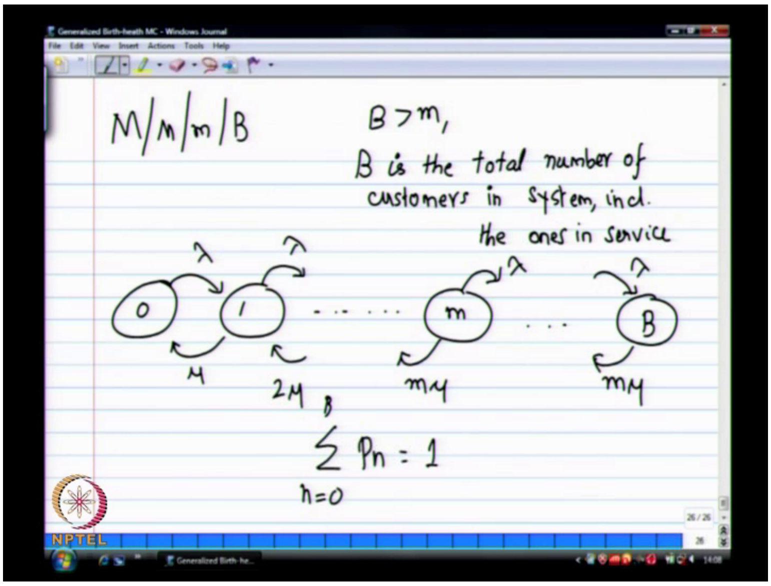
Performance evaluation of computer system

Prof. Krishna Moorthy Sivalingam
Department of computer science and engineering
Indian Institute of Technology, Madras

Lecture # 15
Queuing theory-IV

Welcome back. So, we finished of mm 1, m m m and m m m m. So, what is the next to look at is q th finite buffers. In general, any m m m q can have a finite number of buffers but, m m m B, we can write the equations for that but, is not really closed form of solution. So, we do not have that we just can (( ))

(Refer Slide Time: 00:43)



M/M/m/B

B > m,
B is the total number of customers in System, incl. the ones in Service

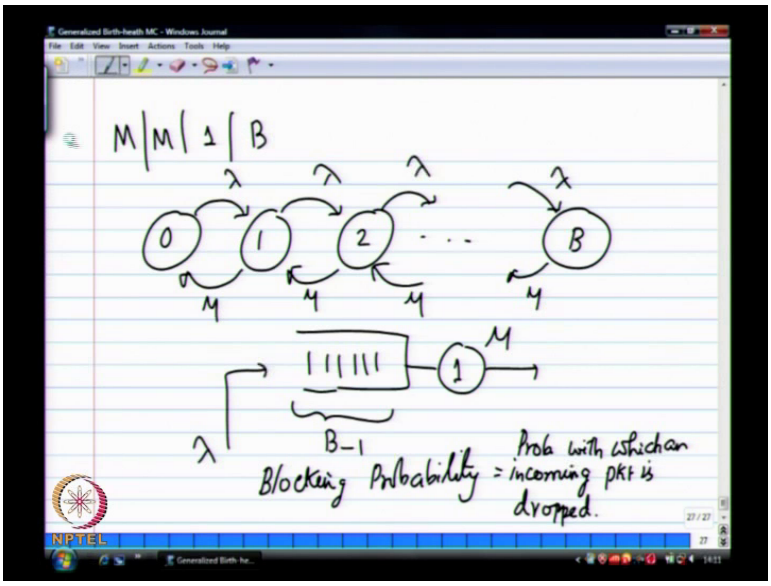
$$\sum_{n=0}^{B} P_n = 1$$

Generalized Birth-death MC - Windows Journal
File Edit View Insert Actions Tools Help
0
1
m
B
λ
λ
λ
λ
μ
2μ
mμ
mμ
NPTEL

So, to look at the M m m B system, where B is greater than m. Otherwise, if B less than m, does not make any sense if you have more servers and not enough people. So, we will assume that B is greater than m and B represents total number of customers in the system. That is the upper limit. So, that includes the ones in service. So, that implies there is B minus 1 buffers to hold pockets, while the one is being serviced.

The ones, if it is m m m, so we will have more than one customer who can be serviced. So, again we go back to our standard set of macro, set of states, lamda, lamda and the reverses of course, mu 2, mu and all that. So, we can write the p n before, simply the product. So, the only difference is that, now the set of state is going to be (( )). This summation p n, that will only vary. So p n, n equal to 0, p n equal to 1. That is the only difference. Otherwise, the equations are going to be the same as we saw for the M m m k. So, we will only look at the m m 1 B for specific results, for the analysis. So, for the m m; no there is n servers in the system. So, past time, the return rate is always mu.

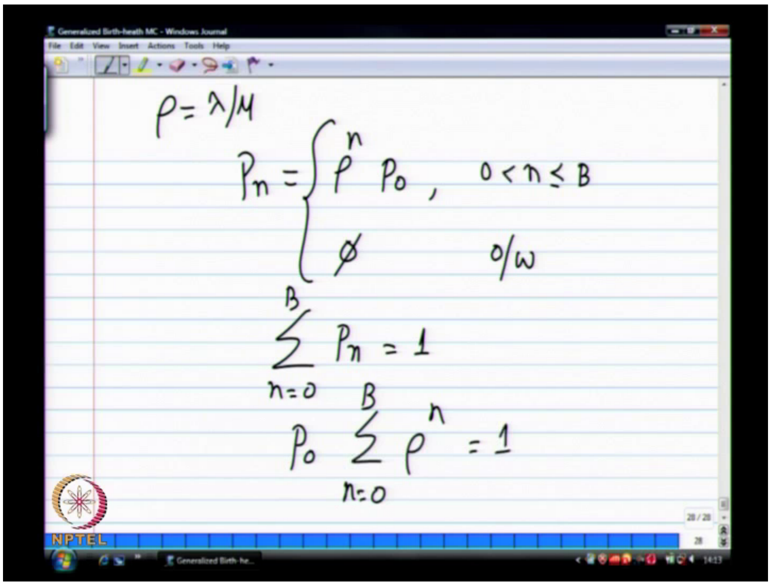
(Refer Slide Time: 04:01)


Generalized Birth-heath MC - Windows Journal
File Edit View Insert Actions Tools Help
M|M|1|B
λ
0
1
2
B
μ
1
B-1
NPTEL
27 / 27

Blocking Probability = Prob with which an incoming pkt is dropped.

This one is as usual. So, the representation for the system is that there are total of B minus 1 slots here. Then, there is one more guy here, who is getting served here and then, the system. So, in mm 1 B system, the pockets will keep arriving but, there is non 0 probability that packages arrive will get dropped. Therefore, the effective arrival will be less than lamda. But, as far as pockets coming into the system, changing states rate is still lamda. We look at the effective arrival rate in just a second. So, in addition to all things, we have this extra parameter compute called the blocking probability. This is the probability with which an incoming pocket is dropped. This of course, is servicing at new pocket, will be semis p B. Just like (()). So, we will compute the rest of the derivation. It will be exactly the same. There is no change.

(Refer Slide Time: 05:55)



$$\rho = \lambda/\mu$$

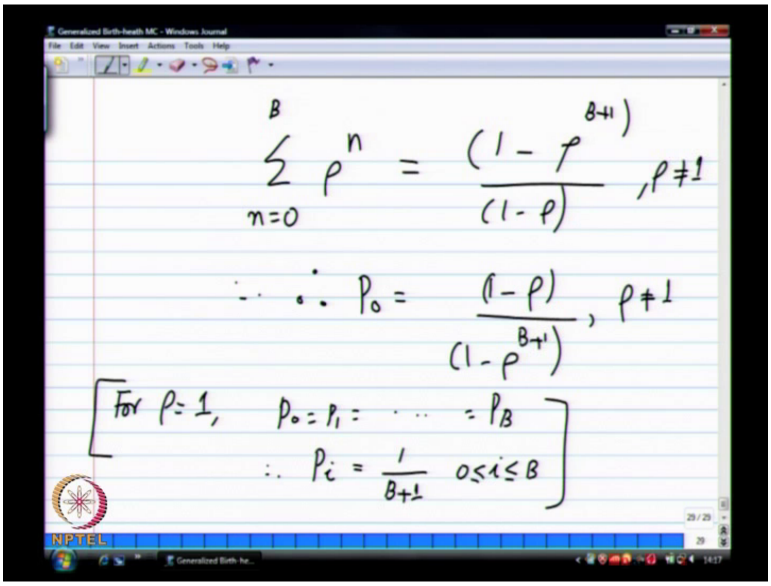
$$P_n = \begin{cases} \rho^n P_0, & 0 < n \leq B \\ \phi & o/w \end{cases}$$

$$\sum_{n=0}^{B} P_n = 1$$

$$P_0 \sum_{n=0}^{B} \rho^n = 1$$

So, this will be the same set of equations. Only difference is that, the normalizing equation will be; now, what is p n. We do not have to derive drive that. What we have done is enough. Simply rho n, where rho equals lamda by mu. So, this is simply that and only difference is that for n less than or equal to B or equal 0. It is defined its 0 anywhere else. No, n can be 0 (( )) . When n goes to 0, it will be really picky like that and then, your, so this is (( )) . Does not matter in this. The summation is n is equal to (( )) and this is our high school geometric series. So, we know the formula for this one. So, p naught is easily derivable.

(Refer Slide Time: 07:37)



$$\sum_{n=0}^{B} \rho^n = \frac{(1-\rho^{B+1})}{(1-\rho)}, \rho \neq 1$$

$$\therefore P_0 = \frac{(1-\rho)}{(1-\rho^{B+1})}, \rho \neq 1$$

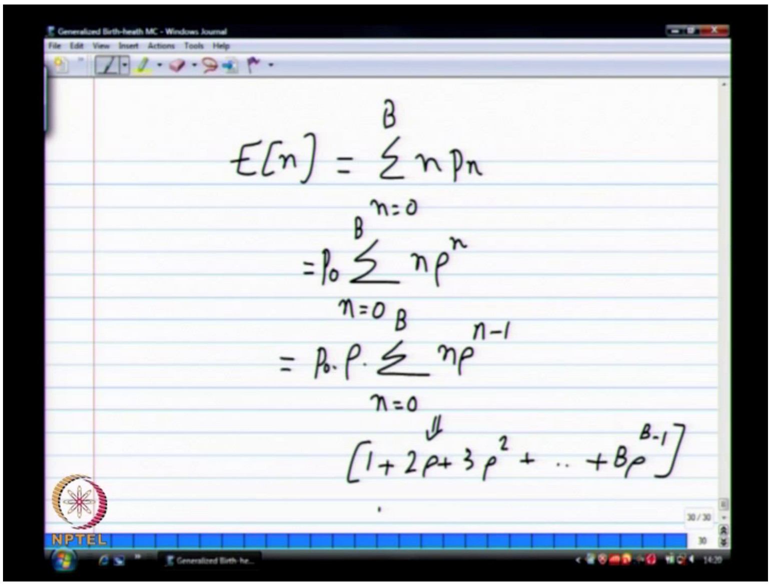
$$\left[\text{For } \rho = 1, \quad P_0 = P_1 = \cdots = P_B \quad \therefore P_i = \frac{1}{B+1} \quad 0 \leq i \leq B \right]$$

So, for this, you know, rho be less than 1 or equal it 1 or greater than 1? Does not matter because, it is going to convert whatever be the value of n. Therefore, the restriction on rho is gone. What is remain means that your system capacity is fixed. Arrivals can be as much as high as they want to be. Arrival rate can be as large as they want but, the pockets will simply get dropped. Only thing, p v will increase as lamda increases. Therefore, this is the stable queue. So, this queue is technically stable. So, what is that? So, 1 plus rho plus all the (( )) B. So, there are B plus 1 term. What is this? Either way it is fine. We leave this because, so whether you put rho minus are this one, if rho greater than 1, then rho power B plus 1. Actually, if rho goes to 1, this is not correct. So, this is only for rho not equal 1. If rho goes to 1 in this technique, and if it does that, we can simply directly solve, that is 1 over B plus 1. So, just look at rho not equals to 1 case. It will be simply be B plus 1, So, p n will be (( )) . If lamda equals to B, then all the P i's are equal that is only there are B plus 1 terms. Therefore, if dow not not equals 1, this is defined. Otherwise, it becomes even simpler. Therefore, p naught is, therefore, it becomes, total mistake, 1 minus rho by, so as I said for rho equals 1, all the p naught equals p 1.

If you can if look at Marco balance equation, lamda p naught equals mu p 1. So, lamda mu are the same. Then, rho goes to 1. Therefore, simply cancel out for all the p i equal. Therefore, that is (( )). Now, if you look at this again, if you plug in B equals infinity, so you can simply see that, when rho is less than 1 and B approach infinity, then the bottom factor, denominator becomes basically 1. Therefore, it is 1 minus rho. That is what we also derived in the case of a first system. So, in general, in the system, will the server be more ideal or less ideal, for rho less than 1. m m 1 B, will it be more ideal or less ideal. More ideal, because a customer is dropped. Finally, when the queue is busy, the server is free; the customer who could have been serviced has already left the system without joining the system.

So that, while looking at rho and since it is less than 1, denominator will be larger in that case. Those are just secondary analysis. But, this is here (( )) this is derive arrival Because, that is the rate of transition of Marco chain, to go from 1 state to other state. You would be with the rate rho only and you go B, then the rho does not really matter. No pockets will be there. Now, what is the next thing to do? We need to compute the number of expected number of users in the system and all that stuff.

(Refer Slide Time: 11:52)


Generalized Birth-Death MC - Windows Journal
File Edit View Insert Actions Tools Help
NPTEL

$$E[n] = \sum_{n=0}^{B} n p_n$$

$$= p_0 \sum_{n=0}^{B} n\rho^n$$

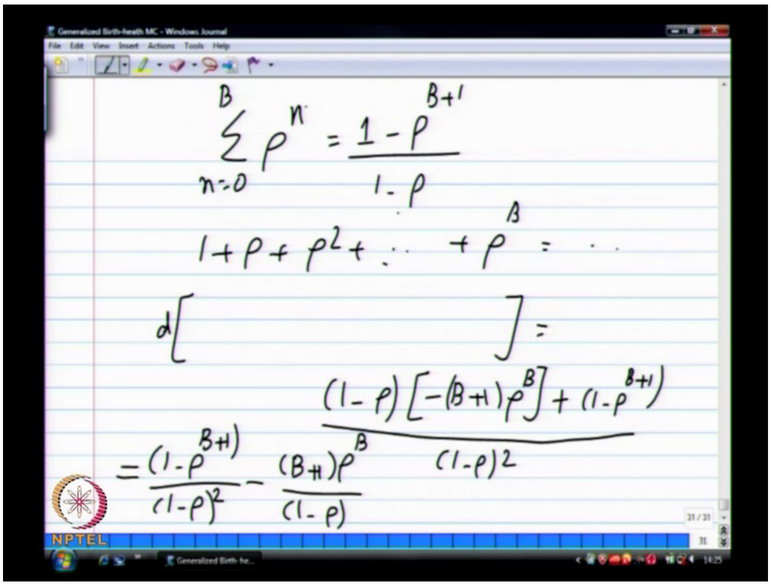
$$= p_0 \cdot \rho \cdot \sum_{n=0}^{B} n\rho^{n-1}$$

$$\Downarrow$$

$$[1+2\rho+3\rho^2+\ldots+B\rho^{B-1}]$$

So that, so it is very similar to the m m 1 except for this. Now, what next in this n rho n? So, we always find that in n rho n, n rho minus 1 is better and this is nothing but, what we trying to do is essentially 1 plus 2 rho plus 3 rho square, all the way to B rho B minus 1. Get the summation that we are looking to get the close form. And what is that? What is that value going to be? 1 minus (( )) 1 minus 0 B plus 1 divided by 1 minus 0 whole square. Next page. So, what is your derivation?

(Refer Slide Time: 16:13)


Generalized Birth-Death MC - Windows Journal
File Edit View Insert Actions Tools Help
NPTEL

$$\sum_{n=0}^{B} \rho^n = \frac{1-\rho^{B+1}}{1-\rho}$$

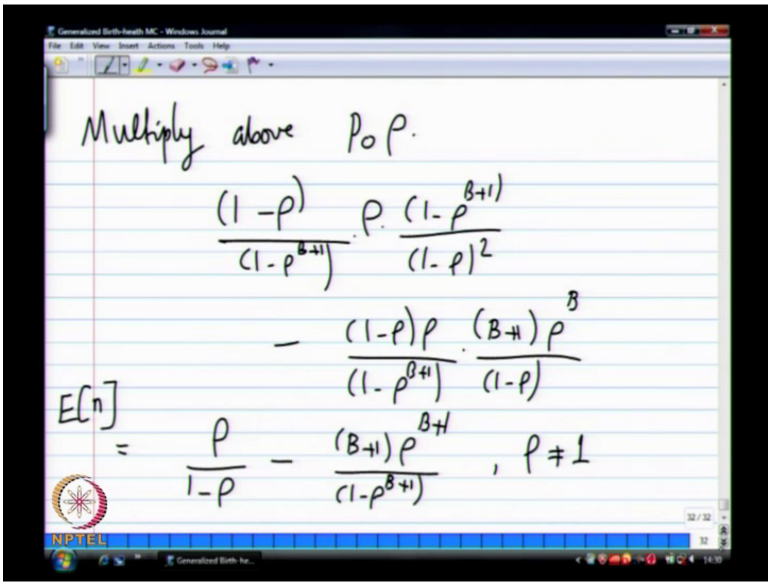
$$1+\rho+\rho^2+\ldots+\rho^B = \ldots$$

$$d[\qquad] = \frac{(1-\rho)[-(B+1)\rho^B]+(1-\rho^{B+1})}{(1-\rho)^2}$$

$$= \frac{(1-\rho^{B+1})}{(1-\rho)^2} - \frac{(B+1)\rho^B}{(1-\rho)}$$

Let me just, this you know, from this we should be able to derive at we want to do. So, what is this? I am doing it with expansion, or else I tend to make mistake. Otherwise, same thing. So, differentiate this thing. Actually, what, so differentiate the node and you get the right hand side differential of this is, now tell, so what is that right side? 1 minus rho into minus. So, it is solved. So, what is the simplified to? So, that is 1 minus rho B plus 1 divided by 1 minus rho and there is this. We will not actually, we just leave it as it is. This is squared here. Did I get it right? So, this is the summation, partial summation that you are looking for. Then, multiply this by p naught and rho, then we get E of n.

(Refer Slide Time: 19:27)



Generalized Birth-Death MC - Windows Journal
File Edit View Insert Actions Tools Help
NPTEL

Multiply above $P_0 \rho$.

$$\frac{(1-\rho)}{(1-\rho^{B+1})} \cdot \rho \cdot \frac{(1-\rho^{B+1})}{(1-\rho)^2}$$

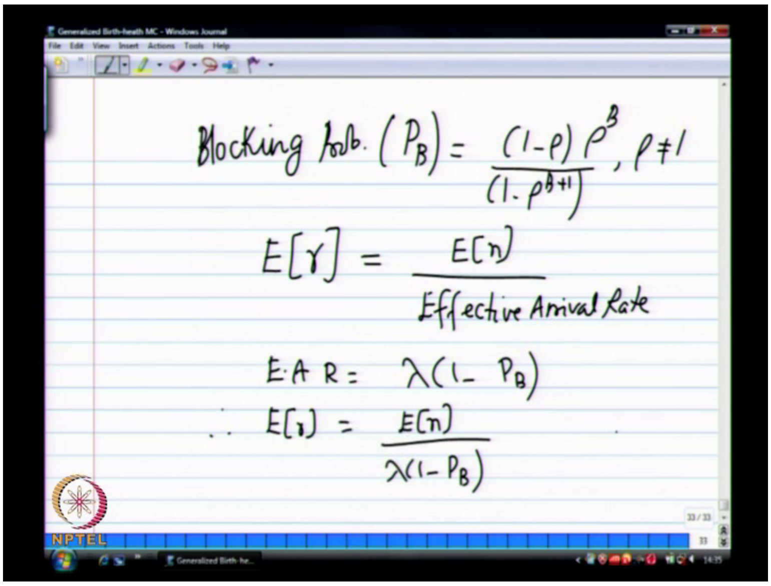
$$- \frac{(1-\rho)\rho}{(1-\rho^{B+1})} \cdot \frac{(B+1)\rho^{B}}{(1-\rho)}$$

$$E[n] = \frac{\rho}{1-\rho} - \frac{(B+1)\rho^{B+1}}{(1-\rho^{B+1})}, \quad \rho \neq 1$$

What happened? Told me something? B is, this is, I am just retaining this as it is. I am not splitting. I am just separating the sums. That is it. B plus 1 rho power B and then we finally, is it correct? So, I have some pre defined derivations. I was doing some in the train and then I finally end up with one. So, it is not easy deriving in the train when it is moving. So what we get? So, this is now, I will write that whatever we had before. So, it is, so what is our p naught. p naught is by 1 minus rho, that is another rho and then, there is, no, this was p naught 1 minus rho by 1 minus rho B plus 1. Then, there is your rho and then, we had these two terms. This was the left part of the term. So, we just repeat this into this B plus 1. This is finally what we end up with. Yes or no? I am the only one getting this in the train.

So finally, we should look at some thing that is starting to get familiar. So, this is basically, rho by 1 minus rho. What is that? It is E of n for infinate queue. Will it be less or more than infinate queues? The number of customer should be less. Therefore, if you have a minus sign here and then, what is your multiplicative factor? It is B plus 1 rho B plus 1 divided by, therefore, as B becomes smaller and then, the number of customer also will become small. So, this is finally the E of n that I was looking for. So, once you get E of n, I can then simply proceed with deriving. I can also derive E of q in the same way like we did n minus 1 into rho of n. Your last term. You go back and check. We got those two terms because, simply I just differentiated this. I just kept the term as it is. So, I took this term out, put that in front and this 1 minus rho gets cancelled out. Where would I have lost the two term? Only two terms here. So, that is the first expected number of customers in the system.

(Refer Slide Time: 24:30)


Generalized Birth-Death MC - Windows Journal
File Edit View Insert Actions Tools Help
NPTEL

Blocking Prob. $(P_B) = \frac{(1-\rho)\rho^B}{(1-\rho^{B+1})}, \rho \neq 1$

$$E[T] = \frac{E[n]}{\text{Effective Arrival Rate}}$$

$$E.A.R = \lambda(1 - P_B)$$

$$\therefore E[T] = \frac{E[n]}{\lambda(1-P_B)}$$

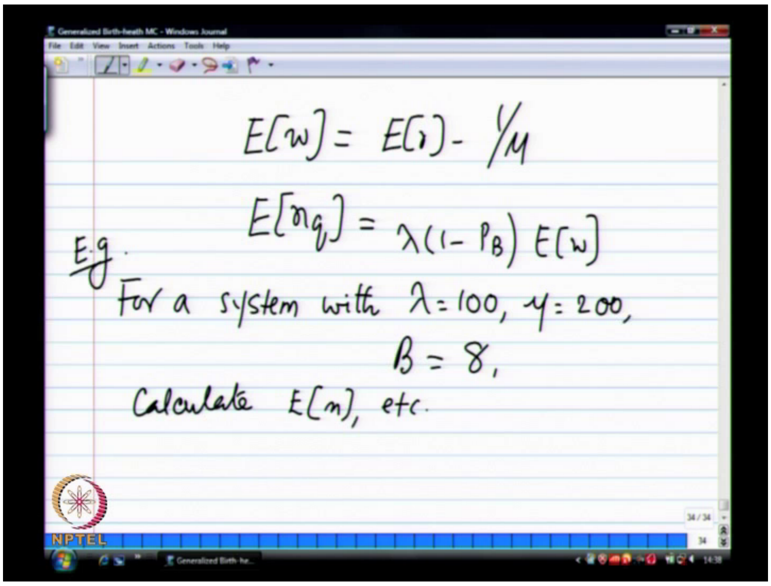
So, what is now blocking probability. What is the probability that the customers will arrive to a full queue, that is a blocking probability or have a full system. So, what is that now? Basically, probability being in state B, p B. That is why, p B here. So, simply 1 minus rho, p n equals p naught into rho n. Then, p naught is the first part. This is p naught and that is n equal p. So, this is our blocking probability. So that is how we derieve the probability of customers getting blocked in the system. So, why this is useful is, we usually tried to do it in the other way. We tried to see what should be the buffer size to meet some target blocking probability. You want to set the p v t to some value and then, go backwards and say for this kind of arrivals, this kind of service I should have atleast these many buffers in the system. This is the total customers in the system. That simplification, otherwise B plus 1. If you include, do not include B plus 1 B plus 2 all the way to B plus m. But, just to avoid confusion, we said B includes all the customers and a service and therefore, B is larger than m. Now, how do we calculate the waiting time, response time?

So, the arrival rate, that is, if you look at system as such, customers are coming and they are getting (( )) out. So, only a sub section, sub set of the customers enter the system. So, what is that rate at which customers enter the system. What is the arrival rate for, effective arrival rate. So, this is, basically E of n but, not lamda but, effective arrival rate. Because, only customers who see an empty part can enter the system. So, what is the rate at which, what is the rate for that. So, effective arrival rate is, we have seen, some of you have seen this, that is that probability p B will get kicked out. 1 minus p B you enter the system.

So, E of r is basically, now simply that, whatever expression we saw before divided by lamda into 1 minus p B. But, g m 1 at d m 1; m d 1 we have results. d m 1 I, I am not, d m 1 it becomes a discrete time given, discrete time change. Every instant of time, you want to get a pocket or some every few instance of time, you want to get a pocket. That, you can model, next, you should to try to model that as, see, these are two things. One is your arrival instance are not there at every instant. So, we have to translate the d into some sort up geometric x series. That is, say that, if I get for sure one pocket every intance of time, then you will have some probability of being the same stated in terms of arrivals but, the return can be some marco. The return rate will be exponantial.

For m d 1, will have. m g 1 actually, derivation we will not able to do today but, that is slightly different derivation is there. So, infact there is other system called semi marco models, where you can use the same approach, if I have given the set of state and set of probability transition matrix, you can, exactly as this just derive this visiting probabiliies, the v's, and then from that, if know the average wait, holding time in each state, then you can compute the steady state probability being in each state. That, we will look at when you look at example for t d m a. How we can, what t d m a is using a, t d m a can modeled in two ways. One is m g 1 with vacation or the m d with vacation and the other ways is using same marco models. That, as we go through, we will look at it. Those things will all come up. Next, you will look at m g 1, will look at m d 1 results then, m d 1 with vacations. That is also there. Where, the server goes on the vacation, after in between. So, any questions about this derivation so far? So, once I determine E of r, then the rest, it can simply follow.

(Refer Slide Time: 29:40)



$$E[w] = E[r] - 1/\mu$$

$$E[n_q] = \lambda(1 - P_B)\, E[w]$$

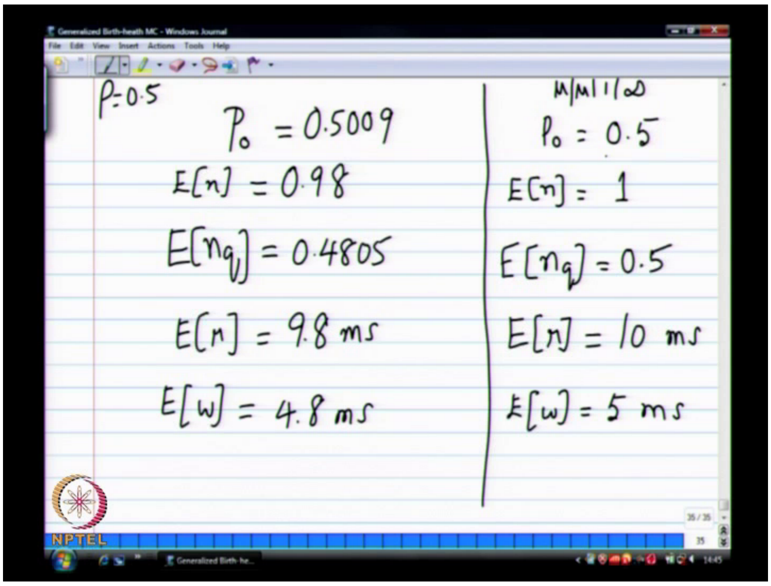
E.g.

For a system with $\lambda = 100$, $\mu = 200$, $B = 8$,

Calculate $E[n]$, etc.

So, what is E of w? The service time for customer does not change. That is still the same, 1 over. Simply, the total server response minus the service time, 1 over mu, this is the mean service time. And then, therefore, likewise, E of n q is, in terms of E w, (( )) bracket the, 1 minus p B. Questions on this basics derivation so far? So, one example to, so let us look at that lamda equals 100 and what was mu before 200.5 atlisation. So, for a system B equal 8, calculate all the four performance matrix.

(Refer Slide Time: 31:37)



$\rho = 0.5$

$P_0 = 0.5009$

$E[n] = 0.98$

$E[n_q] = 0.4805$

$E[r] = 9.8$ ms

$E[w] = 4.8$ ms

M/M/1/∞

$P_0 = 0.5$

$E[n] = 1$

$E[n_q] = 0.5$

$E[r] = 10$ ms

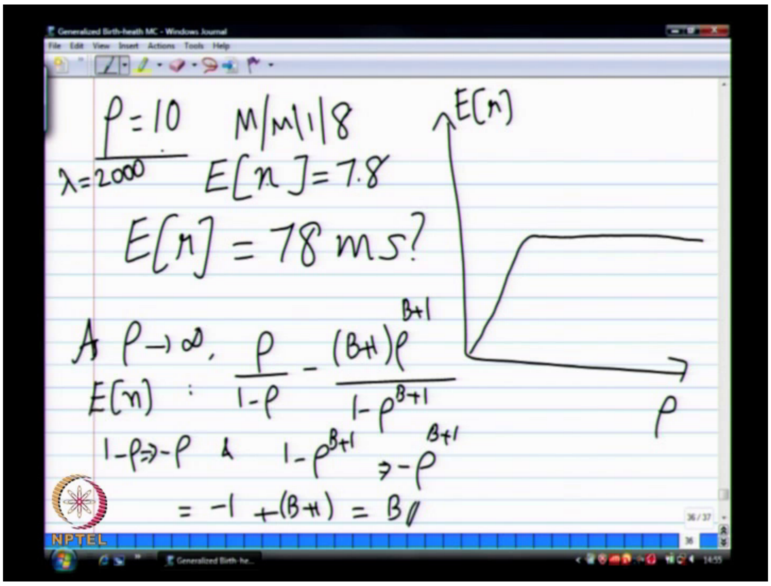
$E[w] = 5$ ms

So, will always try to find out p naught. Let us compute p naught first. We do not really need that because it simply sits in E of n. But, so what is the probability the system being idle, 0.5009 and then what is the corresponding m m 1 system values. p naught is, rho is 0.5. m m 1 system is, idle is simply 1 minus rho. rho is server utilization. So, this is, E of n was 1 rho squared. So, it is 0.5 by 1 minus rho 0.5 into 0.5 by 1 minus rho E of n point n q 0.5 f of n q is more than the E of n q. Check. This is 9.8 then, E of w is 4.8. So, if you have to use this number, still advantage. So, these are the numbers that is reported. So, what you find E of n is less, naturally and because E of n is less, the delay is also less, average delay is less in the system.

So, what is the best way to reduce in a system. Somebody says reduce delay in your system. Simplist is to impose limit on the buffers because, you have more buffers or less buffers, more pockets will get droped. So, fewer customer get service. Sometime you see papers where, they will say, my delay is reduced in the system and then, you go look at proof that how many customers are actually are been serviced. That will be also correspondingly less. So, less customer served, then you will end of with having less queueing in the system. So, the easiest way is to simply gap that without having to increase your service rate or decrease lamda. Simply imposing the restriction on B itself because, when a B reduces your 1 minus, p B also will increase correspondingly. Therefore, less pockets enter the system.

So, what will be a worst case delay. In this the m m 1 B system, what is the worst case delay. It is the, if the, worst case would be infinity. So, my B by mu will be the, on average you will see that if a customer enter your system, then you can look at, how do I quantify that, define the worst case. If you arrive at the system, I need to define my delay. So, on average it will be the largest delay that I will see, if I enter the system. We will have to work out the definition again. Let us try that. It will be last pocket therefore, you see 1 by mu B into B into B minus 1 into 1 by mu ahead of you, then you will see 1 by mu for yourself.

(Refer Slide Time: 38:11)


Generalized Birth-heath MC - Windows Journal
File Edit View Insert Actions Tools Help
E[r]
ρ
NPTEL
36 / 37

$\rho = 10$ $M/M/1/8$

$\lambda = 2000$ $E[n] = 7.8$

$E[r] = 78\ ms?$

$A\ \rho \to \infty,$ $E(n)\ :\ \frac{\rho}{1-\rho} - \frac{(B+1)\rho^{B+1}}{1-\rho^{B+1}}$

$1-\rho \Rightarrow -\rho$ & $1-\rho^{B+1} \Rightarrow -\rho^{B+1}$

$= -1 + (B+1) = B$

Now, let us try that. Let us try rho equals 10. If rho equals 10, what is E of n and what is particularly, delay. Because, what do you expect in terms of with increasing rho and with E of r ploted in the direction. This will naturally increase. As rho increases, this will increase but, at some point it should simply stablize and that stability value would be? Let us see which B (( )). Should be 80 milli seconds. That should be the largest delay that you have seen.

Only the thing is, really at worst case and everybody might have infinitily long times and therefore, you may end of having infinity, infinity into 8. Let us see with this, what is happing and what is coming. So, rho is equal to 10. What is E of r? Only thins is , the service time for a customer is never, is not finite because, your service is mu equal minus lamda is x going to be greater than 0. So, if you are really bad, thus if you look at the worst case is, that definition that all customers might require close to infinity service. In which case, you might be waiting for very long time. So, that is hot to quant. So, what we are saying is, under heavely loaded ysstem, what is the expected value would be.

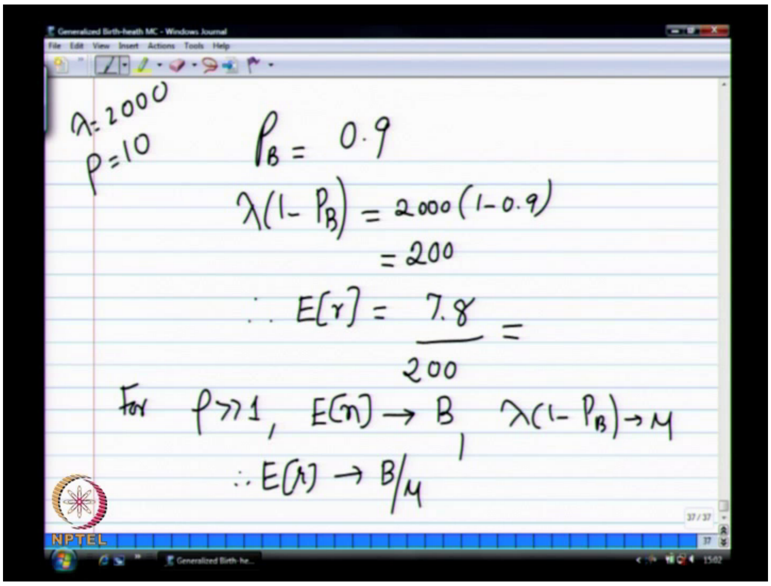
So my, that is the way it will work. No, system will not be unstable. That is the whole thing with finite queues. This is rho equals 10 because, what will happened is large fraction of pockets will never enter the system. Did you make sure that both up, the numerator and denominator are both negative? Then, use the formula. No, for infinity you have maximimum is 1. So, your system will never the v of n greater than 1. It will always be less than 1. It will approach 1, when rho become, well now, sorry, in that case and I am looking at m m 1 8 and not m m 1. So, will be actually more. It will be greater than, will increase 7.8.

So, you will get about 7, so you will 78 milli seconds as a E of r. It will be greater than 1. So, what is happening is, your queue is getting to be full. So, as rho approaches infinity, E of n approaches B. Then, you can look at a formula itself and then verify. Plug in rho equals infinity and see what happens. So, infinity to the power B plus 1 divided by 1 minus infinity. You simply chuck that 1 minus business and all will have is infi rho by rho by 1. So, you will have, will land uo with B. You plug in rho equals infinity in your E of n expression and then, rho by 1 minus rho is 1. Infinity by infinity is 1. You forget the 1 minus rho factor. So, remove that 1 minus in both expressions, you will simply have B plus 1 rho B plus divided by rho B plus 1.

So, B plus minus 1 equals B, except that minus will have to be a appropriaely defined. If rho is very large, then 1 minus rho will become minus rho B plus 1. So, if I simply look at these expression and plug in rho equals infinity, what happens, 1 minus rho is nothing but, minus rho. 1 is so small and therefore, 1 minus rho B plus 1 also becomes minus rho B plus 1. There, rho so large, so that 1 minus makes no diffrence to the data.

So, you will have simply minus 1 then, plus B plus 1. Everything else gets and that is what we should be having. If the load is very high, then you will, all the slots in your system will be filled up. Agreed. If rho is 1000, 1000 B plus 1, 1000 B plus 1. For very large rho, whatever be the particular value of rho, by E of r will arrive into the system will be, your p B will be fairly large. Goes to 1, yeah. It goes to almost 0. If E, for rho equals 10, what happen to 1 minus p B? Customers actually enter the system. This is only for customer entrying the system. This is the response time of the system for only the customers who enter the system. Not the ones who arrive at are not serviced. Customers who are not serviced are never going to you, are not part of our, that is why E of r is only, that using little slot, for only the customers you actually enter the system. This is, let us try that. So, let us verify. So, what is the p B. In that case, for the previous case, when rho is 10, the same example, talking about your lamda is 2000.

(Refer Slide Time: 46:46)


Generalized Birth-heath MC - Windows Journal
File Edit View Insert Actions Tools Help
NPTEL

$\lambda = 2000$

$\rho = 10$

$$P_B = 0.9$$

$$\lambda(1-P_B) = 2000(1-0.9)$$
$$= 200$$

$$\therefore E[r] = \frac{7.8}{200} =$$

For $\rho \gg 1$, $E[n] \rightarrow B$, $\lambda(1-P_B) \rightarrow \mu$

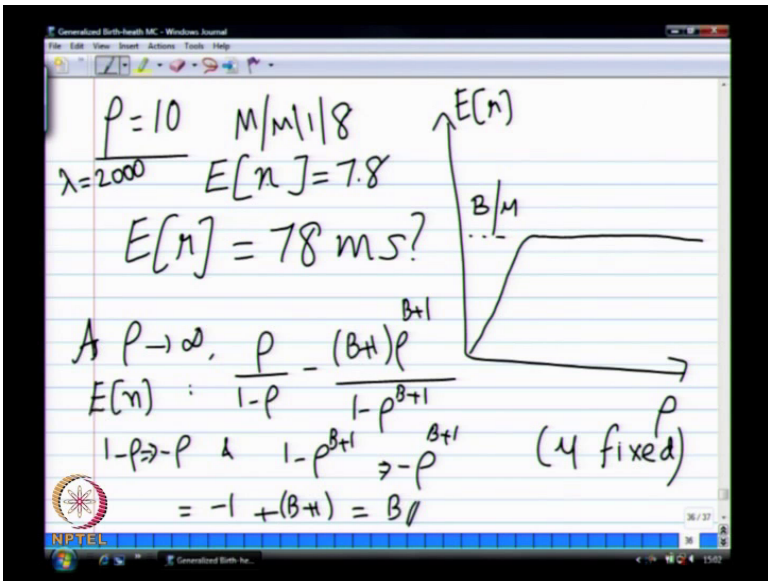
$$\therefore E[r] \rightarrow B/\mu$$

See, your lamda is fairly large, now tell me, what is the 1 minus p b. In your system, you lamba is limited by the mu. Lamda will basically as good as as E of 1 because, your system is constantly removing. Basically, whenever there is one customer who is removed and another customer enters into the system, arrival rate becomes more. So, anything greater than mu are actually, not exactly mu.

We try rho equals 1, you will not get this. Then, in very large system, where there is always instant replenishment, as one guy will kicks out, an another guy is immediately replaced in that customer because, your lamda is so large. So, let us, what is p B for this rho equals? 0.9, okay. So, effective arrival rate is approaching 200. So, that is now getting close to our mu basically. So, it will be, therefore, E of r but, the 7.8 is correct, divided by 200 and that and the long range is B over mu. That is what we are trying to say that and rho goes to infinity, your average delay for the system, when rho goes very large will become (( )) . Your lamda into effective rate of approach is mu and your E of n approaches B, set by B by mu. So, when rho, how about this. How it approaches infinity? rho larger larger than 1. So, basically, lamda into 1 minus p B equal mu. So, your p B becomes, that is it becomes. Effective arrive rate becomes mu. So, equals 1 minus rho. That is what we are looking for. Your p B approaches 1 minus rho. Whatever, now rho is still large. That is not correct. p B goes to 1 but, your lamda 1 minus p B approaches. So, mu, it will call mu because it becomes a balance system. Arrival equals will equal to (( )). In all other cases, we have seen arrivals are less than departure.

Every time a customer departs, another customer is added. The departure rate is 1 over mu. So, the addition is also mu. Only one server will affect here. As soon as the customer leaves the system, this is your effective arrival rate. Not your lamda. Effective arrival rate becomes mu. Questions on this? Now, my graph makes sense.

(Refer Slide Time: 51:19)


Generalized Birth-heath MC - Windows Journal
File Edit View Insert Actions Tools Help
$E[n]$
B/M
ρ
$(\mu \text{ fixed})$
NPTEL
36 / 37

$\rho = 10$ M/M\1\8

$\lambda = 2000$ $E[n] = 7.8$

$E[n] = 78\ ms?$

A $\rho \to \infty$,

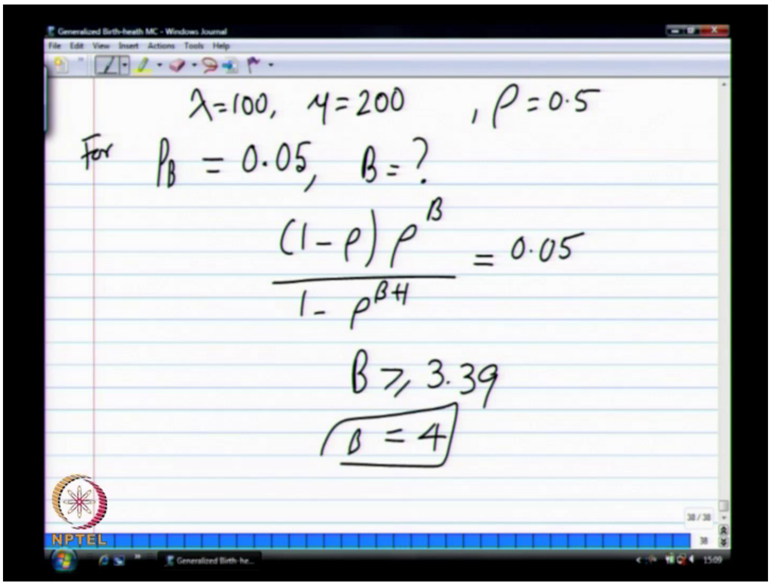
$E(n)$: $\dfrac{\rho}{1-\rho} - \dfrac{(B+1)\rho^{B+1}}{1-\rho^{B+1}}$

$1-\rho \Rightarrow -\rho$ & $1-\rho^{B+1} \Rightarrow -\rho^{B+1}$

$= -1 + (B+1) = B$

So therefore, this becomes, then your lamda keeps decreasing or your rho keeps decreasing. So, then in that case, yeah so here we are. You are saying that in my graph, so let me put that clause there. So, if, or if you fix rho here or lamda rather and then, keep decreasing your mu, you will see the same thing. If your service rate keeps on decreasing for the same arrival rate, lamda wait, just going the other way, your lower rate will keep on increasing. Ultimately, you will, for whatever value of mu that you looking for, each time your mu is going to be varing. In this, yeah but, it be could be that. yeah (( )) In average cases, let us, you know, that customers are coming at 1000 customer per second. That is known. You are trained to find the appropiate customer service capacity. What, how should be a, like buying a machine for example, one of the sample that it gave was, one copy can print at 10 pages per minute in the printer or other printer can print 20 pages per minute and you are trying to find out what is optimal. You want to reduce the waiting time to something that is acceptable. So, in that case, you want to keep that lamda fixed. But, there your mu, so you find out something else. So any other analysis that you want to do on m m 1 B? You have seen how, anything else on m m 1 B. No? So, usually we should do the other way.

(Refer Slide Time: 53:54)


Generalized Birth-Death MC - Windows Journal
File Edit View Insert Actions Tools Help
NPTEL

$\lambda=100, \ \mu=200 \ , \rho=0.5$

For $p_B = 0.05, \ B = ?$

$$\frac{(1-\rho)\rho^{B}}{1-\rho^{B+1}} = 0.05$$

$$B \geq 3.39$$

$$\boxed{B = 4}$$

Given some target propability, how do you determine the value of B. That is more interesting. For that same scenario, lamda equals to 100 and say mu equals 200. So, if I said my p B to be something, say I do not want to miss anymore than 1 in 100 pockets. What was the p B that, did we calculate? p B in the othercase? We did not. So, let us say we make it 0.5. What shoud B equal to? So, it is a difficult router design problem. In general, any queueing problem, capacity design problem. How much capacity should I have in the system? Given that the rates of arrival departure are fixed but, I have to meet some target propability. I do not want to loose any more than that. It should be less than or equal to but, any way if I say that, you know, I do not want any worst case. Is that a closed form solution or do you have any iteration for that. So, you go about by, so, rho equals 0.5. B should be greater than or equal to 3.3. So, you need atleast four buffers in the system if you want meet the stuff and this is state solution. You just replace rho B by whatever x in then and just plug it in, you are done. Need not elaborate on this.

So, that is where this can be handy. This is, of course, a very simple system that what we really have all routers and it is much more complicated and we have series of routers. Let us look at from the delay or buffering falication prespactive, is not that easy. But, we can have some sort of estimate for every router and say for each router, I am going to have this kind of designers.

Effective queue is still the same lamda. It is on a single queue not a multiple queue. It is the total number of customers in the system, so 4 minus number of servers. In this case, it is one single server. If it is more servers, then we have to actually solve a little more. The solution will involve, if it is m m m B, then that is more work has to be done.

So, the buffers can be on the length arts. It can also be, so this is, if you are doing simply output queueing or even input queueing, this is output queueing is what this is actually used for. If it is input queueing, then it gets a little bit more complex. Sometime, just pretend, for analysis sake, what if it is an output queue. Look at the total arrivals to that particular align card, output align card and then look at the service rate of the align cart and then do the dimensioning based on that. Dimensiong only for the router. We are also worried about dymantion for flow. It will get even more complex because, one link is shared by multiple flows.

Then, how do you, how much do you buffered your sign? That is what our buffer design people know more about that. They can do the buffer allocation. But, when an output card is shared amoung multiple flows, then how do you design for the each flow how much. So, I have a total of 1000 buffers available on the line card. But, the number of flows are varing, then just allocate everybody 20 flows or give them or 20 buffers. 20 buffers is the number of buffers fixed on performance basis and then, you will always have overflow at that time too. Suddenly, you see a surge in flows, you want to see a surge in the number of pockets entering the buffers but, the number of flows entering may be large. If it is equal division and you are doing a round robin allocation allocation, then it will be m d 1. This is server is vacation. That that is a time priority replexing system. We will look at that.