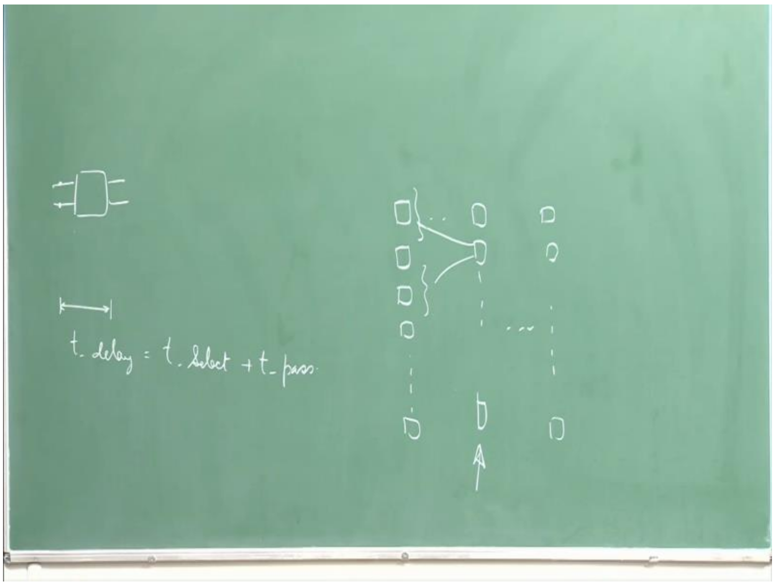
Digital Switching
Prof. Y. N. Singh
Department of Electrical Engineering
Indian Institute of Technology, Kanpur

Lecture – 22

We will start from where we left in the previous lecture. I will now give how you actually, iterate these particular specific scenario of how, you solve a Markov chain. Technically, it is actually, solution of a Markov chain by doing iterative computation actually, because sometimes when, these numbers of states are extremely large, it is very difficult to get a solution. Usually, we would always, teach the solution is we always, say; let us build up a Markov chain, where we have some state probabilities. Based on that then you start building up what you call balance equations, and then using a fundamentalism; that all state probabilities sum has to be equal to 1. You actually, get all simultaneous equations, solved them to get state probabilities; that is what usually, we do. Here actually, I am not interested in that. I am more interested in what is going to be my throughput performance. What is going to be the outflow, but that actually depends on the states.

(Refer Slide Time: 01:36)

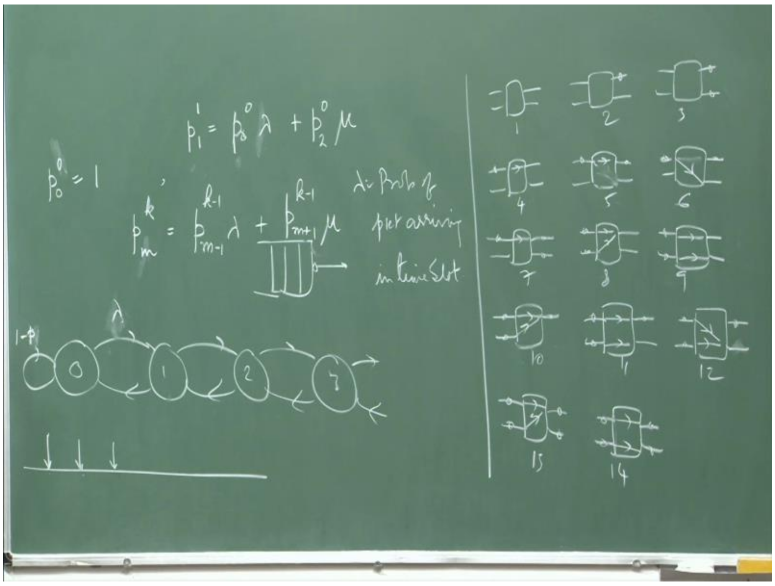


Last time, I had drawn 14 states of a 2 by 2 switch, and we will take two modes. One case will be when in fact, for every switch, when the packets are there at the input, there are two ways I can actually understand. Two ways of analysis can be done. One is the total period, which is required in this case is what we call just a minute; yes, t delay actually. So, t delay is one step; delay from here to here. This delay usually, will consist of t select plus t pass; t select is a time required to identify, which outgoing port, these packets have to go, and then when actually, the transform happens, that requires this much time. So, you will take actually, first the case when t pass will be; t select will be 0 and then t pass will be 0; there are two cases, actually.

Now, one important thing I have given you 14 states. I will assume that because now switch will actually, consist of buffer delta, consist of many switches in state, and there are many states actually, and so on. So, all states probabilities, within the stage are going to be same for all switches. All state probabilities for all the switches in one single state is going to be same; that is the assumption. So, I need not bother about all these switches, because otherwise you see 14 states I have given, and then you have how many switches?

So, 14 possibilities raise to power number of switches; those many number of states will be there, and even, if it is a very small number, say you are going to have 8 by 8 switch, for example. 8 by 8 switch requires 12 such switches. So, 14 raise to power 12 is a huge number, actually, and mentoring those kinds of things in a Markov chain or in any finite state machine, is going to be complicated. You cannot handle that actually. So, analysis cannot be done in that way. First simplification, which was done is this; this is usually, true, because number of ports, which are coming in, these are independent sets, and that is true for all switches, because what is true for first switch; resultant for the second, and so on, iteratively. All switches technically, see the same thing. So, their state probability has to be actually, same. Now, this is going to be true. So, I am going to take this case. This is the one important thing; you have to keep your state diagram handy; all states, actually.

(Refer Slide Time: 04:50)



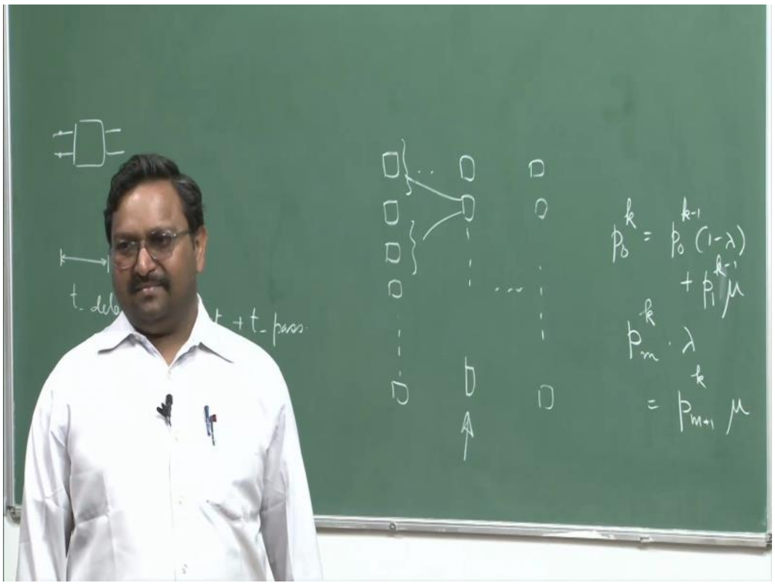
I will just, still put them here; I will not erase them, during the lecture, because I have to refer, again and again, back to these states. So, I think I can redraw them. This is one. Now, before I go onward with the iterative method of computing the state probabilities, method is going to be pretty simple. For example, if you want to find out for a queue, for a single queue, I think I can just, explain how it will be done for that. For that actually, usually, I can find out every simple solution, because my balanced equations can be formed. So, with this, if I actually, being given you 0 transition probabilities, 1 transition probabilities, and so on; for example, this is an infinite queue.

Usually, actually, you can always run an algorithm on this. So, you can start with a queue when it is empty, and now, we talk about state probabilities at various time instances; that is very important. While, when we did a queuing analysis of a simple queue, I was not using time instance for state probabilities. To a t is equal to 0, I will always define p 0 as the state at time t. So, this is k. So, k will be 0 in the beginning. Your initial boundary condition can be that there is no queue; that is the probability. Then, you will have transition rates. So, you have to define transition probabilities. Lambda is a rate for going to one direction, and mu is not going to that; mu is coming back.

So, you will define this as lambda, and not going to go there, is 1 minus lambda, actually. Important thing is transition is happening at every time instant; it is a differential. I am not making a different kind of queue. Here, at on a time scale, at t is equal to 0; packet will come; t is equal to 1; something will happen; t is equal to 2; this will keep on happening. It is not as such a Markovian thing. Lambda, I can replace this thing by actually, p; the probability p; it is a discreet time system now. So, with probability p, the packet comes; 1 minus p, packet does not come. With packet, again, the same thing is going to happen at every time instant. So, from here actually, you can find out what is going to be your p 1. From here, you can actually, find out whatever, was the value, p 1 f stand 1. In fact, you can build up an equation based on this. For one, I will show. Then, I can write it for general. I will use a very similar procedure here. This one will be that you have to be in this state in the previous time instant. Then, there is a transition probability, which was I call it something else, I think; let us call it lambda. Lambda is not arrived at; it is a probability. You have to understand it. So, lambda is probability, but this is a computational procedure; you cannot have a close form solution here, with this, you only able to come here.

Second thing, which you can get is p of 2. From here also you can come, and this probability, you have defined as mu; mu is a probability that a packet will go out of the packet, actually, out within that slot. It still requires the same slot. It is a technically, is not arrived at; it is a probability. It is a conditional probability, technically. So, if you are in this state, chances that you go here is this, but it also turns out to be same as rate. So, this method is not used, while doing for a simple m n 1 infinity queue. So, I have modified the queue somehow. If you get one equation through iteration; remember, my time instances are different here, and to begin with my initial condition is that this is one rest; everything has to be 0, because of this. Rest, everything has to be 0, and queue is also in state 0. So, I will be able to get p 1 1 from here. I can, in fact, solve for all these in general. So, I can get p of say m, in state k as p of m minus 1, lambda k minus 1, plus p of m minus 1, m plus 1, actually, k minus 1 into mu. Now, you actually, can just run a computer program. It starts with this initial condition and as your time moves, you are computing every time, p 1 k, p 2 k, p 3 k, p 4 k; you are just iterating, and p 0 k will also, get adjusted correspondingly.

(Refer Slide Time: 12:25)



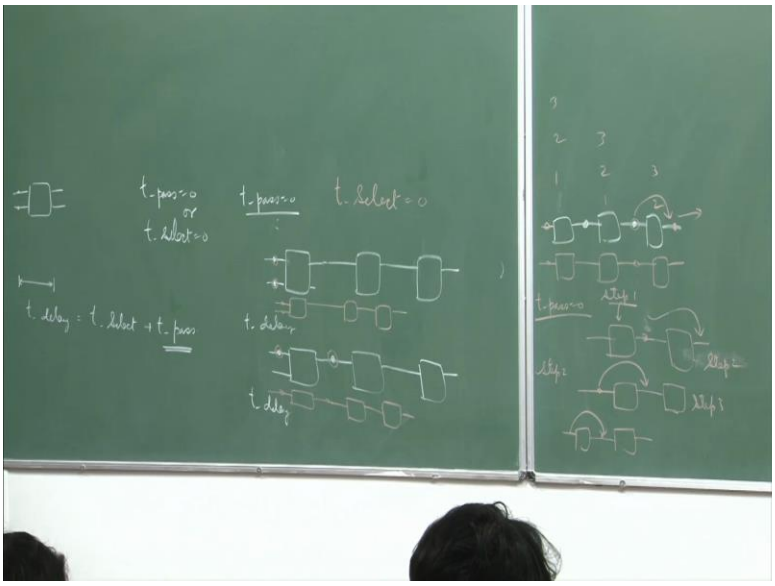
You have to understand p 0 k will also, be changing, will be a function of p 0 k minus 1, 1 minus lambda. So, what I am doing is we are adjusting continuously, and we are reducing it by some amount and increasing all other state probabilities. After some time in the computational procedure itself, it is always bound to converge. You will find there is a steady state condition, which will operate. A steady state condition is when the flow rate almost, becomes constant, out of the system. When for all of them, you will find out that it happens that p of any value, any m and say, k this almost, dot lambda; lambda is actually, constant here; departure rate. So, the flow rates here, statistically, have to become constant; they will not be varying, or when the flow rate becomes gets balanced; that also you can figure out. In this case, condition is pretty simple here.

Student: Sir, you have not we are trying probability of 0 early not transferring the probability from state 1 to 0?

Yes, that also need to be there; you are right. That also has to be there; you are right; I agree; k minus 1, that also you have to meet that. So, when this condition or the balanced condition actually, you see, start seeing your computation things have converged. Take those state probabilities, and you have got your results. With various values of lambda and mu actually, you can find out that thing. It will turn out to be almost, very same, what we did for the simple queue. Now, this same procedure we are going to invocate, because here we cannot get close form solution. This Markov chain was very simple, could have been solved, but that one we cannot do, but just to give an example, I have just simplified this thing into this particular. This is a computational procedure, but this comes in the category of still, analysis. It is not simulation, because you are not simulating anything. We are just computing, and it is an iterative computation, till the computation stabilizes or converges.

We are technically, solving balanced equations and balance equations are when the flow rates from in to out, and out to in; both are same. So, you are just finding out when the flow rate converges, and everything is going to keep on, it will actually, balance. Remember, it is like a first order equation; it is not a second order. So, there is no question of oscillations here. If you carefully, observe, it is a first order actual equation. It is always trying to move to a stable condition. So, convergence is guaranteed, actually. This is fine; this one was a simple queue. Now, I am going to come for a complex one. So, state transition probabilities are complicated. I think, one important thing is you have to understand how state transition probabilities are estimated, and how you actually, understand the whole extraction; that is the important thing. So far, the things were I think, little simpler. Now, coming to this; first case, which we will take is when t pass is 0.

(Refer Slide Time: 16:18)



There are two cases. I can take t pass is equal to 0, or I can take t select is equal to 0; this is or condition. Now, what is the meaning of this, actually? What is the implication of this? I think, once you understand this, then it will be much more clear. In the paper, it has been given later; I am doing it earlier. I somehow, feel this should be done. I have to take the cascades of the switch. Let us take this case. All switches are empty; there is no, nothing in the buffer, and they are having exactly, one buffer at every input port; does not matter. You can actually then do the simulations even, for two buffers or infinite buffers, whatever you will feel. So, if you have two packets coming in, now, you have to look at the time instance. This, at one particular time instance, it will come. Then, the t delay will elapse after this, and then the next state will come. This is how the state transition will happen, and then in this case, I am taking as t pass is equal to 0.

So, this t delay will be nothing but equal to t select; this is a time required for identifying to which outgoing port, the packet has to go, but packets move almost, instantaneously, from input to output, from any buffer to any buffer. So, once the decision will take this much time, you know there is a packet. At the end of this t delay, what will happen? One packet will be almost, instantaneously, just before my time period finishes, just before that; 0 minus t minus, actually. So, if it is 0 instant, then just 1 minus. At that stand, the packet would have moved here. Your situation will be something like this; if both where, I am assuming contending for the same port, you will come in this situation. Another t delay will happen. You will get a situation like this, now.

In this t delay, they will figure out, it has to go to which port; this has to go to this port; earlier also, it was tried, but there was a contention. So, it was buffered. This will also be identified and just before the next time slots, just minus, just slightly, earlier, the packets will be moved, instantaneously. So, you will actually, have this packet, coming onto this position; this packet, coming onto this position. So, you will have this situation. So, that is when, your t pass is 0. I am drawing with a different color. When t select is 0, then what will happen? It actually, means when this particular period from here to here that slot, which is starting, within 0 time, I am able to figure out, which packet has to be going to the outgoing port. Then, it takes one full slot for the packet to be transferred to the outgoing buffer.

So, after this situation, this is what is going to be there, perfect. So, this is what will be the situation; another time slot delay. In the beginning of this particular slot, they will know, this packet has to go here, and this packet has to go here, but the problem is this packet cannot move, because this buffer is not empty. Because in the beginning of the slot itself, we have identified where the packet has to go, but when you start transferring, this packet will take full one slot for transmission to the outgoing buffer. Unless this buffer is empty, this cannot move. So, this will still remain strucked there. So, you will end up with a situation here where, this packet will be here, and this packet will be here. So, that is the difference between the two. So, analysis, when t select is equal to 0, is simpler, actually. So, I will do a complicated version first, when t pass is 0.

Student: Sir, final state will be until, there will be no packet in the middle stage?

The one with the red ones you look at or maybe, I can draw it, separately, if you wish. So, colors, you have to match. So, red one corresponds to red one; white one corresponds to white.

Now, for when t pass is 0, you have to understand now, if there are packets, which are queued up like this in a string, all of the packets will move in the same slot to the outgoing port. Decision will be taken at the same time; one slot period. Then, there is an instantaneous movement. So, all of them will move in a chain. Now, solving this is slightly, tricky. What happens is each time slot or each event is being now, identified into three phases. So, first phase, we call when the packet actually, goes out. When this is the outgoing buffer, when the packet from your outgoing buffer or input of the next stage, is pushed out; that is a phase 1. When the phase 1 is happening; this is step 1, actually; we do not call it phase 1. This is a step 1 here. Step 2 will be when the packet from here to here will come. I have broken things into steps, so that when this goes out, I can immediately, move the packet in the same slot here, which is not possible, when t select is 0. For t pass simulation or iteration, I need to do this. So, that is the step 2, and step 3 is that packet comes to your input buffer.

You have to understand when step 1 is running here; which step is running here? Step 3. When step 2 is running here; no, this is not step 3, this is step 2, actually; packet is going to from input to output of a buffer; this is step 2; this same thing is happening here for this. So, while at this same time, this step 3 is happening for this person.

Student: Sir, that is for t select is 0, these are?

These are for t pass; t select is equal to 0. You do not require this definition. This breakup is not required per slot. May all if you carefully think, I think this is difficult to explain, but this is the way it happens; this is only way, it can happen, actually.

Student: Sir, once again.

Student: Sir, more complicated is t select is equal to 0?

More complicated is t pass is equal to 0.

Student: If t pass is equal to 0, everything is moving in a chain?

But then I have to split it, when I am doing iteration. It is a problem of how we implement the computation. Their computation is simple; every one step; one time slot, one step, but here, every time slot is broken into three steps. This actually, does allow all the chain to move in one go. So, when a time slot starts, you will say, step 1 here. Then, step 2; then step 3; so everywhere, you will do the step 1; step 1 will happen first; then step 2; then step 3 for whole switch. Usually, what will happen is this is the packet. At the outgoing buffer, step 1, this packet will go out instantaneously, here. When the step 1 happens here, then you will apply step 1 here, and at this point, step 2 is going to happen here; it is a chaining, which is which is happening. So, when 1 is happening here, 2 is going to happen here, and then when 1 is going to happen here, then 2 will happen here, and 3 will happen here. Then, of course, 2 will happen here, 3 will happen here, last 3 will happen here; one full chain of event. So, one complete slot will be simulated to 1, 2, 3; all three happened. For this, there is actually, nothing to do with 1 or 2; 1 has happened in the previous time slot. So, current time slot; I can implement this way; otherwise, it is not feasible. Now, coming to the formal definitions, which are required here.

Student: (( )) t pass equal to 0, we require minimum three stages?

t pass is equal to 0, three steps per slot.

Student: Any three stages or sir?

Any number of stages; there can be just three.

Student: Minimum, three stages.

Not necessary. Even with two also, you can do. It is like a state machine. One full slot gets done, when all three steps are executed.

Student: What happens in the second step?

Second step, packet is coming from your input to your output. First step, packet from your output has to go.

Student: Output has to go outside?

From your output, the packet is moving further; going out; it passed. From your input to output, then packet has to come. Then, the packet has to come to your input. These three things will give a slot for a switch.

Student: Which is my input (( ))?

Student: Any particular switch.

Which one?

Student: In step 2, and then what is the?

I am talking about this switch. When for this switch, this step 1 is happening, this packet is moving out. For this switch, step 2 is happening at that time.

Student: Sir, which is output in step 2 and 3?

Which one? This step 2 is for this.

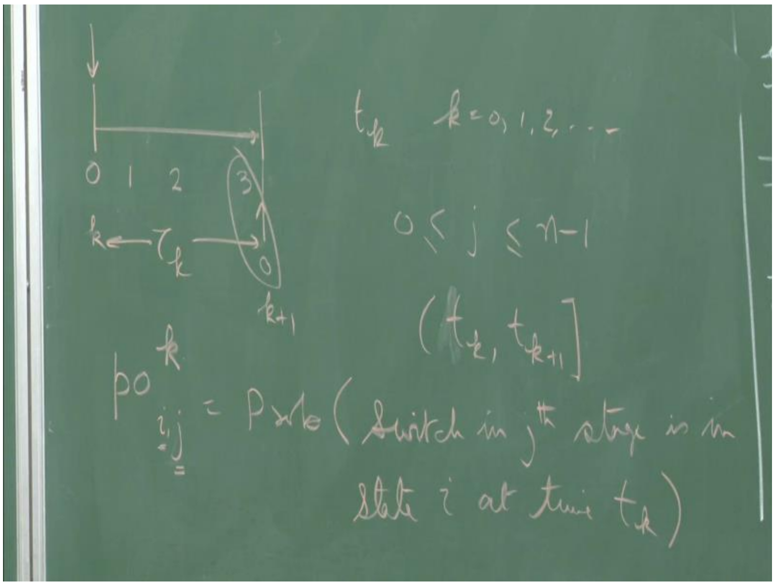
Student: Yet, a packet is going (( )).

Packet is going from input to output, but then for this switch, the packet is coming to the input. So, there is a step 3 for this, and step 2 for this. Last one packet is coming to my input. So, the previous switch, you are having a?

Student: Two.

Step 2 happening. So, it is always 1, 2, 3, and it actually, propagates backward; that is what is happening. So, when full 1, 2, 3 set propagates from output to input, one slot has finished. That is the way we assume; t delay is one delay; t delay is delay here, remember, and every t, after t delay, packet moves further. So, one t delay requires three instances, three steps. Selection time is full t delay in this case, but packet passing is instantaneous; packet transfer. So, we defined now what we call about probabilities, before we come to these states. So, notation is very simple.

(Refer Slide Time: 28:46)



What happens at time, when slot starts, you are at a state; you are at step 0. Then, step 1 will happen; then step 2 will happen; step 3 will happen. Step 3 will be nothing but step 0 for k plus first slot. So, step 3 for k th slot and step 0 for k plus 1, are same. Now, we will actually, define a time instant, t k. Now, k can go from 0, 1, 2, and so on, because we say, iterative calculation. These are the stages in the switch. So, I am not worried about how many input ports are there; 2 raise to power n by 2 raise to power n, but number of stages matter, because I have made an assumption, which is a correct assumption for almost all delta networks; that in a stage, all switches are equivalent. Their state probabilities are going to be same. So far, they are in the same stage under that assumption. We will define now, a probability, we call it p 0; this is after step 0. Actually, in time instant, in time slot k, k starts from; that is a slot and I think here; do you understand, what is the meaning of this? Closed and opened intervals? In this case, t k is included; t k plus 1 is not included; that is what it means. It is opened on this side and closed on this side.

So, this particular probability is a probability that your switch in j th stage; this is this. It is in state; this is not stage; this is state, I, this one. So, notation, you have to remember. Do not change it; it has to be consistent, actually; at time t k. So, if this is a t k instant when the time starts, this is that value. Now, this whole interval is known as tau k; tau k is in this interval; this side is not included, this is included.

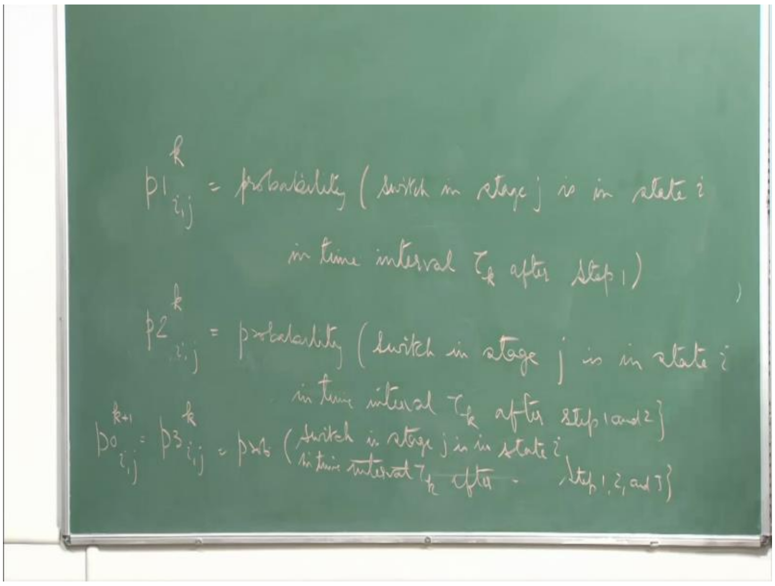
Student: Sir, this tau k, I am assuming that stage 1, 2, 3, will?

There, states are there; steps are there; steps 1, 2, 3, are happening. I am not going to write down those probabilities; p 0, then p 1, p 2 and p 3; p 3 will be nothing but becomes p 0 for the next time step; next delay.

Student: Time t k the switch is in state I?

Switch is in j th stage at state I; there many stages. So, for multiple stages, you have to solve.

(Refer Slide Time: 32:21)



I am just giving the formalism; actual calculation has to be done through a computer; p 1. So, this number identifies the step after which, you are looking at the probability; state, stage, time instant; time slot; and this is pretty common in most of the switching analysis. You will use lot of subscripts, superscript; this all kind of notations. This is a probability that your switch in stage j, is in state I, in time interval; I am not writing time instant, now; time interval tau k after step 1. There is no step 0; third step is step 0 for the next time.

Student: That notation p q 1, then?

Which one?

Student: In this p 0?

This is after 0 th step; just before the first step.

Student: Just before the (( ))?

When first step is taken, second has happened; third has happened. Third is nothing but 0 th of the next one; next time slot. So, I will change there; k will be changed there.

Student: But it counts 0 only, three steps, but counting 0 only?

I am counting 1, 2, 3.

Student: 0 is same as 3?

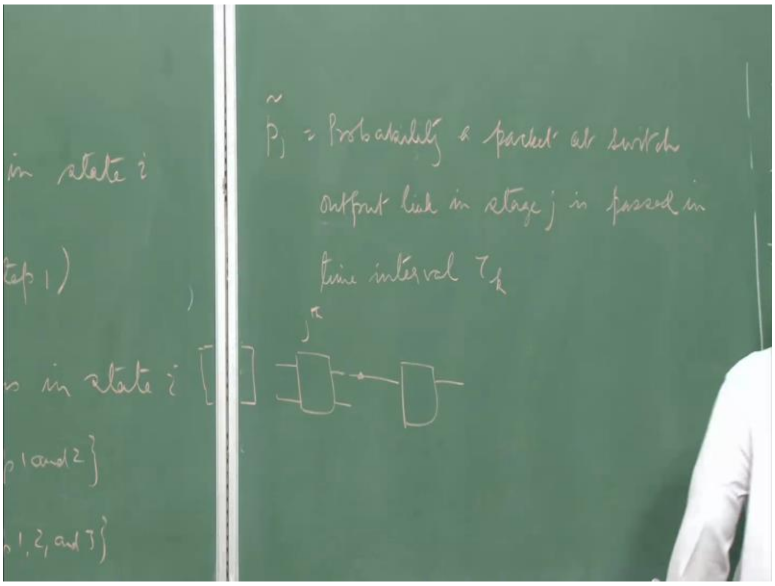
0 is same as 3, but time instance changes; k will become k plus 1. Why the steps, I would not count 0, 1, 2, 3; is the objects, which are counted or ports, which are counted 0, 1, 2, 3. Now, I am being consistent with the paper, because ultimately, I know this is a difficult to communicate in the class. You will anyway, have to depend on the paper, and if my notations are not same, there is going to be a problem in that case in understanding. So, I have to be exactly, same as what is given in the paper. Certain nuances, which have not been explained; for example, t pass and t select difference has not been explicitly, told. The procedure of computation is not explicitly, stated, because the author assumes that what happens in most of the paper; we do not give details sometimes. We assume these are kind of a common knowledge, and person can figure out on the other side, but I know the class cannot figure out, as of now. Unless, you are a PHD student with a good amount of experience, is going to be a difficult to figure it out. I could do that but that does not mean that everybody else can do.

I have to be explicit, and at least, you have an option to fallback. You can always ask me, if there is problem. I never had that option. With this paper is pretty old, I do not know where are these guys, but I think this was the first one, which k buffer, buffer delta analysis part, and it matches with the simulation; that was good thing. This has taught me a different method of solving markov chains; iterative computation, which is again usually, not taught at most of the places. So, switch is in switching stage j, is in state I, in time interval tau k, after step 1 and 2, both has happened. Similarly, you will have p 3 k and this has to be equal to p 3 I j k plus 1. Rest, everything follow exactly, except in the last you will write, step 1, 2 and 3; rest, everything is same.

Student: Next time interval will be tau k plus one, sir.

Yes, next will be tau k plus 1. Let me write it. I think this is better, because switch in stage j, is in state I, in time interval, after step 1, 2 and 3, and we will define one more probability actually. Now, come the trickiest part. Now, what I am defining is actually, will be the probabilities, which are used for computing what we called transition probabilities at every state, and these will also keep on changing with time, as time evolves, and they will stabilize after some time in the Markov chain.

(Refer Slide Time: 38:31)



First probability is p j tilde. I am going to use three of them. I think one has to be careful; I will be using this; I will be using this; these three have all different meanings. So, first, I will define the top most one. This is the probability that a packet at switch output, link this stage j; this j actually, identifies this; is passed in time interval tau k. So, actually, it is indeed a function of k; I am not explicitly, mentioning it, but all these p j tilde, p j bar and p j will be function of k; I will be not writing them; paper also, has not written it. So, I am keeping it as it is. Ideally, I should have put a k; k superscript there, for doing this.

This actually, means this will be in terms of now; the states; whatever those 14 states are there, and I have already defined how the probability will be defined, after every step. We know what these probabilities actually, mean? What happens in step 1, what happens in step 2, and what happens in step 3? From here, if there are two stages, if this is a j th stage; what is a probability that a packet will go out? I am trying to estimate that. Now, usually, what will be that function? This will depend on at the start of the time slot, what was the state of this switch, and then the step 1 will be happening for this, and step 1 for this, and step 2 will be happening for this. There is a switch previous to this. That also, you have to take into account, because their index j actually, will change, when I am going to estimate for p j. I will be using in this case, j plus 1, actually, and that decision will happen is what happens after step 1 here? Because step 2 outcome will be decide by what has happened in the step 1 here. That is why the expression will actually, contain all j plus first terms, but what is going to happen after step 1; let us see what is going to happen.

Student: At expression tau k is passed interval three q.

Time interval tau k.

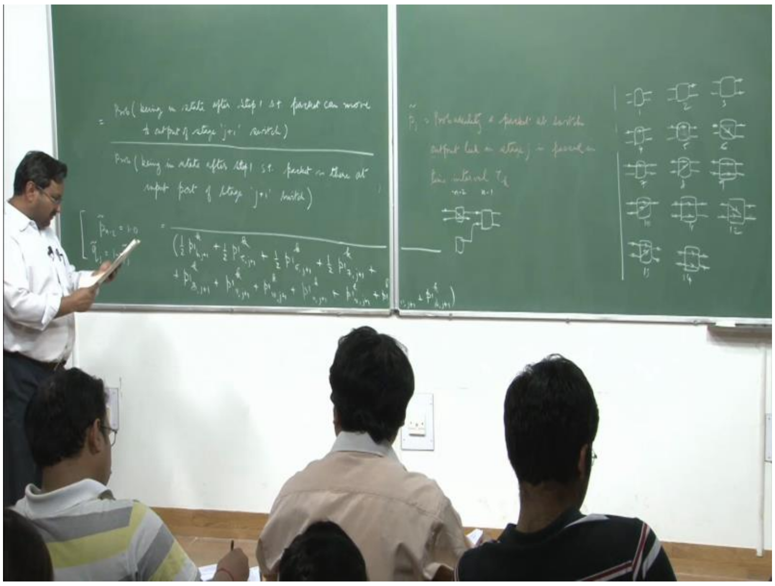
Student: No, is passed, second line?

Second line, the probability that a packet at switch output link in stage j, packet is here, is passed in time interval tau k, and passed to the output. Then, it will never pass anywhere else, can only move forward. So, it is being moved in time interval tau k; that is what it means.

Student: If transition function wants to skip two, sir?

Step 1, 2, 3; we have to identify. Currently, I am only looking at if there is a packet sitting in here, what are the chances it will move; it is a conditional probability, remember, it is a conditional probability that packet is sitting here; what are the chances that it will move here? You have to identify basically, the states for this. What are the earlier states, and after step 1, what is happening here; that will decide the probability of transition for this. Whatever happens here, we will have an impact here. If packet moves here, then only, state will change here. If packet does not move, state does not change here. This actually, will be used for computing transition probability here; this is a outgoing packet probability, but that will be function of what happens in step 1 here, after step 1; if the packet goes out or does not go out. If packet does not move out from here, then nothing will happen. This packet cannot move. So, essentially, what I have to do is I have to list.

(Refer Slide Time: 43:27)



This thing can be written as; this will be nothing but probability of this, such that I state. I have to essentially, identify states, when this is possible, and remember, this is a conditional probability that I have to identify all these states, when the packet is there at the input of j plus first stage. This will become clearer, when I will give the example now; build up all that. Now, I will give the example of how this actually, works out. Of course, for this, the boundary condition before that is this. At every stage, you will identify q of j by this. These two are; why these two? In the last stage, if the switch is there, there is one buffer here. Packets are instantaneously, removed; there is no packet queued up here; there is no buffer required at the output. If these packets are there in the last stage, and they both have to go to any port, even, they have to go to the same port, there is only time required is t delay, which is for selection. Selection is done, when both of them will be instantaneously, put here, and they will be instantaneously, taken out. So, probability that this particular of packet will be, which is there in; this is n minus first stage; remember, this is n minus 2.

So, probability that packet at the output of n minus two will be take passed out, has to be always equal to 1, if the packet exists condition on that. So, that is the initial condition. It is known as boundary condition of this solution. You have to start with certain boundary condition, always. Initial condition will be when all switches will be in this state; state 1; no packet in the system, when you start your iteration.

Student: Sir, boundary condition p tilde n minus 2 to one point; this is there is no output contention thirty eight minus three?

Not possible, because packet is taken out instantaneously; t select is equal to 0. When t select is equal to something is equal to 0, then this will be, problem will be there; you cannot take. Then, it is p n minus 1 tilde is equal to 1; it is not p n minus (( )).

Student: If t pass is equal to 0, t select is what we are modifying t select is equal to t tilde.

In full slot, both of them will be selecting one output.

Student: Sir, if there is output contention?

Does not matter, because time t is 0 for reading out the packet. So, both of them can be read out instantaneously, and they can be read out from the output port, is 0 (( )) case.

Student: Sir, only consider in packets set to the input that is why I am taking only step one?

Packets set the input; I am bothered about the j plus first stage; what is happening here? If I am in some state; I have some packets queued up; what happens to my outgoing packet; that will decide what will be my next state, whether packet can move or not move; I am looking into transition probabilities, now. Whatever is my current state here, this can only get modified, if my packet moves, and that depends on what is the state probability after step 1 here. That is why, this is done this way.

Student: If there is a contention, in that case there would not be any delay, sir. Both the packet will be (( )) simultaneously. In case, there is a contention?

Only at the last port; only at the last stage; not here. What happens, for example, if you put packets here, if these two packets do not go out, I am sure, both of these want to come here. So, you have consumed the full pair and finding out they have to come here, but this buffer is not free. If buffer is not free, they will remain there. Remember, the patrinet model of the switch; there is backward flow of signal, which is there. Packet can only move if the buffer is empty in the next, at the input of the next switch.

Student: Here, we do not consider packet dropping?

There is no; packets are not dropped; they are all buffered; is a full loading condition. Remember, the initial assumption is there. There is a maximum loading condition; I am only estimating what is the maximum achievable throughput, but I am actually, using a trick. I can actually, modify the input probability, but the input probability, initial conditions or boundary condition; I am taking this as a maximum loading condition. I will write actually, more; this is not only the expression; I have to write p j bar also.

Now, this will be, I have to first of all, write down all the states, when the packets are there, at the input. Then only, they can go out. If the packets are not there at this port, they cannot go out. I am worried about here. So, I have to look into all the states in j plus first stage, with the packet is there at the input. So, first one; can I put 1, I cannot; there is no packet at the input. So, 1, 2, 3, cannot be used; 4 can be used; there is only one packet. If the next stage switch is in 4, there is a chance I might be connected to the one where, there the input buffer is not occupied, or I might be connected to the top one where, the buffer is occupied; both of these can be connected to my switch in j th stage with equal probability. So, I will use half of this, p 1, 4, j plus 1; I have to write all that; time instance k. Half is because I can be equally, likely, connected to anyone of those, if the switch in the next stage in this state.

So, my possibility is half 50 percent of time, I will be connected to the one where, input buffer is occupied; 50 percent time, buffer is not occupied. Even, if I flip these input ports, this I can take on top; this on bottom; it is still, the same state, remember, plus what is the next one? You have 5 also in similar situation, but remember, in 5, the next stage in that state; the packet will not move. Packet cannot go, because the next buffer is not empty, after step one. This is after step one, actually. Step 1 is already, taken place in j plus first; the outgoing packet has not moved out. So, my packet also, cannot move in.

So, 4 will be sitting in the numerator, but 5 will not be sitting in the numerator; it is only in the denominator for this conditional probability. So, I will write p 1 5. I am intentionally writing it, first thing here, and then I will only take some of them on the above part; all of them will not go; p 1, same as with the sixth; there is only, one input port, I can have connected to anyone of these, but six packets will go out. So, it will go in the numerator also. Similarly, next one will be half p 1 7, j plus 1 k. I think you have to take a large copy. If you take an a 4 sheet print of the paper, that tilde and bar are not clearly, written on the paper, actually. So, if there is a confusion, then you ask me, because there, tilde will be actually, is visible as a bar in the paper.

Student: Sir, in sixth, why tilde half factor will come, because now the states are not?

No, but my stage, j th stage is going to connect to j plus 1, either here or here.

Student: Half factor?

50 probability, I can be connected here; 50 probability, I can be connected here. So, when I am connected here, input buffer is not occupied.

Student: Sir, eight should not come, step 8?

Step 8 will be half. No, step 8 will not be half; it will be full. This is the probability that packet is here, at the input port. Here, only 50 percent of time, packet will be there at my outgoing port, or the input port of the next stage. There, it will be sure that packet will be there in the port. So, p 7, and then of course yes, then after that I do not want to putting half here; it will be now, as you have correctly, identified; p 8. Similarly, you will have p 1, 9, p 1, 10 will be there? Yes, it will be there. I think almost, all of them will be there. 11 p 1, 12, p 1, 13, p 1, 14; I think we are going to close.

Only I will put the numerator part; remaining, we will do it in the next lecture. So, numerator part will be, you can quickly put. In case of 4, yes, packet is going to go out. So, I am going to write it positively here, when packet can move. Half of p 1, 6 j plus 1; packet can move in case of 6 also; it is free. 5; packet cannot move, actually; go out. So, that should not count to my probability by definition. So, half of p 1, 8, p 1, 9; I think both the packets can move out. So, it is a 100 percent probability; half of p 1, 11, half of p 1 of 12; this gives you the probability of p j tilde, in terms of the state probabilities, after step 1 in the next stage. So, I will use this p j tilde to essentially, compute. So, we will start from one edge and then compute all the way, in the forward and backward direction; two ways, we will do the computation, actually. So, I think here, I will close today, and then we will move forward. We will find out what is p j bar and so on, in the next lecture