu minus lambda, right.

So, therefore, the expected waiting time is L by lambda, or what it means is that our average. So, this was, the average number of people in the system will be, lambda times t, average waiting time that a customer spends. So, and this is known as the famous,

should be t here, little's formula. So, this is attributed to little who first, you know, get this differentiate between L and w.

So, this is again you can, we can say out in words. so that you do not; and then if you want to find out the probability that, w is greater than t, then this is we have the p d f for w; this will be t to infinity, mu minus lambda, e raise to minus mu minus lambda t d t, which we now is this therefore, right; e raise to minus mu minus lambda t.

So, what can you say here it has been if you want to say that, yeah; so this probability that your average waiting time would be greater than t, again you can talk about in terms of mu and lambda, right; because this is essentially equal to 1 upon e raise to mu minus lambda t. So, this, if you want to this probability to be small, then obviously, your mu should be greater than lambda, quite, you know, substantially, so that this probability then is small because e raise to mu minus lambda t would be large, and so 1 upon that would be small.

And see, all these relationships and this quantities will help you to in modeling very efficient queuing system, and depending on what parameters you consider important you can accordingly concentrate on those, and then accordingly you know design your system so that you are mu and lambda conform to that.

So, that in the sense that if you want your L to be small, that means, you do not want its place to be crowded all the time then you concentrate on this. And if you, it is important that people should not have to wait for a long time then you will concentrate on this, right; but the 2 are related. So, L is equal to w, and therefore, you can say if you concentrate on this, you concentrate on this, depending in respect to lambda.

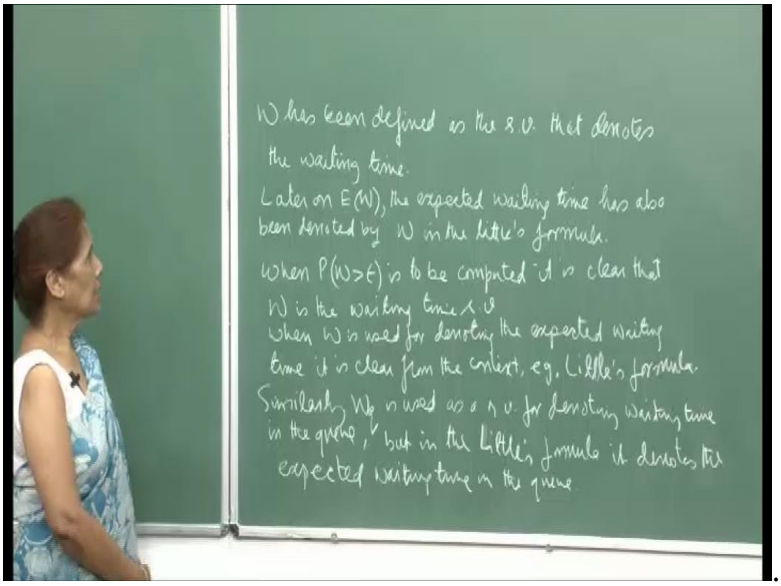
Now, the other quantity would be expected queue length. See, L was the expected number of people in the system which includes the person being serviced, but now here you are talking about expected queue length; and so that will be n minus 1 into P n because if there are n people in the system, 1 person is being serviced, so then the number of people in the queue are n minus 1.

And, this summation will be from 1 to infinity because if you have n people then 1 person is said to be being serviced and therefore, n minus people, n minus 1 people are waiting in the queue. So, that will be; so you want to compute the expected value of L q of the people in the queue, right, which is L q. So, then this is n minus 1 into P n. Now, I can separated out as n P n minus, sigma n varying from 1 to infinity, P n, right.

So, this we know is L, because anyway when n is 0, the contribution is 0. So, this is also the same as L. So, that I write as L. And sigma n varying from 1 to infinity to P n is actually 1 minus P 0 because when you add P 0 then the whole thing adds up to 1; so 1 minus P 0. So, this is it. So, l upon lambda upon mu minus lambda is your value of L; then 1 minus, 1 minus lambda by mu, this is P naught.

So, therefore, this becomes your; so that means, this is essentially lambda by mu into L, right; because lambda upon mu minus lambda is your L. And this is lambda upon mu into L. So, interesting this thing; and what you can see, in fact, the little's formulae also say that w q should be, at lambda times w q should be L q. And we will show this also because here lambda times w is l, so lambda times w q should be L q; one can derive this results also.

(Refer Slide Time: 47:23)



See, w, I have used as a random variable, notation w for random variable that denotes the waiting time; and then I computed a expected value of w, but then again in the little's formulae either it should be capital L, in the little's formulae I again use the word w only. So, what I am trying to say is that because in the little's formulae they used L capital L, capital W. So, did not want to change it.

But then what I feel is that is not really much confusion using w, you know; using the same notation for the random variable as well as for the expected value because you see when you are computing this probabilities like this then it is clear the w is being used as

a random variable because you do not associate probability, expected value of w is not a random variable. So, you will not associate probabilities with it, right.

So, therefore, probability, w greater than t, is to be computed that it is clear that w is the waiting time random variable. And when w is used for denoting the expected waiting time, it is clear of the little's formula then it is clear that this w denote the expected value, yes; may be one could have used to different notations, but that is, ok.

I just want to make sure that, I will make it clear that it should be possible to see from the reference to the context in what way w is being used. And the same holds for wq, because wq, I am using as a notation for denoting the random variable for the waiting time in the queue; we know just before your turn comes to be serviced.

So, before that the time you spend in the system, so this is the rate random variable denoted by w q. And again in the little's formulae we will use the, for the expected value of w q, I am again using the notation w q only. So, the same reasoning that it should not cause any confusion. And one should be able to see from the reference to the context in what way w and w q are being used. So, please keep this in mind.