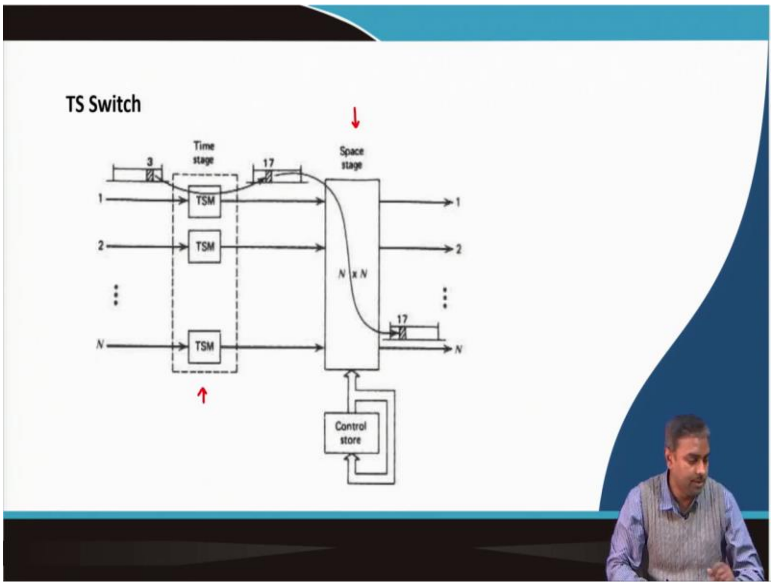
**Communication Networks**
**Prof. Goutam Das**
**G. S. Sanyal School of Telecommunication**
**Indian Institute of Technology, Kharagpur**

**Module - 03**
**Circuit Switched Networks**
**Lecture - 11**
**Space-Time Switch**

So, we are still discussing Circuit Switch Network, and we have discussed space switch time switch advantage of space and time switch what is the difference between them. So, we have in the last class; we started discussing whether maybe there is a combination that we can do so, which is the Space-Time Switch.

So, this particular class will be covering this combination of space-time switches, how they can be combined and what are the effective features of them, and how I design a space-time switch. So, that is something which we will be preoccupied with, ok.

(Refer to Slide Time: 01:01)




So, we have already talked about this time switch management or time switch interchange unit so, which is an essential part of time switching. So, now, what we are trying to depict over here in this diagram is a time switch space switch, so, basically first portion of the switch, which is this part.

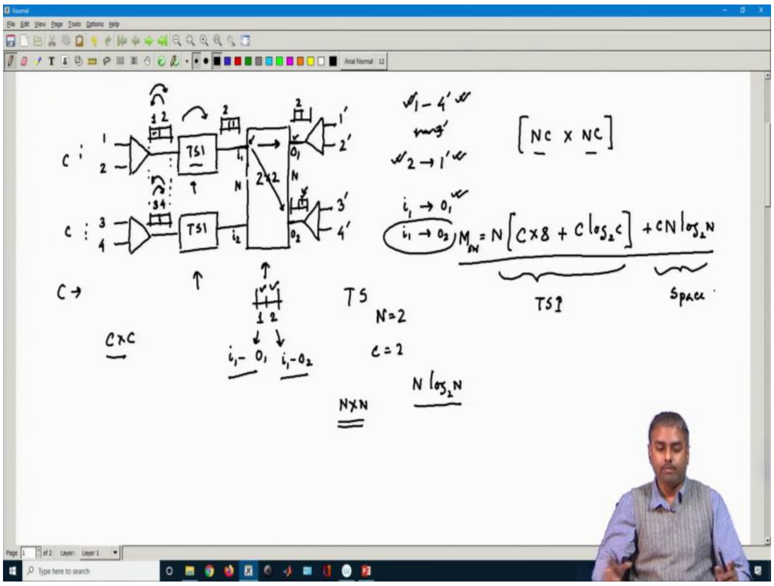
So, that is the series of time switches followed by a space stage. So, we are like earlier; we have done three stages of space switching; now we are actually making stages, but it is not a homogeneous stage; it is a heterogeneous stage of switching. So, some stages are time switches; some stages are space switch means our next discussion will be preoccupied with this kind of switching architecture.

So, where we will be combining the space and time switch, we have already seen there are some advantages. Time switch if we want to make it multistage, then probably the advantage was it for every TSM or TSI it goes through, there will be a 125-microsecond delay, and there is a restriction on the speed also; we have seen that.

And space switches, if we give too many space switches, then the stray capacitance effect, as well as the switching complexity, goes means it does not happen to be scalable. So, that is why we are trying to actually combine these two time and space switches, and we will try to see how we actually do switching through this ok.

So, this is the typical architecture of a TS switch ok. So, the first stage is the time switch the second stage is the space switch so; how does this operate? So, what it is done, let us try to give one example of this one. So, we will come back to this later on; let us try to do it.

(Refer to Slide Time: 03:03)



Let us try to see. Let us take an example of a 4x4 switch ok. So, basically, there is a multiplexer, and multiplexer has only two inputs. So, basically, as you can see over here in a particular frame. So, I am just showing this frame; this will be 125 microseconds. We have already seen that as many we put over here input that many slots will be coming over here.

So, that is the slot size in terms of means every slot is occupying 1 byte, ok. So, that is PCM encoded voice that will mean one sample of that represented by PCM encoded voice sample. So, 8 bits will be occupied over there, and this frame is repeated, so it is the 125 microseconds we are showing.

We will assume over here, of course, that is a very important criterion that we will also be discussing that things are all synchronized. So, at the input of the switch, whatever is coming, all these frames are somehow synchronized; otherwise, the switching will not happen.

So, that is something we will be trying to appreciate over here, and then we will see how to solve it; that means it might be that two of these inputs are coming from different locations.

So, how do I guarantee that their means frames will be synchronized and their clocks will be synchronized? Those synchronization issues will talk about that later on. Right now, we will assume that they are synchronized; that means this frame boundaries are perfectly synchronized.

So, they exactly come at the same this one; the clocks are synchronized, so the frame durations are all synchronized; ok, so, this is what we will be assuming. Now, this TSI can do at the output it can swap these things. So, initially, what will be happening over here is data from 1 will be there; over here, data from 2 will be there ok. So, that is how the data will be occupied.

A similar thing will be happening over here, so data from 3 and data from 4 ok. So, this is how it generally happens because multiplexers will be doing that; the first one will put in the first slot, the second one will put in the second slot, and so on, it will keep on repeating every 125 microseconds.

So, this TSI can actually swap this, so that is the functionality of a TSI; TSI can actually do this swapping. So, it can put it can keep this sequence. So, that is one of the switchings that I do not swap anything. So, basically, I want to keep that thing intact, but I can also swap 2's data. I can put in 1 and 1's data; similarly, I can put in 2.

Similarly, 3 and 4 I can swap, or I can choose not to swap. So, any of those options I can do at the output of TSI. So, followed by this TSI, what we are doing will be putting a space switch of 2 crosses 2. So, basically, I need a 4 cross 4 switch, but I am actually eventually doing it because of the time switch.

Now, I can go away with 2 crosses 2 because it is all multiplex. So, I can demultiplex it, and finally, I get an output of 4 crosses 4. So, this is 1', 2', 3', and 4' ok. Now, think about the switch, so let us say I want to do a connectivity from 1 to, let us say 4'. I want to make the connectivity ok.

So, 1 to 4' I wish to do now let us try to see what will be happening over here; what should I do? So, if I have to connect 1 to 4', what are the options I have? So, first, over here, I have to decide whether I should do time swapping or not. I will come back to that should I decide that ok. So, this data 1 should come out from this 4.

So, for that first of all space switch, let us try to see what kind of connectivity I should give. Definitely, because if 1's data that is coming over here has to be connected to 4, which is coming out from here, so I have to give this connectivity; space switch connectivity must be like this, ok.

So, now let us try to see whether once data can come to 4. So, think about over here this time slot whichever data comes over here in this slot that must be coming out from 4'. So, if I have to bring that data from 1 over here, it must be over here that data of 1 must be over here.

So, how can I bring that over here? Because over here, the data was in the first slot, so I must swap it. So, this is where I get the swapping decision. If I was connecting 1 to 3' instead, then I could not have swapped. I could have kept 1's data in the first slot only in the time slot interchange. I do not swap anything and then do the space switching like this. It will be once data appears in the first slot, and that data will be coming out from 3'.

But because I am doing this switching 1 to 4'. So, it's absolutely essential that at the time slot interchange, I swap the data. So, 1's data comes over here now; this connectivity should be there. Now if you carefully see that, let us say 2 that data from 2 now I add this. So, this is not my switching 1 to 4' I want to connect.

Now, 2's data I want to connect it to let us say 1' ok. So, 2's data must go to 1' means this. So, the switch connectivity must be like this, ok, because 2's data will be coming over here wherever it is in the time slot interchange; it will be connected to the space switch in this particular port, ok.

So, and to data must go to 1', so, therefore, it must be connected to this port also. So therefore, the switch connectivity should be like this ok and 2's data 1' is that a valid option because now 2's data will be over here, and whenever I do this switching, 2's data come over here, and that data will be eventually because it is in the first slot so it will come out over here.

So, this is possible, but now you might be asking to look at the space switch configuration. Simultaneously we have a space switch configuration that makes the connectivity; let us say this is input 1 of the space switch ok. So, let me erase this part, ok. So, that is input 1 of the space switch; this is input 2 of the space switch.

So, what is happening is the space switch; this is output 1, and this is output 2. So, in space switch configuration, if you try to see input 1 to connectivity output 1 that is required for this connection and also input 1 to output 2 that is required for this connection.

So, simultaneously space switch has to be connected in two different places, which is a danger that we have never done. Earlier, the switch connectivity was always giving one port to one port; then, only the switching was happening; now, the space switch gets confused. Is it so? Probably not; there is a solution.

If you carefully see within the time slot, if you now see if the space switch is also if I just consider within this time slot in the first time slot and there is a second time slot in the first time slot, space switch must be connected. So, the first data is 2, so the first-time slot space switch must have this connectivity. So, over here, it should be i 1 to o 2.

In the second slot, ok, sorry that that is wrong, I should not have this. So, if you see the first slot means this data is from 2, so that connectivity is this one ok I 1 to output 1 in the second slot is this data ok. So, that connectivity is this one. So, second slot only, I need I 1 to o 2 connectivity. So, that means, now, this space switch has different time slots and different configurations.

Whenever I do space-time, switching the space to switch configuration also within the 125 seconds, there are multiple slots at every slot different configurations will be required ok, so that is something you have to keep in mind, ok.

So, that is the extra thing that you will have to do; that means, within a particular time slot or within a particular 125-microsecond slot, you have multiple slots, and in every slot, there is a corresponding switch configuration, and you have to keep on doing that.

Remember this particular configuration will keep on repeating in the next slot and the next slot. So, every 125 microseconds, this particular configuration will be repeated. So, if I put these configurations, then this will be repeated every 125 microseconds. So, corresponding to every slot, these kinds of configurations have to be put in ok.

So, that is what you will have to do in space means a time and space switch. So this is called a TS switch, and this is how the TS switch is realized. So, basically, in TS switch, you might have an option of time switching. Is it depend on where at the output port you have to go, so you have to decide it the way I was deciding?

So, first, you try to see should I require a time switching because it depends on the output port where you want this one to go. And then, accordingly, you have to see whether there is a space switch requirement and in what slot it is coming to the space switch. So, accordingly, you put at that particular slot a particular space switch configuration; ok, this is what you will have to do.

So, generally, these are all done automatically, so you will have to do this auto reconfiguration. So, for auto reconfiguration, what you need is the kind of amount of memory that you will be requiring. So, now, let us try to see what is the overall memory that you will be requiring. So, now, I will generalize it; so let us say there are C number of slots over here ok, and there are N input ports and N output ports.

So, overall the switch dimension, as you can see, is an NC cross NC ok, so, that kind of switch dimension. So, now, you can see by just making a space switch of N cross N. Of course, with this N cross N again, you can do multistage space switching to save some of the switching ports; you can do that, but with the N cross N switch, because of this time switching, I now have an NC cross NC switching ok.

So, I can actually enhance the dimension of switching by just doing this combination of space and time switching. So, now, with the N cross N space switching, and if I have C multiplex stream, then NC cross NC switching I have over here, it was this space switch was 2 cross 2. So, my N value was 2 C value was 2.

So, that is why NC crosses NC; that means 2 crosses 2 into 2 4 crosses 4 switches, I have realized. So, 4 input ports and 4 output ports with the 2 cross 2 space switch, I have now realized 4 cross 4 switch ok. So, similarly, in general, this will be the case now; let us try to see how much control information I need, ok.

So, what are the control things we need; that means the memory that we need. In the TSI, we need this memory of data storage, and we need the memory storage ok or the control bit storage. So, let us try to see for a C this is a TSI is a C cross-C switch. So, for that, we have already calculated what should be the amount of data required.

So, that is C amount of data has to be stored, and each one of them is 8 bit long. So, C into 8 amount of data storage is required, plus there is control storage. So, C is the number of control, and each one of them has log 2 C; this is something we have already calculated. So, this is the requirement of every TSI. How many TSIs are there?

So, as many input ports to the space switch, are there that many TSI I have to put at the input? So, N number of such TSIs will be there ok. So, these many control bits or memory bits are required for the entire first stage of the time switch ok. Now, let us try to see how many control bits are required for the space switch.

So, I have a space switch of N cross N right in an N cross N switch, you remember, because it is a symmetric switch. So, whether we do input-associated or output-associated, it does not matter the amount of this multiplexer or demultiplexer control that is required was N log 2 N because it is symmetric. So, it does not matter whether input-associated or output-associated will always be getting this number.

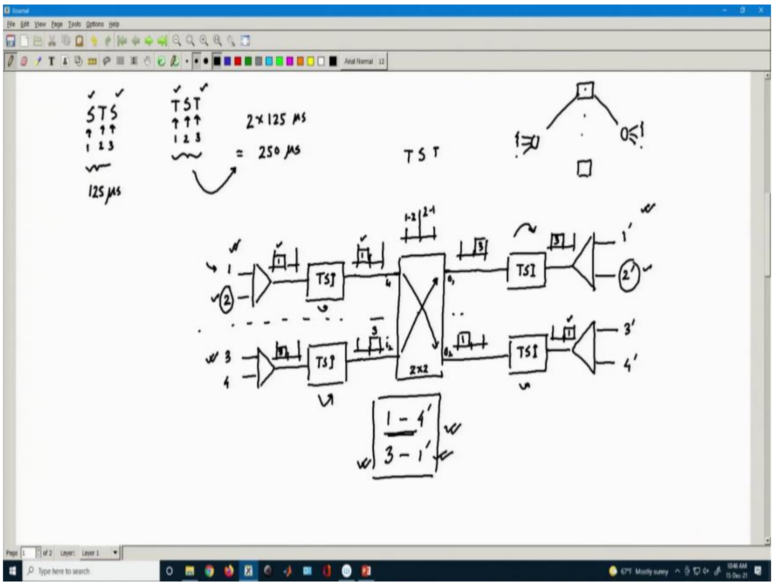
So, for this space switch, I need N log 2 N number of control bits that have to be stored. Now because of C time slots, you require those control bits for every C instance or C for every slot. There are C number of slots for every slot. These many control logics are required. So therefore, C into this many control logic will be required for the space switching.

So, that should be the memory requirement of the switch M switch; it’s a very simple calculation. So, whenever you do this, you will be able to see how many memory bits are required. So, this particular portion comes out from the TSI part or time to switch part, and the portion of this particular part comes out from the space switch portion.

This is exactly what happens in our space-switch time,, which means space-time switching ok. So, I think we have now understood this part. Now we will make it more like that Clos’ argument three-stage switching ok. So, we have just understood how to actually combine time and space switches, but this is probably not the means most interesting switching that we can see over here. This might have restrictions, ok.

So, we will try to see means this might have blocking. So, that is something that we will be discussing next. So therefore, we need to actually again come up with some kind of 3-stage switching where there is enough sharing. So that we have a better switch design, so, let us try to see a particular combination of 3-stage switching by this time and space switch. So, that should be our next target.

(Refer to Slide Time: 19:43)



So, now we will be doing, we will be doing two combinations one is space-time space switching ok, and the other one is TST switching. So, basically, it is a 3-stage switch, as you can see already, that is stage 1, that is stage 2, and that stage 3 or over here, this is stage 1, stage 2, and stage 3, ok.

So, it depends on what you put; they have to be symmetric in nature; you have also seen in 3-stage or 5-stage Clos' switches you have seen that they are symmetric. So, it is also a multistage switching. So, we will be doing symmetric switching over here. So, either both side space or both side time middle portion is space switching, so one of them will be doing ok.

So, if I do this kind of switching, what will happen? Of course, we will be requiring less amount of space switching, ok. So, that is one advantage that we will be getting over here if we do this ok. But the problem is if I do that, then because there are 2-time switches so, it will always have two into 125 microseconds delay overall delay or 250 microseconds of delay.

So, every switching element of this kind of configuration will have a 250-microsecond delay. Whereas, if I do that, I will be only getting a 125-microsecond delay, but of course, because there are two space switches. So, the switch will be bulkier, and more switching elements have to be required. So, this is the trade-off that we will be trying to make.

So, let us try to understand how, again, we can talk about this blocking probability of these switches, how we design a strictly non-blocking switch of this kind of configuration, how we design a rearrangeably non-blocking switch, or how we actually go about doing some amount of blocking or an allowing some controlled amount of blocking the way we have done for 3 or 5 stage switches.

So, corresponding to Lee's graph, if we can talk about it because that is part of the designing, we want to actually reduce the number of elements that are put forward for switch designing. So, we have a cost advantage, of course; we will have the stray capacitance that will be lesser, we will have less amount memory requirement probably, and the switching speed will be adjustable. We can scale the switch.

So, due to all those things, we will probably be trying to design the way we have done it for the space switch alone, ok. So, let us take one example; let us take the first TST switch, ok. So, we will take a TST switch. I will again give an example of a 4 cross 4 switch ok. So, let us try to see how I do a TST switch ok.

So, basically, what we will be doing is initially time stage so, so it will be followed by a multiplexer going to a TSI; it's a 4 cross 4 switch with a space switch requirement of 2 cross 2 so, so I am again doing the same thing. Now, that gets into an ok; let me provide some space. So, this is the 2 cross 2 space switch.

The only difference is it again goes to a TSI ok. So, now, let us try to see how I do switching and how I make it a strictly non-blocking one ok. First, what will I do? I will do the general regular switching, and then you will probably mean getting the appreciation of blocking that will be coming to this switch, ok.

So, let us say again what is happening if you see this middle part of this port. Ok, that is where up to that it's all separated, right? So, that is where there will probably be a sharing that will be happening so, so we will try to appreciate that part where exactly the sharing is coming up; we will see those things, but that is probably where things will be getting shared, ok so, this middle part ok.

So, now let us try to give connectivity. Let us see where the blocking condition comes into the picture, so when we talk about this blocking, if you remember this Clos' particular argument. So, you had something over here middle; there were some stages, k stages, and each of these stages was getting shared among the input and output.

So, what we were trying to see is from here, there are multiple inputs that are coming, and at a particular target node, there are multiple outputs. So, suppose all these are already occupied; that means some connectivities are there, and this last one is trying to connect to the last element. So, a similar condition will be trying to create.

So, let us say from 2 to 2dash is my targeted this one; my 1 is already occupied. Remember the Clos' argument was 1 was occupied, but it was only occupying the input portion output; it was going to some other block. So, it was not coming to the same block, and 1 dash also is occupied but not coming from here. It's coming from some other place.

So, that is the kind of thing we will be putting to appreciate the blocking. So, let us first see, suppose I already have regular connectivity 1 to let us say it's 4 dash ok. So, 1 2 because 2 to 2 dash I want to connect. So, think about this argument 2 to 2 dash. I am trying to give connectivity, whereas these two are already occupied.

So, I have to search the blocking condition that when these two are already occupied in the worst case scenario when 2 to 2 dash, I have to give connectivity, but I am unable that is when the blocking will be happening. So, when the worst case situation happens according to Clos' argument, if this one is already occupied, but not going to this one because then things will be shared.

So, 1 is going to some other place. So, 1 to 4 dash we have taken, and this 1 dash is also occupied, let us say 3 is giving connectivity to 1 dash. So, from some other block, it is coming to 1 dash. So, let us first make this connectivity because these are the past connections that are already there. So, let us make this connectivity so, initially, let us say if I have 1 to 4 dash this connectivity, I will have to give.

Now, if you see, I have added flexibility. So, let me put all those synchronized slots so there are 2 2 slots. So, everywhere I have those synchronized slots. So, they are initially free nobody is connected, so I will be putting my connectivity. So, everywhere I will put this, remember I only have restrictions at this input and at this output.

So, if 1 has to connect to 4 dashes, that means 1's data will be coming over here. I have no control over this. So, because this is being fed over here, that must be coming in the first slot, so, at this instance, I have no control. I have control after TSI. So, over here, I can do the switching, and I also have control over here, ok.

So, 1 is coming over here, and then I want to put him to 4 dashes so; that means that should come over here this 1's data must be coming over here that I do not have control because if I have to switch it to 4 dash, I have to put it in the second slot there is no other option ok. Now if you see this connectivity, I have to give.

So therefore, I can already see, of course, this switch also will have a slotted option ok. So, in different slots, I can do different reconfigurations of the space switch also. So, now, I have two options; as you can see, 1 is in the first slot, and over here, 1 is in the second slot. So, what I have to do is, I have to flip it so this flipping can be done either over here in this TSI or over here, ok.

So, that is my decision. I can do anything, ok. So, I do not have to because I now have two stages of time switching; earlier, that was not the case. I was only having only TS switch, only one option I was having. So, I had to switch it at the beginning wherever I got the chance.

So, that is why that is more destructive over here. I have the flexibility; I can choose to put it over here. So, suppose I choose that, then what happens once data comes in the first slot? So, definitely, because the space switch cannot do time swapping so, I have to do this switching, and over here, only 1's data will be coming.

So, basically, I have chosen not to swap in the first-time slot interchange, and then the space-switching configuration has to be like this. So, basically, if I say input 1, input 2, output 1, output 2. So, basically, space switches in the first time instance. So, this is the first time instance I have had to do a cross-switching; that means I have to give connectivity 1 to 2 at this first instance; I have to do otherwise, that particular data will not become.

And in TSI, I will have to do a swapping in the second TSI, which is something I will have to do. Now, because of this configuration because of this decision, I have already decided on a particular switching ok. So, what will I now do? I will try to see how do I give this next connectivity.

So, suppose these connections are coming in order. So, the 1 to 4 dash request first came. I did this switching now the next connection, 3 to 1 dash is coming ok. So, how do I do the switching? Now from 3 again, I have no choice. This will come over here. So, this is threes data that will be coming over here, ok, and 3 to 1 dash I will have to put. So, it must be over here, ok.

So, now, what can I do? What I have told is something like this, I might actually choose to swap it I might not choose to swap it. So, suppose I decide to swap it, ok, because this is at the beginning, as long as I can do things, I will be actually taking my decision, ok. So, first I do a swapping ok so, I do a swapping. So, data for 3 come over here.

Now, after swapping this swapping, what will happen? It's coming in the third slot, remember, ok. Now I have to put it over here, so I have to do this switching. So, data from

3 comes over here, ok. So, data from 3 come over here. Now I have to do another swapping because 3 have to go over here, then only I will be getting at 1 dash, so I do a swapping.

So, now the switch configuration is at the second instance I have to do this. So, again 2 to 1; that means the switch can be either cross or bar state. So, it has to be cross-connected so that 1 input to 1 output mapping is being done. So, input-to-output mapping always has to be 1 to 1 in a space switch; otherwise, there will be confusion. So, there will be data collision, so you do not want to do that.

So therefore, this is all right. I put at the second this one 2 to 1 configuration. So, all this is fine now. In the next class, what we will try to see once I have done this now is if I try to give that 2 to 2' connectivity, you will see that there will be a blocking scenario.

We will talk about that blocking scenario. We will also talk about how we can make it rearrangeably non-blocking. So, that is something we will talk about, and we will also talk about additional changes that we will have to make in this switch configuration. So, we get a strict non-blocking so, which comes from Clos' argument again; we will talk about that in the next class, ok.