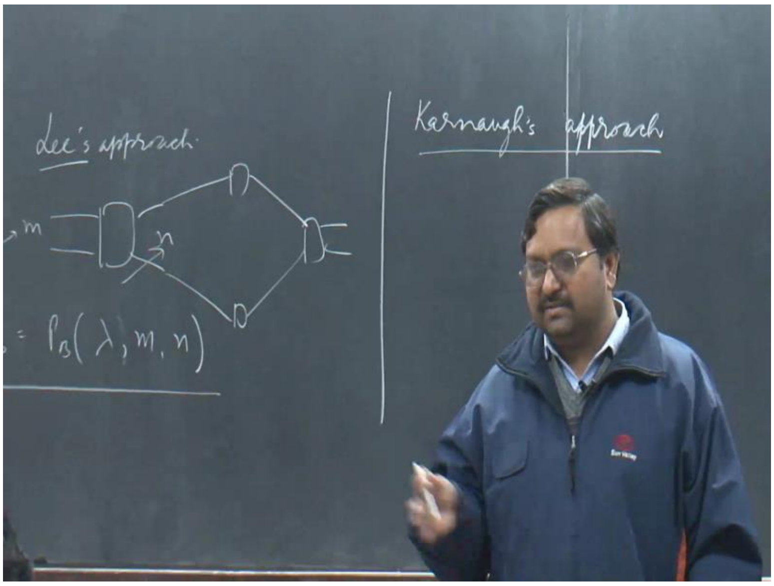
Digital Switching
Prof. Y. N. Singh
Department of Electrical Engineering
Indian Institute of Technology, Kanpur

Lecture – 6

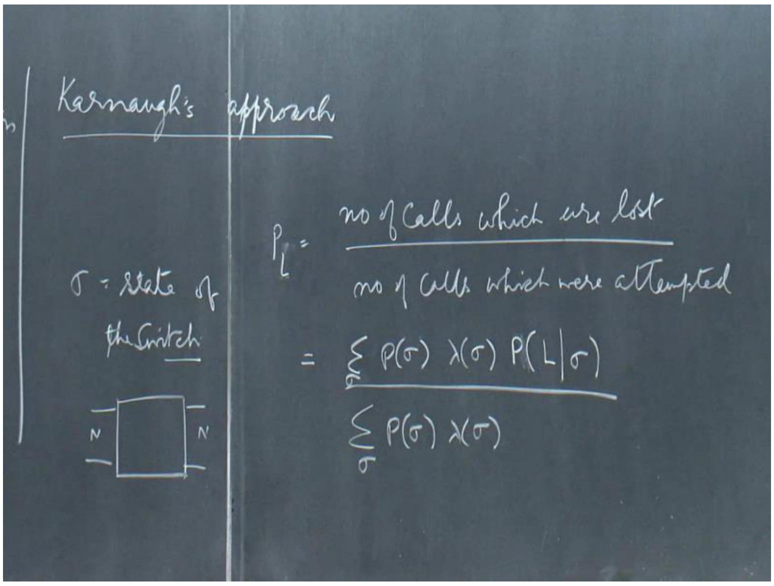
(Refer Slide Time: 00:20)



So, you are actually making the probability of blocking estimate P B. And for a lease approach, which is very simplified thing; the basic assumption was that, irrespective of the state, probability that input link and output link – both will get occupied – was all independent of the states. It was just simply P and we have approximated that using a balance equation and we got the expression. This will usually work for larger dimensions of the switch. So, whenever the value of a is very large; value of… In that case, we have taken I think m and n; m as input and n as the output. So, whenever this is going to be very large, this m and n – that approximation usually will work, but not for the smaller values. But, we can make still better approximation. Then this Lee's approach – this was because of Lee.

So, today we will do Karnaugh's approach. And in fact, nowadays, because computers are available, you can actually look… you can iterate through all possible states of the switch and actually can numerically compute, the precise call blocking probability. That is technically feasible. You have to just write the appropriate software for a corresponding architecture; do all that computation and come up with probability of blocking for a given load condition. And remember probability of blocking will always be function of… It will be always be a function of arrival rate lambda and the ports, which are there: m and n. So, even the expression actually, which we did; there was m; there was n; and there was a. a was equivalent to lambda – arrival rate – arrival probability. It is always function of these three things. So, this can also be done numerically. But, we are not bothered about the numerical simulation and estimating from there. This is a close form thing; a classical approach; and we will do that.

(Refer Slide Time: 02:49)



So, we have to assume the probability of blocking. In fact, I should call it a probability of loss. So, number of calls, which are lost… Basically, you are trying to make a call, but you find that, resources are not available or path cannot be set up. And we call the call is lost in that case. It is not that call was already there; something happened and call was lost; that is a call drop. Call loss is when you are trying to set up a call, but call cannot get through, because the resource was busy in the switch.

And, call actually means a connection between input and output port of the whole switch. It is again is a three-stage switch; number of calls, which are lost and number of calls, which were attempted. So, this has to do with the… This is call loss probability technically. In earlier case – Lee’s thing, we were only estimating the probability of switches in the blocked state. So, that was time congestion. So, here Lee’s approach gives you time congestion. And then, there is a third approach, which is again very precise, very similar to this, but which is used for estimating the time congestion directly. But, we would not do that.

I will just state the result of the Jacobean's approach, which is the third one. And this gives you the probability of switch being in blocked state for the same switch using very similar model. So, in this case, I will assume that, sigma be the state. Sigma is the state of the switch. So, I am not going for the state dependent variables – the state of the switch. So, I can write this thing as probability that switch is in a state sigma. That is the first condition; you have to do always a summation with respect to that. The calls will arrive; arrival rate also depends on the state of the switch; m probability – you are in a state sigma; under that, what is the conditional probability that call will be lost? So, call has arrived and then there is a conditional probability on that; that call will be lost.

Student: ((Refer Time: 05:38)) Sigma means…

Sigma is a state of the switch; how many calls are through; what is the current setup. So, you can always say the switch; there is no connection setup. For example, this is 2 by 2 switch; no connection is set up; this is one state. Another possible state is there is only one connection being set up; this is this. There can be another state, which is this; this for single connection only. But, these two connections are mathematically equivalent; can be represented by only one entity. There if you set up two connections; one possible state is this; one possible state is this; they are also equivalent; you just twist the switch input-output thing. How you enumerate or put the numbering? This can be converted to this. So, these are the possible five states of a switch. So, in F 6, since I am identifying, the switch is state by sigma; no call, one call, two calls; if there are three switches, 3 by 3 switch; there will be more number of states; of course…

Student: ((Refer Time: 06:55))

This is sigma – overall states.

Student: About the two states, you are talking the ((Refer Time: 07:03))

This is the general switch. You can actually do an exercise that, take N by N switch – cross bar. You have to set up one connection. How many possible variations are there in this case? How many states will be there? You can take the first top input; it can connect in N possible ways. So, N states. Next one – N states; N square states total will be there with single connection. No connection – only one state; with two connections… So, there we have N C 2 possible ways you can take the inputs; they can be connected to N C 2 ways; they can be connected to the output.

So, based on that, you can actually make an estimate of what will be total number of switches. Let us say it will be huge number; it is not going to be small actually For zero connection, for one connection, for two connections, three connections, we will use that technique here and abstraction. And as I told that, for example, these two switches states are equivalent states. So, technically, this switch has only 1, 2 and 3; only three possible states, because these two states can be merged in one state only. So, we are going to also use and limit our search space or what you call a state space; we will try to limit that to a very narrow range. That trick you have to always use.

Even if you are doing a computer simulation to identify a switch performance, you have to do this; otherwise, you require a tremendously large amount of computing capacity and we will be doing redundant computation. So, that of course, if you try doing the simulation, you will understand that. So, let us come back… I will just come back to this state, but let me write down the expression first. So, I think motion of a state is clear. N-state transitions happen whenever a call goes off, whenever a call arrives in. And whenever a reorganization happens, there can be multiple-state transitions, which are possible whenever rearrangements are done. So, this is the call loss. Call switch are lost; this is equal to whatever was written in the numerator. These are number of calls, which would have arrived; total number of calls, which would have been attempted. So, this is the blocking probability or call loss probability. So, we have to find do this.

Now, what is sigma question is this. Sigma has to be interpreted in some way; I have to find out some state variable, so that it is finite actually and we can easily solve it – small sigma, not capital sigma. I will always call this as summation, this as sigma; I will never call that thing as a sigma actually; that will be summation. So, now let us do this.

(Refer Slide Time: 10:29)

Student: Number of paths… Number of call from ((Refer Time: 10:30))

Small sigma is not… This is an abstract thing; you cannot identify by… For example, this figure – you cannot give a number, but this is a possible state. This can be identified by some variable, some notation – some abstract notation. So, it is a set of all these possible states and each set. You can it a for example; you can call it this b; you can call it c; there is no number. So, sigma is that abstract entity, which belongs to set of all states.

Student: ((Refer Time: 11:05))

In this case, you cannot do.

Student: ((Refer Time: 11:10))

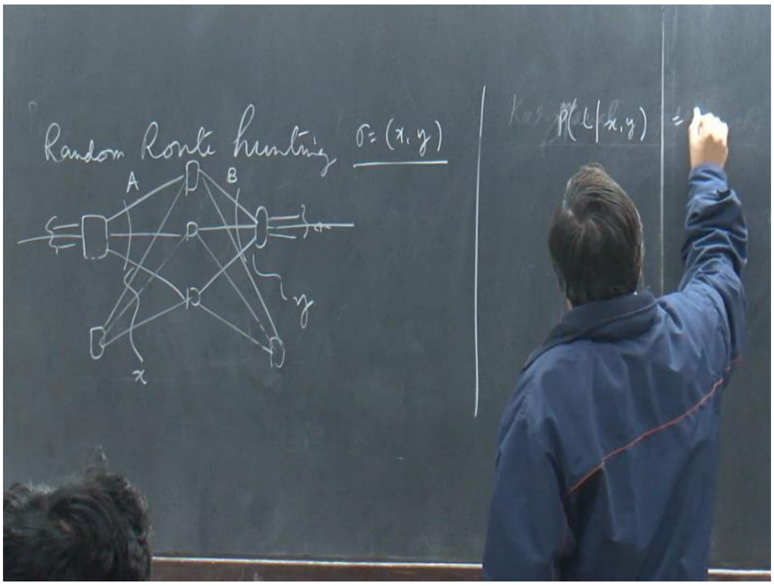
There also Markov chains are also similar concept; when you write the state variables, that is an abstract thing. Of course, when I did a queue thing; I always say let the state being represented by number of elements, which are there in the queue. So, with my convenience, I transform that, abstract notation of a state to the number of nodes, which are there sitting inside it. So, I was able to identify state by the number of packets, which are queued up in the queue. That was my convenience. But, in general, state to space notation in Markov chain or any place does not bother about a numeric value. So, all the states form a set. Whatever state variable you are taking; that is member of that particular set.

In fact, for that matter, even numbers are nothing but they are also members of a set; whole mathematics actually is about sets. So, 0, 1, 2, 3 – whatever it says going to be member of set, which goes from 0 to 9. There are only 0 to 9, is 10 possible members of a set, which represents a digit. And every number is, every digit is coming from the set. Combining multiple digits, defining rules, you are able to build up the whole mathematics after that. And only what if we do is, we start making a, b, c, d also as members of the set; start doing algebra on that. We feel awkward, because we do not count the oranges by a, b, c, d; we count oranges as 0, 1, 2, 3 or any counting, which we do. So, that is a perception.

Student: ((Refer Time: 12:59)) number of…

Number of packets inside this or number of incoming ports, outgoing ports, which are getting occupied. Here also I will do the same thing, but it is not going to state variable; one single state variable. There will be two variables, which are independent of whether combination of those will give me the state definition.

(Refer Slide Time: 13:26)



But, we have make certain assumptions before that – before I can define this thing. So, again, I am taking lot of metal state, which are there; I am only taking one input and one output, because if I take some other one, this will also be connected to everybody. So, between this, it is independent; it has no conflict when I am trying to set up connection between these or I take this particular pair. So, I can take only one input; one particular case and one particular case here. And based on that, I can make a judgment this should be true for any other input switch or any other output switch also. So, I need not bother about the whole design, I will only bother about one single input state switch, one single output state and all middle state switches, because they do not have any conflict; they can be independently analyzed; I can take this pair and analyze.

We are now going to take a very important approximation. Approximation is from each input port to each output port; the connection is made with uniform probability – uniformity of traffic actually; extremely important thing. Once this is there, the probability each one of these links will get occupied; I am also again assuming to be same. So, when you want to, for example, set up a connection between a free link here and free link here, you have to find out first of all a free path here, then a free path here has to be figured out. Now, there are ways and means of doing this. You can start doing search from 1, 2, 3, 4, 5; you get a free link; find out there whether there is a free link available, not available; go next 1, 2, 3. So, you can… You are now counting in an order till you find a free link and you set up the path.

Next connection request come; again you will go in the same order; I am not assuming that; it is not an ordered setup. What it will do is it will randomly pick up; just randomly pick up whatever is a free link; and based on that, it will choose; it is not in order; that is very important. We call it random route hunting approximation.

Student: ((Refer Time: 15:51))

No, otherwise, I cannot build up the state variable model. So, I want to simplify my calculations. So, at least whatever I am going to build, you can only… What we call… It will give me some kind of an idea, some kind of a bound actually on probability of call loss probability, because at least this much I can achieve; it can be only worse than this; it cannot be better than this; or, it can be better depending, but this random route hunting gives me a bound essentially. So, we are estimating that. So, in the random route hunting, these and these are independently being made busy. What we do is we assume the x links are busy here and y links are busy here. We usually call them A links and B links. So, number of a links, which are busy are x; number of y links, which are busy here; why? It is not question of one path or two paths.

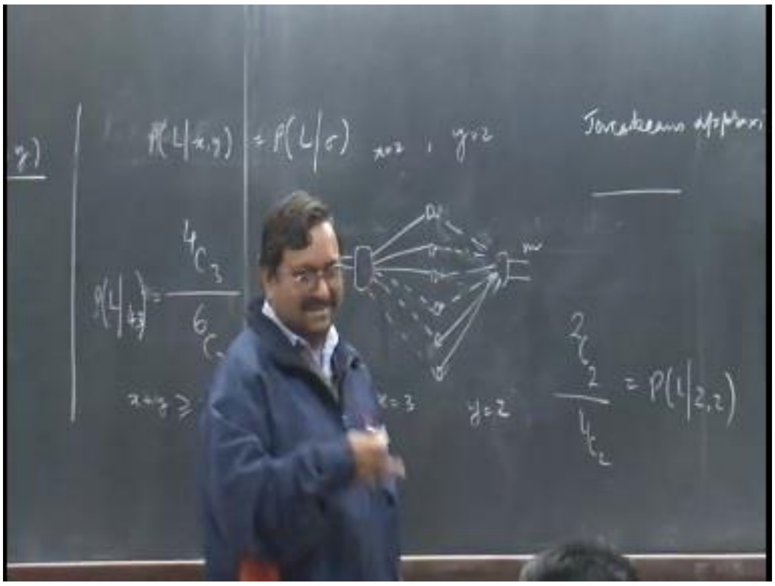
Now, if you are careful enough in observation, while the removing these, I have actually merged lot of other states into these states corresponding to only these two switches already; one merger has been done; otherwise, there would be a larger R 1 number of switches here; again same number of switches – output states switches. So, all other inputs also will be connecting to these. So, actually, number of states should be huge, but I have merged to all these possible states into states, which are only because of this one input and one output and middle state switches.

Student: There are particular instant of time ((Refer Time: 17:45)) particular state...

But, for analysis propose, because I say because these are independent devices; I do analyze for release for this; analyze for these two pair; analyze for this or this pair; they are all independent. So, I am only analyzing this technically means for all other pair combination of input and output switches; I have merged all of them into one single scenario. So, already one merger has taken place. Random route hunting again further simplifies; I can now only represent by x and y; it should have been ordered situation, the things would have been far different. So, I can now represent the sigma as nothing but x comma y. So, x and y pair; I cannot add x plus y. So, it is not a single state variable; it is a two-dimensional state variable. So, when I am doing summation over sigma, this will be actually two summations: 1 over x and 1 over y.

And now, blocking probability is dependent on the values of x and y; I will explain how that happens. So, I have to now estimate what is P sigma, what is lambda sigma, what is conditional loss probability. For certain states, loss will not happen; I have to take care of that. While in case of Lee’s approximation, we were not bothered about state; there were states, where there was no blocking, but you are just making an approximation there. So, the state will be sigma is equal to x, y. You have to find out P sigma, lambda sigma and P L given sigma – three entities. So, we will take them one by one and solve; we will try to understand how to build up a model for this.

(Refer Slide Time: 19:50)



So, let us take… I think the first thing, which I should do is which is nothing but probability of call getting lost or probability of switch being in blocked state, because call arrives; at that time, switch is in the blocked state; then only the loss will happen. So, it is a conditional probability there switch is in blocked state in a given state. And within that state, I am multiplying it by arrival probability. So, it becomes call loss actually in that case. But, this is a blocking switch being in blocked states – probability for that – conditional one. So, this is an interesting thing. I am taking a very example of four middle state switches; these links can be occupied at any point of time. So, let me take x is equal to 2. So, x can take value from 0 to 4; there are only four middle state switches; no rearrangements are permitted, remember. So, x is equal to 2 and y is equal to 2 I have taken.

And, if this is a scenario, these two links are being used; these two are free, remember. But, these two are used because this switch – this particular path is being used to set up maybe somewhere else actually like this. So, that is why I am… This is because of the independence of these A links and B links; you cannot set up a path in this case; the switch is in blocked state. You can have a scenario when… You can set up a path through the bottom link. These arrangements can happen; this combination of two occupied once happens independently out of all possible variations. This happens independently actually of A links for B. So, what I have to do is I have to now only enumerate all possible combinations and find out the cases when the switch is in blocked state. So, number of enumerated states, because now, I am looking at the state even for x is equal to 2 and y is equal to 2. I told you x is equal to 2, y is equal to 2 is 2 comma 2; that is one state. But, that actually is not one state; there are so many possible states.

Now, again there is a question of merger. I am representing this state by 2, 2. So, all states for which x is equal 2, y is equal to 2 are taken care of by this Jacobean's approximation – the same gentleman. So, ((Refer Time: 23:13)) another merger of states, which happen. So, I have to find out all possible combinations, which can exist and for which the blocking happens and total number of possible combinations. So, it is very simple. I will assume that, there are… m and n I have been taking; m and n – this is fine. So, there are n links here, m links here. So, take one particular combination – some particular value of x fixed. So, this pattern is fixed now.

Now, this pattern can vary independently. In how many possible ways you can create two busy links?

Student: ((Refer Time: 24:01))

4 C 2; there are only these two guys, which are there; which I have actually occupied. Or, in fact, these are two actually now free ones. In how many possible ways these two free ones will coincide with these busy ones? 2 C 2 – that is the only possible way out of these two. But, these free has to coincide with this. In how many possible ways you can do this? 2 C 2 only ways. And total number of possible ways with which these free links can be arranged here is 4 C 2 again; does not matter whether these are busy links or free links. So, these are total number of ways in which these two free links are coinciding with this divided by this value will give you the call loss probability approximately if everything can become busy or free with equal probability.

I am still assuming the same thing – equal probability; that is a uniform loading condition. What would it for Lee's approximation? So, this divided by 4 C 2 will give you the loss probability given 2, 2. I can take up another example actually here to exemplify this whole thing. So, this time let me take six. So, x is equal to 4 and y is equal to – maybe I can take it say 3; that is fine; I am just taking any arbitrary value. So, I have freezed up… And remember if whatever is true for these four busy and these two free links, there are many other combinations. This ratio will still remain same irrespective of whatever I take here. So, I need not count the number of ways in which this arrangement can be done. I am only worried about the outgoing side. Or, you can do it reverse way round. You fix this one and make an estimate based on this; you will still get back the same result both ways. So, in how many ways these three busy links or three free links, which are there? These three free links can be arranged in this case.

Student: ((Refer Time: 27:05))

No, there are total six possibilities; and these three have to be arranged – 6 C 3. So, 6 C 3 is the total number of possibilities. And these three free links can be arranged over these four busy links in how many ways? 4 C 3. So, this should be the call loss probability.

Student: ((Refer Time: 27:40))

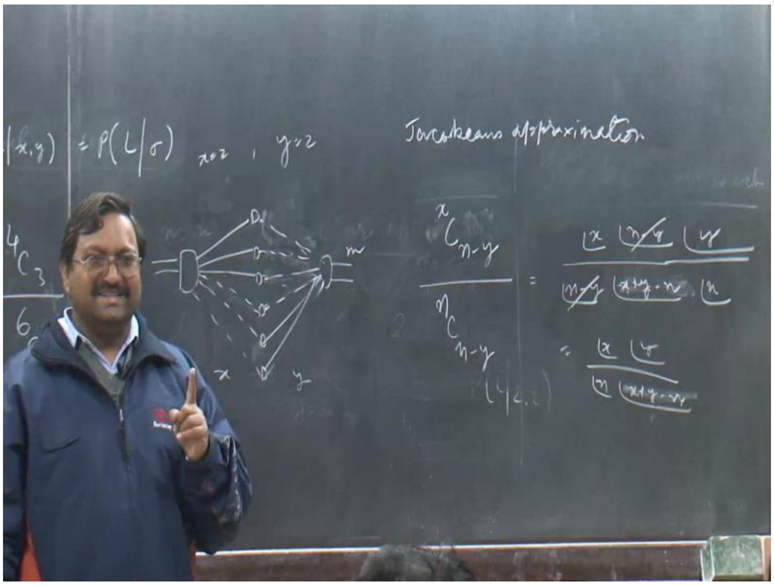
Which one?

Student: ((Refer Time: 27:46)) In case of input is coming… ((Refer Slide Time: 27:53))

You have to tell me. No, there can be blocked; why not; there is a possibility. In this case, which I have shown – it is not blocking. Let us draw the case like this. There are only three free links. Now, can you set up the connection? So, there are certain combinations for which blocking happen. And if you observe, this actually means x plus y has to be greater than or equal to n; then only you will have blocking. If x plus y is less than n, then you cannot have blocking; it is not possible, because even in worst case offset scenario, you will have certain free links. So, here total number of… What is the value of n? Is 6. Here it is 4; here it is 3.

You make it actually say 1 or make it 3 and 2. So, 3 will get occupied; 2 will get occupied. You take whatever combination; there is always one free link available – one free path, which is available. Call can be always set up. So, 3, 2 case you take for example. I am just modifying this. Now, there only two busy links; in worst case scenario, two are here; there is no overlap; there is still one free path available; blocking is not possible in this case. So, blocking – this is the condition. In fact, you would not be able to estimate P L given x, y unless this condition is met. We will come to this. Our factorial will not give a solution.

(Refer Slide Time: 30:09)



So, now, generalizing this whole thing. So, I will not be using now numbers; this was an example. So, let us put this as x and this as y. So, in how many possible ways you fix up one particular combination. For x connections, I have already freezed. For y, how many n minus y – those many links are free – big type links. I have actually freezed through which the connection can be set up. In how many possible ways you can arrange these? n C n minus y. And in how many possible these free links will be overlapping with the busy links on the input side or A type links? So, that will be x C n minus y; same thing, which we did earlier. This is x C n minus y; this is n C n minus y; same logic. Agreed?

You expand on this. So, this is the numerator. And let me expand now on the denominator part. This is what it will be. This cancels and you note and observe this particular factorial. This is only valid when x plus y is greater than or equal to n. For anything smaller, you cannot define this. You do not have factorials for negative values. This is only for positive values you define. And 0 factorial is always 1; but, that is the condition. And then, you ((Refer Time: 32:39)) define the factorial. So, that condition has to be made whatever was written here. So, this clear?

Student: ((Refer Time: 32:53))

Just a minute; just a minute; just a minute; you are right, because when I do x, minus n plus y. So, it has to be… Thanks for correcting; it has to be this way. So, everybody agrees with this. So, we have already got this thing.

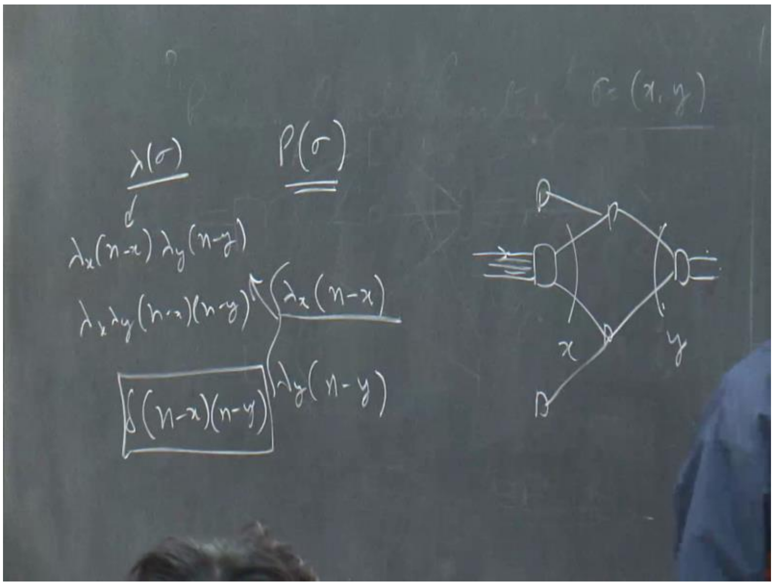
Next, we need to get what is the probability of being in a state sigma.

Student: Sir, that n minus y ((Refer Time: 33:40))

n minus y is the free links. So, these are the… How these free links, which are there; in how many possible ways they will overlap with the busy links on the A side? So, A links I have taken the busy ones, which is number is x. So, x C n minus y possible ways the overlaps will be here. And what are the total possible combinations? n C n minus y. So, that ratio will give this, because I am using random route hunting remember. If it is not random route hunting, this will not be correct expression; you have to change that.

So, how you are setting up route is important. So, you cannot simply blindly use this formulation if the route hunting is different, if the call has been set up through a different hunting process. Hunting is… You will keep on observing, keep on finding out when the free path is available. Here you do it randomly. So, with the random route hunting, this is okay. That is where the equal probabilities have to come into picture. So, I can just do the enumeration. And ratios of that can be used to estimate the probability; which cannot be done otherwise. So, next is P sigma. You will remember this; I have written it in the corner. So, I will keep it here.

(Refer Slide Time: 35:14)

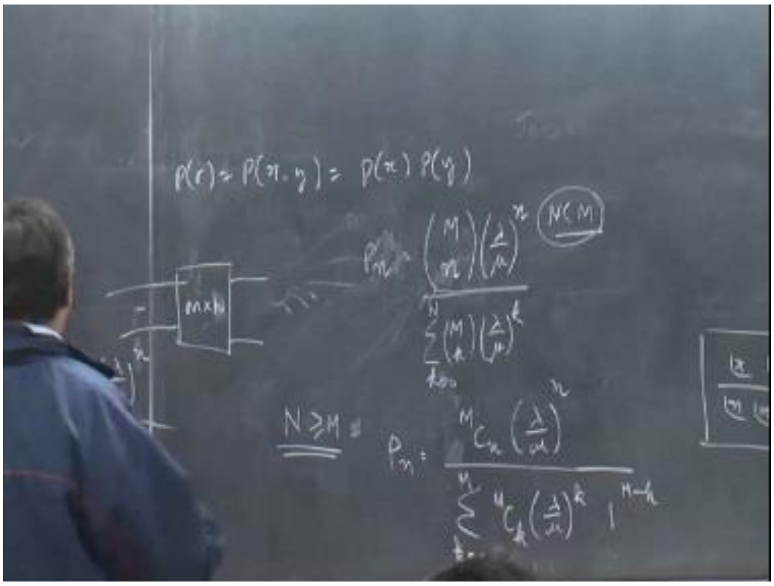


So, I have to now find out probability of being in a state sigma. So, one of the important thing is that, in this switch, the value of x here and value of y – both are independently driven; they are not dependent on each other, because some other people will be setting up the connections and the y links are busy only because of other nodes. And these links are busy because this guy is trying to set up to anyone of the outputs. So, x and y can be independent.

So, I can safely assume that, for each line, I am getting lambda x is the arrival rate; I am going to use whatever we did for composite switch – same thing. And of course, if there are x lines, which are already occupied; so lambda x into n minus x, because each line I am looking at now the arrival probability. So, I am not worried about… This actually cancels out whatever it is the value. So, actually what is the occupancy probability of a link? I am bothered about that; m does not matter, because again I can do balancing and then find out what is the probability of occupancy of each line here. So, call arrival rate is proportional to lambda x into n minus x for this particular links.

On the other side – on this particular side, y – I can again take as if the call is coming from the output side. Actually, it is not… It is coming from these other switches and causing these two be busy. But, this is symmetric thing; the switch is symmetric. So, I can very well assume that, they should be where I will call arrival probability from the other side. So, this is not the probability P sigma; this is lambda; not p sigma, but this is lambda sigma, which I am estimating. So, this can be represented as… I will come to this probability later on. Lambda x lambda y n minus x n minus y. So, these two I have used here. And I can write this thing as delta n minus x, n minus y. So, these are second, which we will be using.

(Refer Slide Time: 38:12)



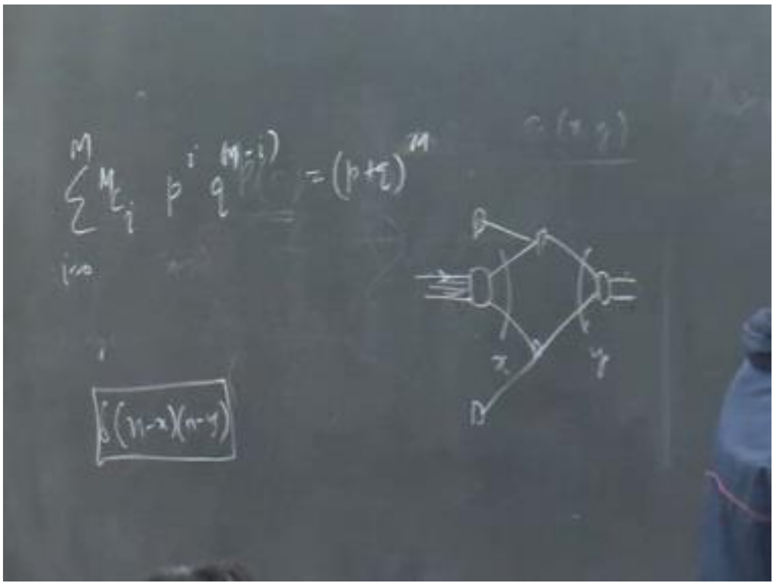
Now, coming to probability of being in sigma state; so take a composite switch M by N. And so P of being in sigma is nothing but P of x comma y. I have already mentioned that, these are independent. So, I can always write x and y; you can be independently controlling their probabilities of being in those states. So, what is the value of P of x. Remember the composite switch thing; we made Markov chain; and there was a state variable thing, which was done, and we got the state probabilities. So, we will use that and that value will be…

Student: ((Refer Time: 39:08))

Right, we will use n set probability distribution. So, probability of – you are going to be in a state x can be written as… Now, when N is less than M in that case. Probability that you are in state n was represented by… This was for a switch of M by capital N; this was the n set probability distribution. This is the condition, which is satisfied. This is the probability being in state small n, where n is number of outgoing links, which are busy; that is the estimate, which was there.

Now, when N is greater than or equal to M; in that case, summation will be… This M k will not be required; this can be replaced by… This capital N can be replaced by M; where, N is greater than or equal M; you can only have – go till M state; you cannot go to M plus first state because all the incoming links will be busy. So, when this condition is satisfied, you will have P of n. So, I will write with C actually. Summation of k goes from 0 to M; the state cannot be higher than this in Markov chain. So, this will be now M C k lambda by mu k. Now, this is nothing but a complete binomial thing; whole range is there now. Whatever I am using the factorial; I am also having those many terms. So, I can convert it to a close form solution. So, close form solution will come from…

(Refer Slide Time: 41:57)

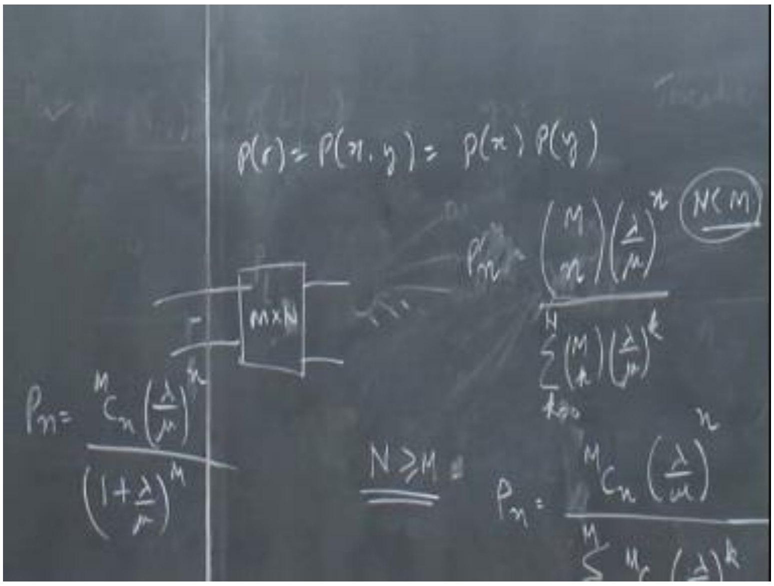


So, I think all of you are aware of M C i summation of p raise power i q m minus i; where, i goes from 0 to M. And this is nothing but p plus q raise power M.

Student: ((Refer Time: 42:19))

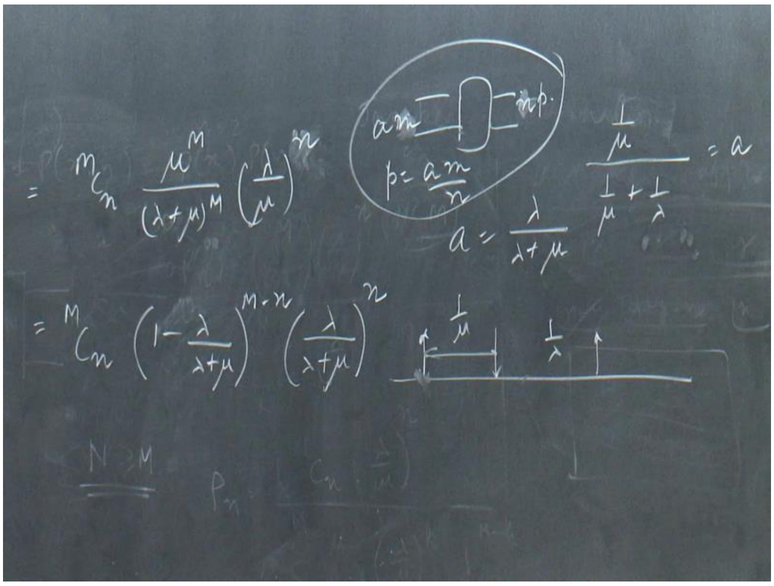
This is M minis I; right.

(Refer Slide Time: 42:30)



So, I will just use simply the same thing here and you will have P of n will be given by M C n lambda by mu raise power n divided by… Now, what is the other variable? This P raise power k. So, I can safely write this thing as 1 M minis k. So, this will be 1 plus lambda plus lambda by mu raised to power M.

(Refer Slide Time: 43:10)



And, I can further modify it. You will define load actually for the telephone switches. So, I can write this as m C n mu raise power M lambda plus mu raise power M; I have actually solved this particular thing; expanded. Basically, mu will come here – mu by mu. And expanding, I will get this. So, this can be now rewritten as…

Student: Sir, lambda power ((Refer Time: 43:55))

This is n; right; this thing. So, we will define some – a factor called lambda. And why I have done this? Because on an average, a telephone line gets occupied for… or any circuits line for 1 by mu duration; and this is the duration. Once the call is finished, the average time after which the new call will arrive, this 1 by lambda is that time. So, on time scale, the first call arrived; this was the time till the circuit actually was set up. So, this is 1 over mu. So, I will call it call arrival. Then, there is a blank period and new call arrives. This is 1 over lambda. So, this is nothing but the fraction of time for which a line will be occupied on an average. Or, we also call it occupancy probability. Then, I will define this thing by a. This is a is same factor, which we used in the balanced condition for doing the Lee's approximation.

So, there it was… I defined that a into m is equal to whatever is the lambda p – n into p. And then, from here we got p is equal to a into m by n. So, this was used in Lee's approximation. This a is actually same a, because load cannot be identified in case of telephone circuits, because in packets switching system, we assume that, packets are atomic and quickly they come at an instant. So, how many of them are coming? So, lambda and mu; we decide based on that basis. And for mu, that departure rate is such that, we actually assume that packet is going as one instant. But, two packets cannot go consecutively; there is a gap. Gap is because of the packet length of course; but, it is not holding up the circuit. In this case, I cannot do it, because there is nothing else can arrive till the time you are busy.

In packet switching system, in the input of the queue, a packet arrives as one single atomic entity. At the next instant, at t if you get a packet, at t plus also you can get a packet. And circuit switching – you cannot get a call till the call itself is over. So, here loading condition is different; we call it a. a usually will be represented in terms of erlang. Erlang is fractional utilization of a line. So, only for one line, if it is 0.95, you say 0.95 erlang load on one single line. If a switch is of say 1000 incoming ports, every line fractional utilization is 0.95. So, 0.95 into 1000 – 950 erlangs will be the input load. So, we define it that way. So, expanding it further, we will have… This same thing can be expanded into this form; you can verify this. So, now, you can represent this by a and this by a, and I will get an expression. So, we will actually in the next class, we will go – move ahead with this a thing and we will now put all the three things together into a summation and solve it to get the call loss probability. So, that will be our next step.