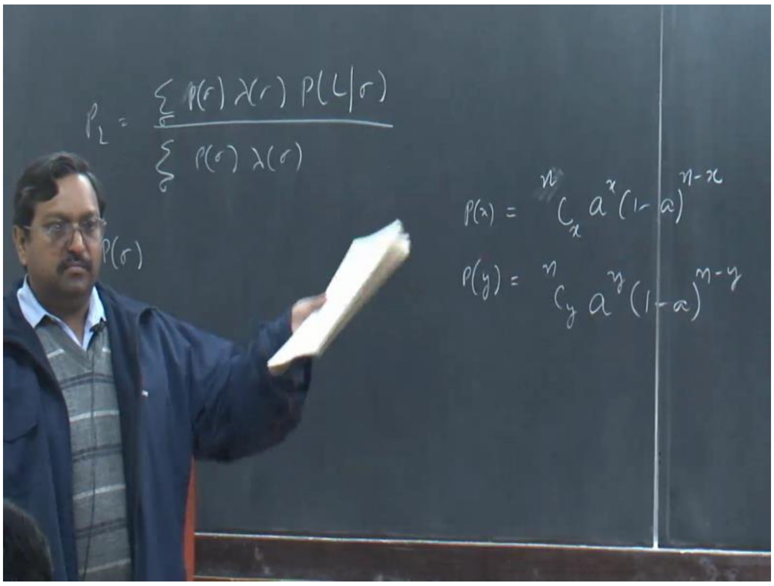
Digital Switching
Prof. Y. N. Singh
Department of Electrical Engineering
Indian Institute of Technology, Kanpur

Lecture – 7

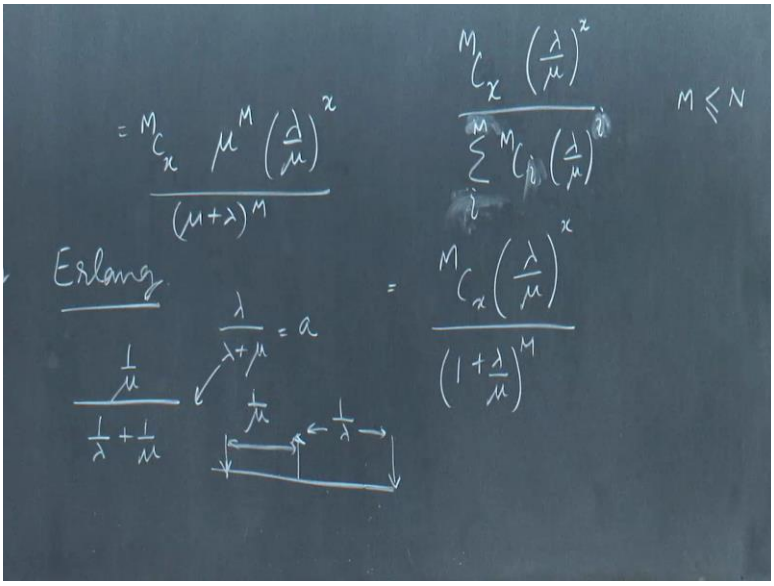
Start from the place, where we left last time. And we will solve to get a close form solution.

(Refer Slide Time: 00:30)



So, the P L – the call loss probability is… I think that is where I left. Probability of being in a state sigma – arrival rate – probability of call loss condition on sigma – summation over all sigmas actually. So, that is where we started. We have identified individual components. I think for P of sigma, that is what we were doing and we could get to a solution, which was… I think I have not proven that; it was n C x a raise to the power x into 1 minus a raise to power n minus x. That was probability of being in a state x and corresponding probability of being in a state y was n C y a raise to the power y into 1 minus a raise to power n minus y; that is what we had done. This I had got from ((Refer Time: 01:44)) distribution.

(Refer Slide Time: 01:49)



So, very quickly just recalling that, what we did was being in a state x was M C x into lambda by mu whole square 0 to M… We are taking case, where M is smaller than N actually. So, this can only go till N; you can only have M occupancy, not N. So, that was the probability of being in state x; and correspondingly, it is true for state y also.

Student: ((Refer Time: 02:44)) summation…

x; you make it i ((Refer Time: 02:53)) that is better. So, it will not confuse. And what I did; I think if you recall quickly, I think I did very fast, because it was in the end of the class. Now, this is a complete binomial. So, I can always write this thing as 1 plus lambda by mu raise to power M. This is coming from complete binomial and then I moved it… We just modify this thing. It is mu plus lambda – this power M; mu is this power M. And what I did is I said let us put lambda plus mu as a. The reason for doing this was – if I divide by actually… If I take mu out, this will be N lambda. So, this will be 1 over lambda plus 1 over mu. This value will be equal to this. So, on time scale, if this is a first call arrives, the call remains there; call actually goes away; call is cleared; the next arrives. So, this duration is approximately 1 over mu and this duration is approximately 1 over lambda. So, this technically gives nothing but the fractional utilization. And we also defined this will as a load in erlangs. Erlang is a name of a gentle man and then… So, since he had contribution to switching, to respect him, it is given as erlang.

So, now, how much erlang of loads? So, for one telephone line, the fractional utilization cannot be more than one. If you have 1000 input ports and if you have 0.9 fractional utilization per line; so it will be 900 erlangs load on the whole switch. That is the way it will be defined. So, we solve it further actually.

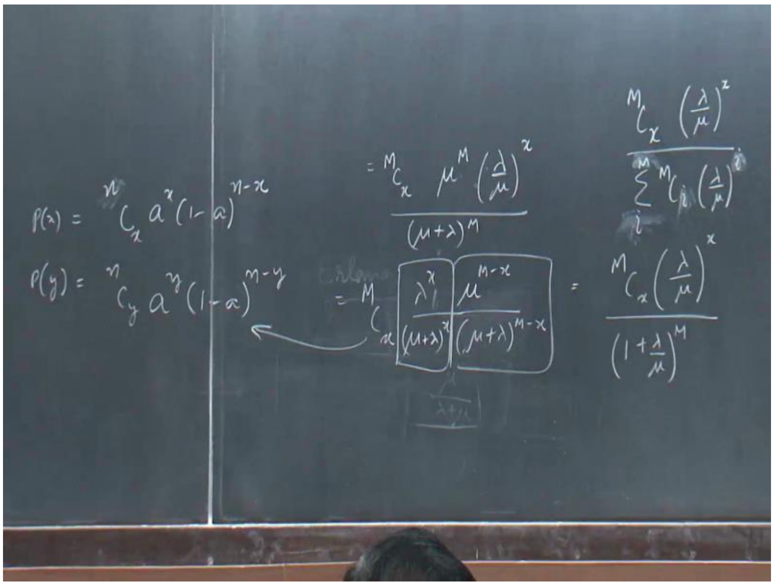
Student: Sir ((Refer Time: 05:18))

Because call cannot arrive. See when the line is busy, the call is on hold, no call can arrive during this period. And it is not between two, because it is a special kind of process; you take any particular time instant; it need not be that call should have been arrived that time. And after how much time the call will arrive? The average time will always be 1 by lambda; that is the beauty of this particular distribution, because call cannot arrive here; call will only arrive after this. See at this ((Refer Time: 06:06)) should not be included in the arrival rate, because that time line is busy, occupied; call is already there. So, 1 by lambda corresponds to this entity. If you converse, put to all 1 by mu is equal to 0, then it will be exactly whatever is the ((Refer Time: 06:21)) process, which we have learnt. So, that is why it is done this way, because telephony it is accurately in this fashion, because it is a circuit switch scenario.

(Refer Slide Time: 06:45)

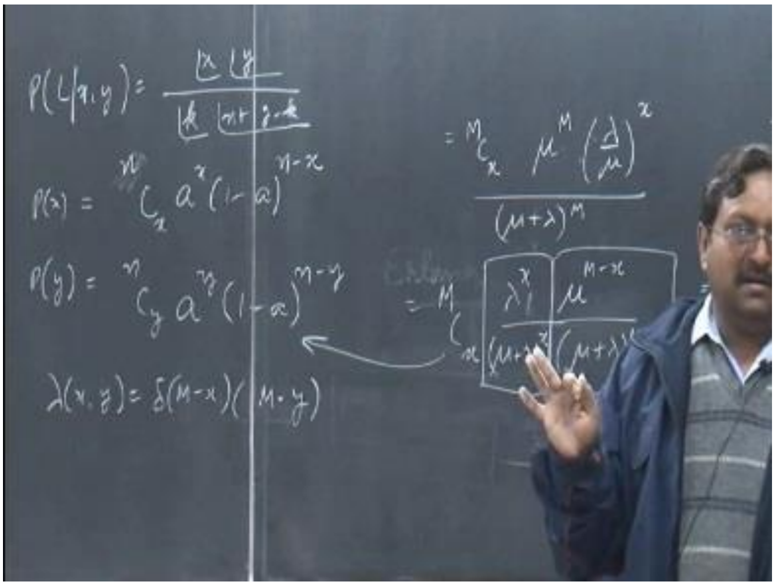
So, now solving… I need to just convert everything to a and this will give me this thing actually. You think this is the step, where I missed out. So, I require… And of course, 1 minus of a is 1 minus of lambda plus lambda plus mu. So, I will just do that conversion. So, from here you know what has to be done.

(Refer Slide Time: 07:25)



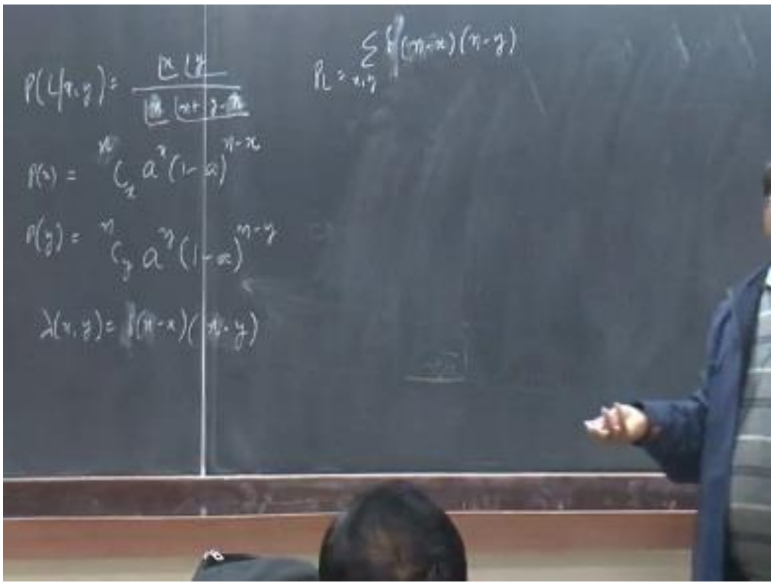
So, mu M minus x. So, I have to take out mu plus lambda M minus x. So, here mu minus mu plus lambda; x will be left because this sum has to be equal to this; mu M minus x I have already taken; lambda raise for x will be here. So, take this and take this and you will end up in getting this. So, P x and P y – both are known to me.

(Refer Slide Time: 08:01)



I have already determined what is going to be lambda x y; this also we had determined. We also had determined… This is a middle state switch – k… So, k is the number of switches in the middle stage. With that we had also found this. So, all things are there; I have to now put it in that summation and then solve for it actually. So, I will do this.

(Refer Slide Time: 09:01)



So, remember it is a dual summation. This has to be…

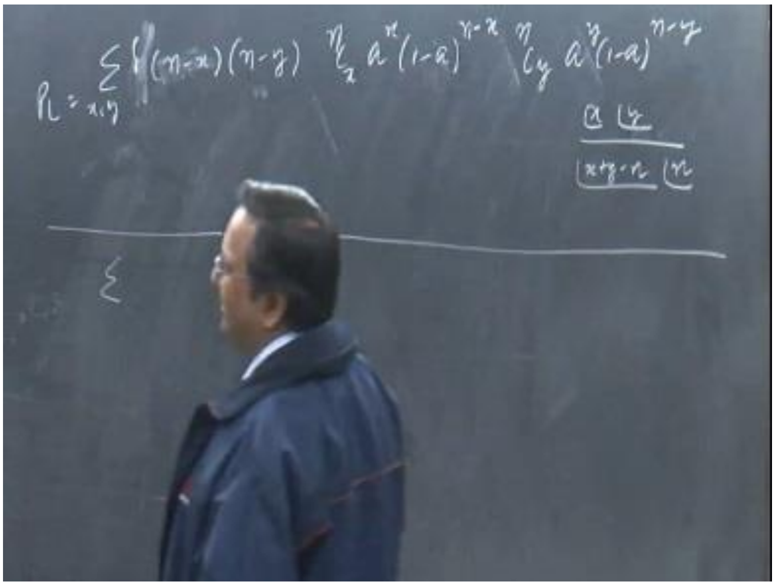
Student: ((Refer Time: 09:18)) For this lambda x y equals a set sigma delta n minus x into n minus y. And similarly on the top equation of ((Refer Time: 09:31))

You are right because r will be equal to n; I am using a different notations that is… But, anyway now, I will not use this, so that there is no confusion I do it correctly. So, it will be n minus x n minus y…

Student: (Refer Slide Time: 10:28))

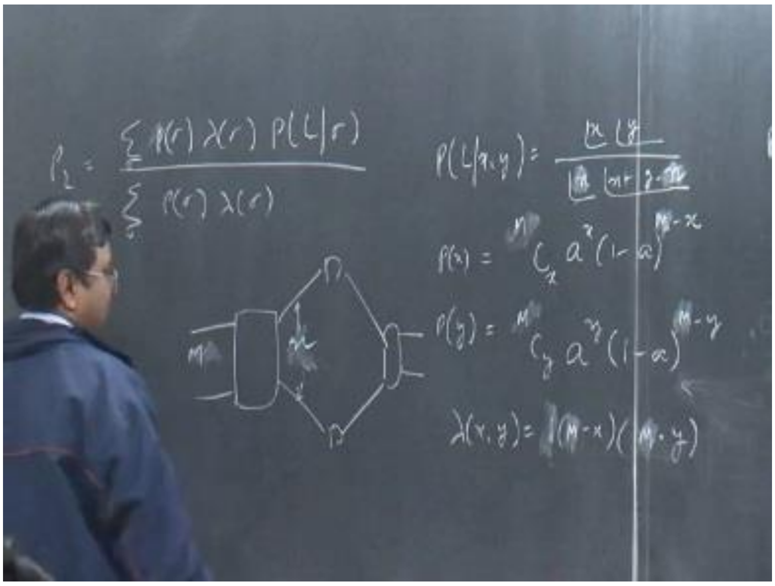
Summation over x y; x y is the… Delta or gamma? What it is used? I think there I have used gamma last time; just check in the previous… I have used delta. So, I will keep on… This anyway will cancel; I am not bothered about it; this anyway will cancel. That is why I have taken a constant for which I do not know the value. So, I have put in the lambda sigma, has been put.

(Refer Slide Time: 11:11)



Now, let us put P of sigma, which is multiplication of these two: x and y both are independent. So, I have done this P x P y. Now, conditional probability x factorial y factorial divided by x plus y minus n n factorial. So, numerator part has been done. What I will do is I will try to solve in this equation and I will go slow, so that… because you have to rewrite the equation; I can erase the board and do it. And this will be… I think this cannot be n; this has to be k, because…

(Refer Slide Time: 12:34)

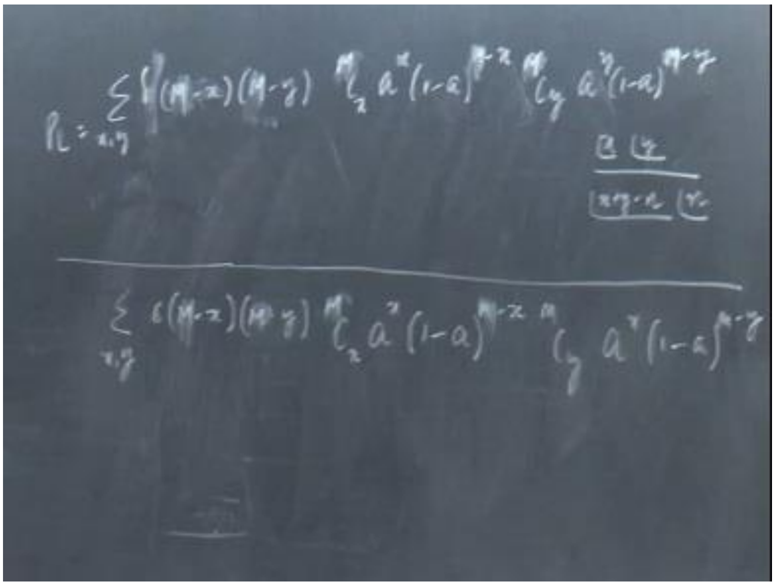


The reason for this is… This I think the way I have taken, because k is greater than n; that is why this gets converted to n C y ((Refer Time: 12:54)) n maximum connections can take place at any point of time.

Student: ((Refer Time: 12:58)) Input line is m, sir?

Input line is M and this is n. So, in that case, I have to use M here actually in that case. And look at the expression; this should be m. So, I think to be consistent, I will just change… Wherever n is there, I will just put… This should be again m; I can observe the inconsistency otherwise. This is then n; it will be fine. So, this is actually n and this we are using M, because the derivation is with n k actually. Is there anything else wrong? No.

(Refer Slide Time: 13:43)



So, this I have to change; this is fine. So, kindly verify that everything is correct. So, we expect this notation now.

Student: Lambda sigma in terms of n ((Refer Time: 14:19))

Lambda sigma… No, this cannot be n. Incoming line – how many are free?

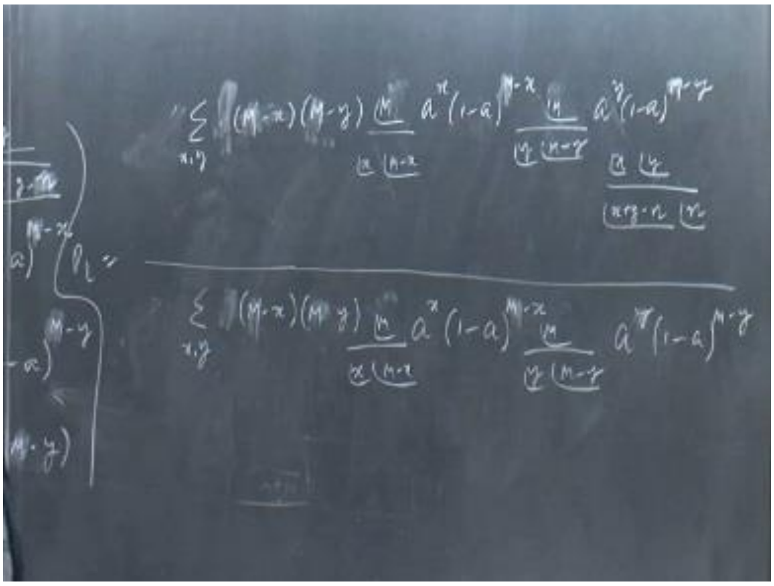
Student: ((Refer Time: 14:27))

Small n is greater than M.

Student: So, number of ((Refer Time: 14:33))

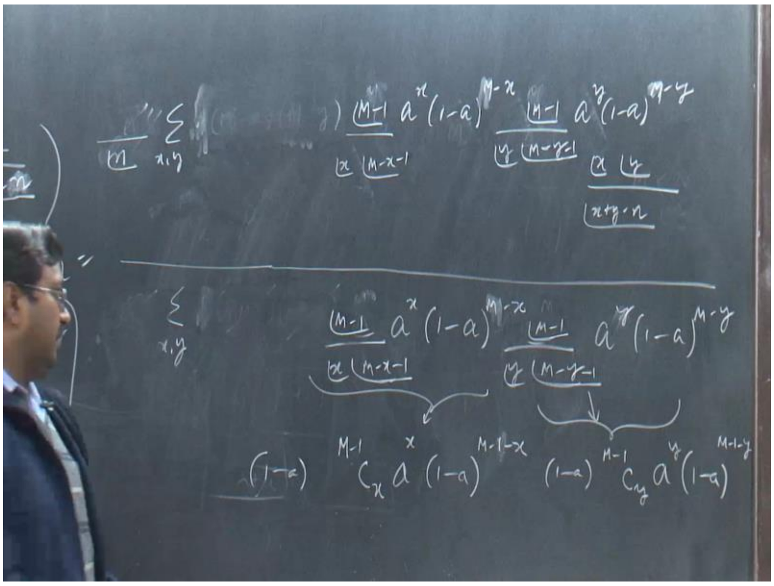
M; number of maximum calls can be n. Even if n is larger, you cannot have more than M calls, only M I am (Refer Slide Time: 14:47) can be occupied. That is what I derived; I think there is same confusion n, k things. In my lecture notes, I think it is written as n k on the website; that is where the mistake is. But, anyway forget that website. I have to consistent in whatever I am doing in the class. And you should be capable of modifying all the symbols whenever required, because I also remember it was n k; but, then since I have been doing M n. So, I will be doing M n, not an issue.

(Refer Slide Time: 15:32)



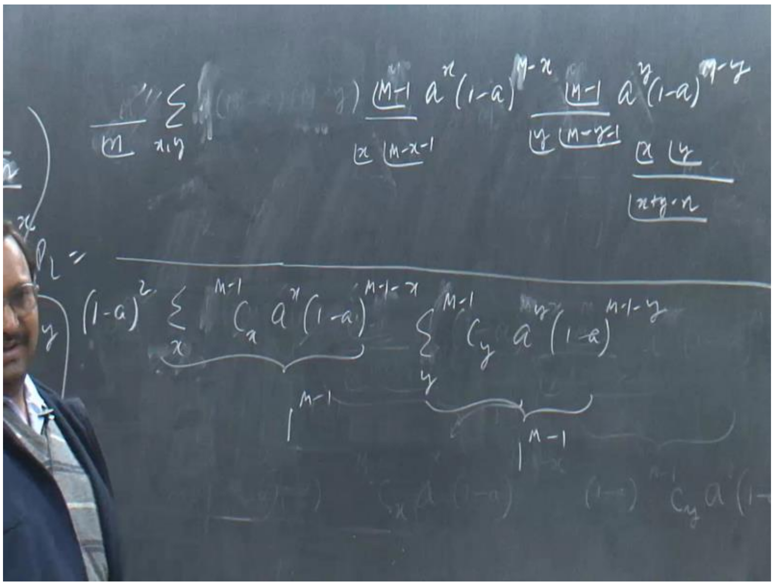
So, I have to actually now… I cannot take M minus x and M minus y out of this x y summation; you have to understand this. But, I can take delta very clearly without any issue. Here delta can be taken out; this delta will be canceled. Expand this M c x and M c y. Do the same thing here. It is nothing but step by step evolution now. So, I can cancel this and it will become minus 1; this will become minus 1.

(Refer Slide Time: 17:05)



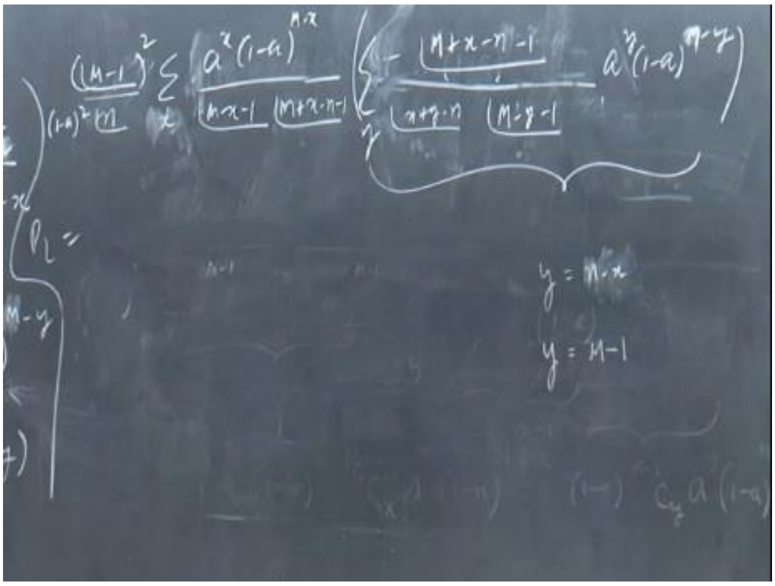
I can write the same factorial x. Similarly, this factorial can be written as… I can take these M square out; I can do the same thing on the numerator side. So, I will end up in M minus 1, M minus 1, minus 1. Is this fine? n factorial is independent of x and y. So, this can be taken out. So, M square cancels. I can remove this. Now, this particular thing can be written as… If you look at this expression, this is nothing but M minus 1 C x – M minus 1 minus x. So, there is one extra 1 minus a, which will be there. Similarly, for this, I can also write. 1 minus a; M minus 1; c of y.

(Refer Slide Time: 19:05)



So, I can very well replace this whole thing. And 1 minus a square can be taken out; x and y – both are independent here. And there was earlier one extra term; that will actually become zero; obviously, which you can verify. So, one term has been now reduced; these two are independent. So, it is nothing but complete binomial. And I can replace, I can actually simplify this thing as summation over x and this as summation over y; that is possible. This thing is nothing but 1 raise power M minus 1, which is equal to 1. So, this whole denominator is gone. Perfect; do you have only 1 minus a square.

(Refer Slide Time: 20:54)



Now, let us look at that top stuff what we can do. I have to use some trick here. M minus 1 factorial square actually can be taken out. So, this no more required. And now, I will split into two summations: one is x; other one over y; I will do x later; first over y and second one over x. So, anything which is having y; I will keep it on this side. This x factorial cancels with this; this y factorial cancels with this; this is what you have. So, I am just going to rewrite it actually. Let me rewrite first. So, that is what you are going to have. I can now multiply and divide; just a minute I will do something.

Student: ((Refer Time: 22:26))

I have to do that. I am just splitting that. This one I will keep here; summation these two terms actually will come here after this summation. So, this is what is a summation for y; that is summation for x. And I will then multiply and divide by something. So, what is sum of these two? That has to be independent of y; that is the only condition; it should not be function of y. So, one is having plus y; other one is having minus y. So, it will actually cancel if I add. So, I can multiply by that – whatever is the sum – factorial of that I can put that in the numerator as well as in the denominator. That is the simplest way of doing it. So, it will be m… Just add these two – plus x minus n minus 1. You can put in the numerator; same thing I have to put. And this I can take out, because it is independent of y. So, this one is…

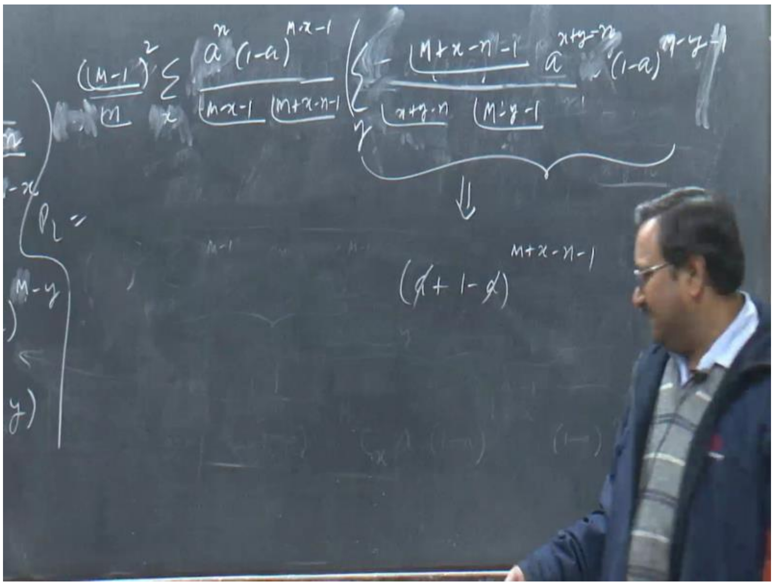
Now, this is 1 minus a square; I need not even keep this line; 1 minus a square I can just put it in here. And this thing; this is summation over y. Now, what are the valid values of y that you have to figure out. So, valid values of y will be… They will be ranging from… So, first thing is this actually should become 0; and secondly, this should become 0. That is how you will actually get the range. So, y will be actually varying. One extreme it will be x minus n.

Student: ((Refer Time: 25:06))

n minus y; you make it 0; n minus x. Now, this is the condition when x plus y is equal to n just when the blocking starts. If x plus y is less than n, blocking cannot happen. This is just when the blocking starts. This is the first case. The y will actually range from this value. For rest everything, this will go to 0 anyway. You cannot have summation over those terms; other terms have to be removed. Even if for another possible values of y is there; there is 0. I am only going to sum up only this range when blocking happens. Only those terms are there. So, I am not estimating when blocking is not possible; those cases I am not considering in my term. So, that is why when that condition x plus y is actually smaller than n happens; those cases are not coming in. If that situation satisfy for a switch, this will give you a value 0 actually. So, we have taken care off.

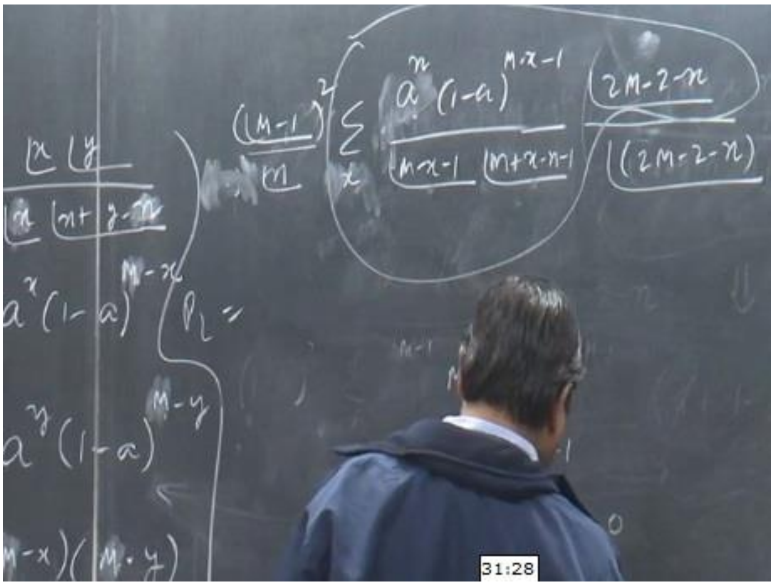
Earlier case – in Lee's approximation, you were not isolating those conditions. We are not removing them. Here in the summation term itself, when the blocking is happening, I am removing those, because those become invalid values; otherwise, I cannot sum it up. If I take y is equal to going from 0 to n, it is not possible, because terms will be invalid in that case; I cannot build up a closed form solution. So, while building up closed form solution, I am actually removing or excluding those terms. This is actually… Usually, it is not evident. So, I am explicitly stating this. So… And the next range will be coming from here; y will be M minus 1 obvious, because there is one line, which is free for which we are trying to… On which the call is coming. You are not looking blocked state; you are looking at arrive… When a call comes, whether it will get through or not get through. So, in the worst case, it will have M minus 1 states. You cannot have higher than this. This is not possible. So, this is the range over which this summation will be done. Then only I will get a closed form solution; otherwise I cannot get a closed form solution. So, because the moment I take the summation… because this is nothing but a binomial now. I can now convert it into binomial. I have to put something…

(Refer Slide Time: 27:37)



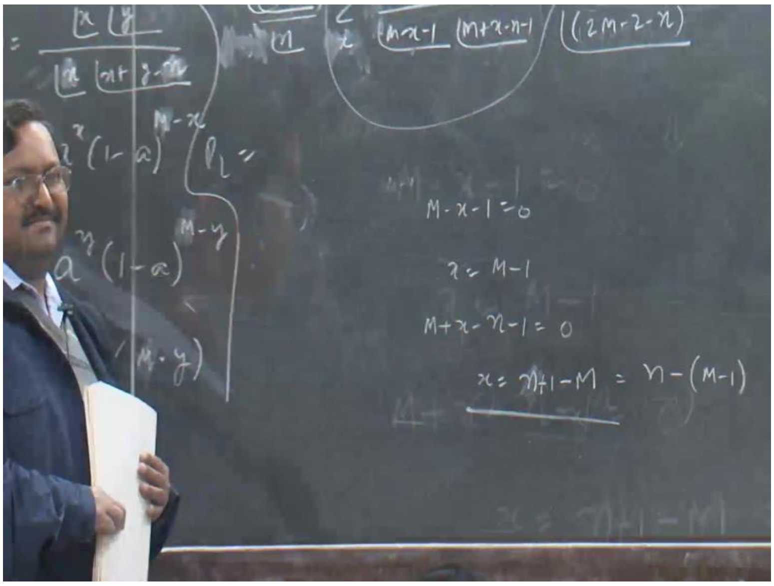
This a raise to power y is there; x – I can take inside now. I can call it a x plus y. There is minus n here. So, I can put… Just a minute. This is minus 1. So, I have only one left here. I will also even remove this. And this will be converting into 2 minus 1. I will put… I require minus n here; I will put a raise power n here. So, still everything is in balance. Now, this is value matches to this. And M minus of this or basically this value comes here. This is now complete binomial. This is the range, which is also a valid range when the blocking happens. I am removing all other terms. So, I can do the summation. And I will end up in getting… So, range concept is very important because usually… Unless you appreciate this, you will not be able to figure out that, why you are getting the correct probability of call blocking. Correct you are getting, because I am not taking this summation. Other terms which corresponds to the cases, where blocking does not happen; those have been removed. So, this now, you can solve. So, y variable is not there on this side. So, what will be the value? a plus 1 minus a raise power M plus x minus n minus 1. That should be the value. p plus q raise power n in binomial. This is nothing but 1 raise power something. This whole thing is unity. Once it is unity, I can simply remove this. Excellent; we have got this; one step closer to further solution.

(Refer Slide Time: 29:52)



Now, if I add M minus x minus 1, this M plus x minus n minus 1. If I add these two, again I will get a range of x by keeping this as 0 and then this as 0. So, sum of these two is how much? 2 of m; x will cancel; minus 2 minus n. So, I have to just take the factorial of this and multiply and divide; done? This anyway can go out. So, I am not bothered; I am only bothered about now this particular piece. Again you have to look at the range. So, what terms you are excluding? What all values – possible values of x you are excluding is important here.

(Refer Slide Time: 31:08)



So, now, one possible range will be M minus x minus 1 is equal to 0 you put. So, x will be M minus 1, which is valid, because that is the maximum, which you can get. But, what is the minimum value of x? So, x will be n minus 1– n plus 1 – minus m. That is the minimum value of x, which is there. I remember in earlier case, it was based on – y was ranging… y’s range was depending on x actually. When you are computing range for y, it as depending on x. So, whatever… But, x has to be minimum chosen as… Any one of them can be least; actually depending on what is the value of n. If n plus 1 is on this side; remember this – these many number of ports and these ports. Physical… These number of ports are double the… This n plus 1 is greater than or equal to double the value of m. Then, x will be… This value will be… This… Usually it will be lower actually. This will only be higher if it is more than double the value of M.

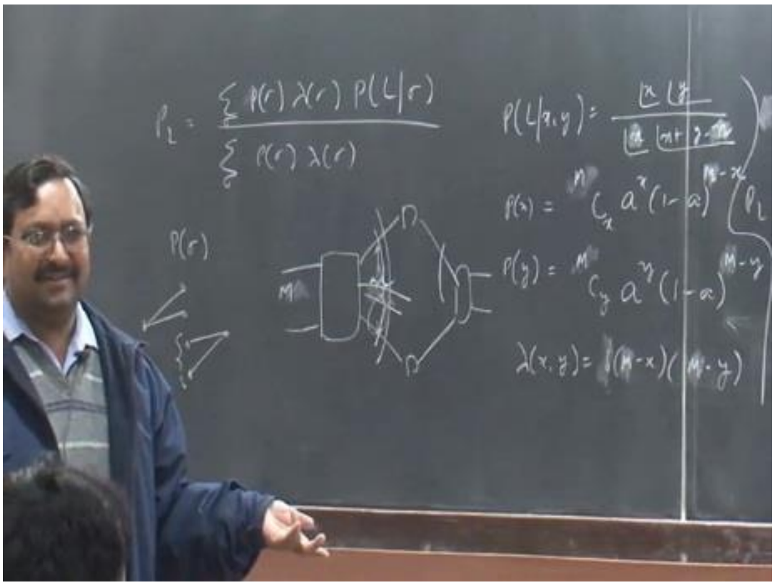
Student: ((Refer Time: 32:54)) Highest will be x and ((Refer Time: 33:06)) That we take M that…

Only M minus 1 calls are there; worst case scenario.

Student: So, n minus M minus 1 is equal to ((Refer Time: 33:19))

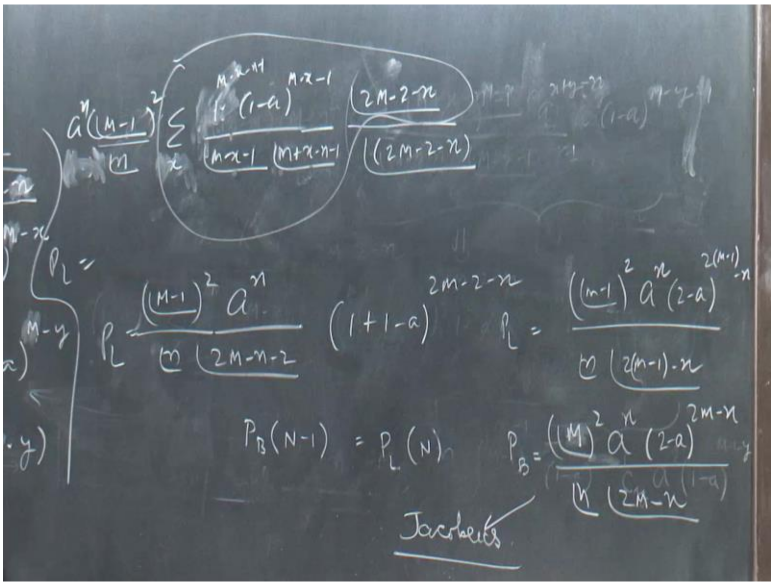
You are right. This will be… because if you actually go out of this range, again the blocking will not be possible; blocking will not happen. Now, why this is happening? If n minus M minus 1 is this; if x is less than that; on the other side, m minus 1 can be occupied. And I am having value less than this; there is always possibility whatever combination would take; it will be always non-blocking only if this difference is fully covered.

(Refer Slide Time: 34:05)



Remember I have done a case. In this case, can blocking happen? Blocking cannot happen. Blocking has to happen only if whatever is the left over is covered by this minimum. Minimum value has to be this. And that is what I am writing – n minus M minus 1; n minus M minus 1 is total subtracted. That is this value. So, this value minimum has to be equal to this; and maximum it can go to M minus 1 anyway. So, this value has to be lower, because x cannot be higher than M minus 1. So, this is the lower range – lower value. And this is higher. And the case which I am taking when n minus M minus 1 becomes equal to M minus 1. That is exactly non-blocking switch in that case; probability of call loss will be 0. That will immediately… So, with that summation, all terms will go out actually in that case; you will not be including any term in that case.

(Refer Slide Time: 35:23)



So, again it means I am only taking the values in this range. Then, it will be a complete binomial; otherwise, it will not be. So, I can now solve it. So, I have M minus 1 factorial square n factorial 2M minus n minus 2. And this is nothing but a plus 1 minus a whole raise to power 2M minus 2 minus n – complete binomial; a cancels with this; it is 1. So… No, I made a mistake. What mistake I have made? I have done something, but it is incorrect. You have to have these terms, which corresponds to these. I do not have term for a, which corresponds to this here. So, I cannot simply write a plus 1 minus a and then say equal to this; no, that is incorrect. This a raise to power n actually should come out. This is independent of x; this does not vary. So, I have to take it out. I have to put some other term here. So, I will put 1, M minus x minus n minus 1. That is the way of doing it actually. So, once I do it, it is 1 plus 1 minus a…

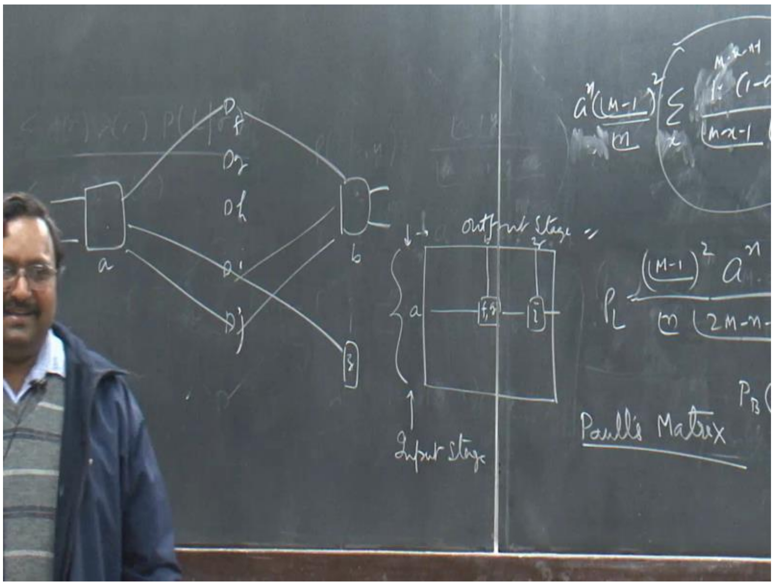
Student: Sir, this is we have a raise to the power n…

Yeah, a raise to power n, because we had minus n was created or used earlier in the previous step. In the summation, where we did summation over y, we have moved it inside – a raise to power minus n was required. So, we created a raise to power n here. a raise to power x actually was moved inside there, so that you can have a raise to power x plus y minus n. So, this is what is the closed form expression for probability of call loss. So, I will write it as M minus 1 factorial a raise to power n 2 minus a 2 M minus 1 minus n; n factorial 2n minus 1 minus n. And remember this is nothing but call blocking probability. And we have estimated that, call blocking probability will be function of n minus 1. If n minus 1 is replaced by n, this will end up in… This is the switch being blocked; this is the time congestion; this is a call congestion; this we had already computed earlier for a composite switch.

So, there is an alternative proof, because there is a call blocking probability. You can also do when the switch will be in the blocking state; that estimation. That is known as Jacobeus approach for call congestion estimation in the same 3-stage network. And you will actually get the result, where P of… This is P of L. P of B will be… This is M… Jacobeus got this expression. And can you observe these two and these two? The same relationship still holds. Same relationship still holds. So, this… There is an alternative approach through Jacobeus – the name of the gentleman, who did this alternative derivation. So, that gives you the call blocking probability.

So, now next, we have to go into the all kind of theorems, so that we can formally figure out that, why a switch is still ((Refer Time: 40:08)) blocking or why a switch is rearranged every non-blocking; and why a switch is wide-sense nonblocking. And can we reduce their cross-point complexity further? I want to minimize on cross-point complexity as far as possible. So far, what we have figured out is O and 3 by 2 for a strictly nonblocking switch. Can I make still better than that? Yes, we can do still better than that. We can get O n log 2n; log 2n's power 2.58. But, how this 2.58 comes is interesting. This can be derived; it is not ((Refer Time: 40:48)) number. We will do that derivation…

(Refer Slide Time: 41:26)



But before that, when I go through now formal proofs and formal theorems regarding this, you have to understand a concept of Paull's matrix. So, again it is a name of gentleman switching theory. And this is a very handy tool in giving all kind of formal proofs, theorems ((Refer Time: 41:12)) Everything comes from here. So, I have sufficient time to actually to explain what is this Paull's matrix is; what is the meaning of this. 2 by 2 – it is simple to do, but if multistage, number of ports can be arbitrary – any kind of thing. And then, you have to do a formal proof; you require this. So, you will have 3-stage switches. And I have lot of middle-stage ones. Important thing is that, if I take any other input and output pair switch, the path between these switches is not blocked by the paths, which are corrected between these two switches. That is extremely important. I can set a path between this and this independently of this and this. If a path between this and this is being set up, that may lead to blocking of a path being setup between these two. So, at least either input or output, which has to be common for the blocking to happen or further interference – interference in setting up of the switch. So, in fact, I need not look into this; that is very generic. I think you should be able to appreciate that.

And, I will give now some values say a and b. This can be numbers; numbers are also nothing but symbols to represent. They are formed from a set; but, we have built up our own arithmetic on that or an algebra by which we can define the complete number system. So, it does not matter for me. It can be any arbitrary symbols; that is good enough, because the algebra here is pretty simple. I can write this as the middle-stage switch – f, g, h, i, j – whatever you want to put. And what we will do is we can always create a matrix. That is why it is known as Paull's matrix. And if for example, this – between a and b, if I want to set up a connection. So, all the rows corresponds to the input stage switches. So, these are input stage switches. And what should be these columns? Output; this is obviously. Remember the science is very symmetric in nature. So, if you understand symmetry and beauty of it, it is going to very simple. Anything which is ugly – it is probability incorrect; that is a thumb rule.

And, you will have here output stage switches. So, I will just explain something and then leave it for you. And we will go ahead with this particular thing later on. If a switch a is here; switch b is here; so number of switches, number of columns; number of switches, number of columns. a and b – I can set up path through many of them. So, if I set up a path through f, there is already a path happening. So, I will put an entry f in this cell, which is common. But, the cell need not have one entry; it can have multiply entries also. If I am using g, h, i, j also to set a path between these two, I will put those entries also – f, g, h, i, j. But, f and g has already been used for a and b. But, a I am using to connect to some i to some other – say some z. So, for the z column, in that cell, I will put this i. So, that is what is known as Paull's matrix.

Only thing now you have to understand is – I am actually giving it as an assignment. You try to think about it. You can even look into the notes; that is your choice, but I will appreciate if you try to think on your own and come with the conditions; and later on verify with the notes or anyway I will tell that. What are the conditions of validity of this particular Paull's matrix? If I give you a Paull's matrix, can you check whether it is a valid Paull’s matrix or not. Conditions like how many times entry can come in a column; can it repeat. I have only f – can come only once in the row; or, f can come only once in the column; can f come at 2 or 3 places; how many entries can be there at most. So, try to think about all those conditions. You have actually understood quite a bit already. Only thing you have to just convert it into now this. And amazingly, you did not visualize the switch after this; you can just work with this Paull’s matrix blindly after that.