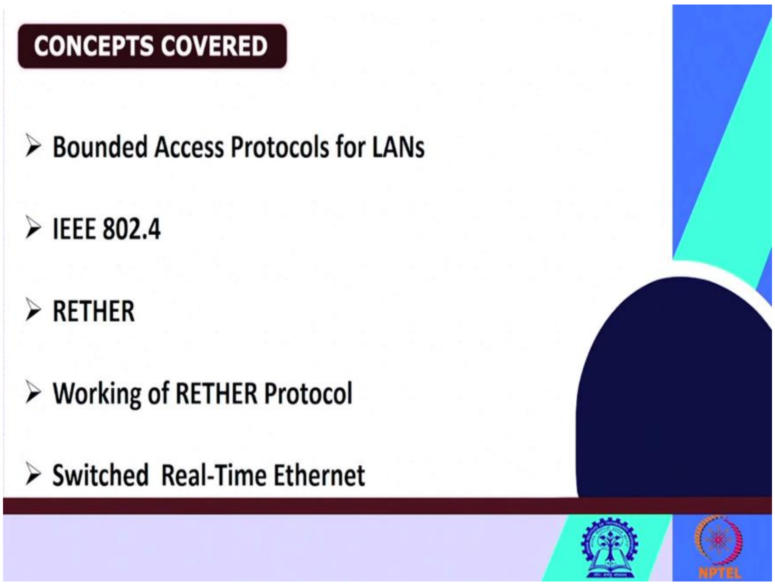
Real Time Systems
Professor Durga Prasad Mohapatra
Department of Computer Science and Engineering
National Institute of Technology Rourkela
Lecture 52
Bounded Access Protocols for LANs

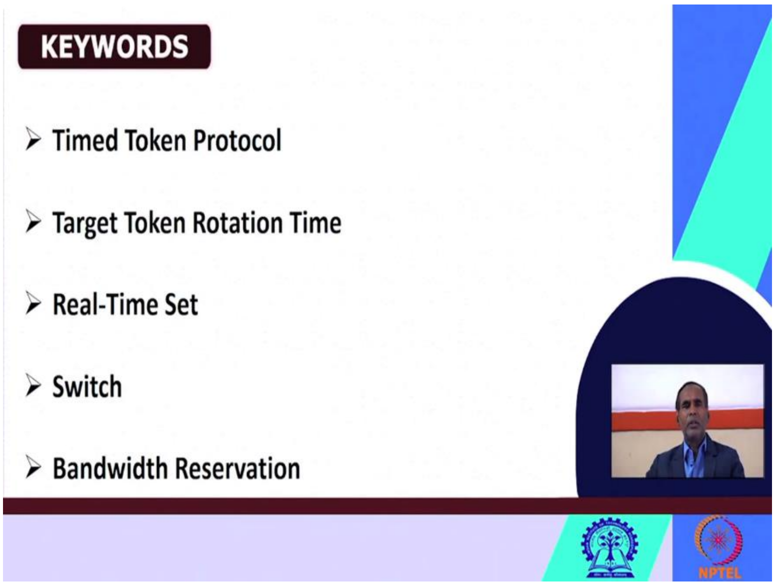
Good morning to all of you. Last class we have discussed about hard real time communication. Under that we have seen two important categories of protocols. One is global priority protocols and other is calendar-based protocols. So, now today we will discuss about the third category of protocols that is known as bounded access protocols for LANs.

(Refer Slide Time: 00:46)



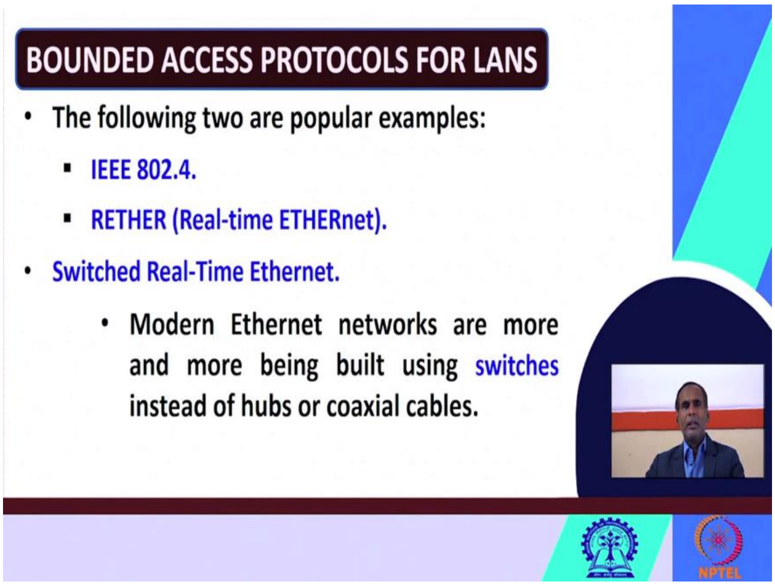
We will see some of the two important protocols which are coming under these bounded access protocols for LANs, those are IEEE 802.4 and RETHER. We will see in detail the working of both IEEE 802.4 protocol and RETHER and besides this we will also look at some basic concept of switched real time Ethernet.

(Refer Slide Time: 1:07)



These keywords we will use where the timed token protocol, target token rotation time, real-time set, non-real time set, switch and bandwidth reservation.

(Refer Slide Time: 1:20)



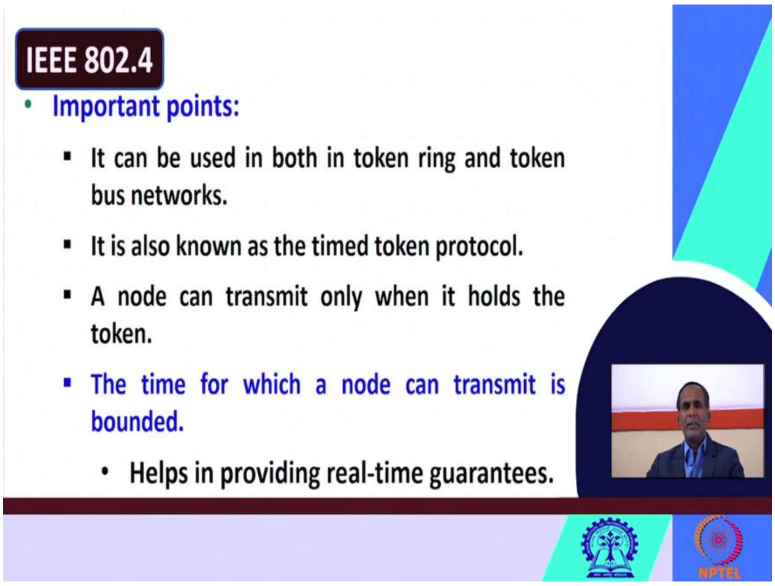
Now, let us see first about the bounded access protocols for LANs. As I have already told you these are the protocols for hard real time communication in LAN, we have last class discussed two important categories of protocols that is global priority protocols and these calendar-based protocol under global priority-based protocol we have seen IEEE 802.5.

And besides that, we have seen what? These virtual time protocol, countdown protocol, window-based protocols, etc. So, the IEEE standard we have discussed for which one, this global priority protocol is IEEE 802.5 please remember. But here under bounded access protocol we will discuss this IEEE 802.4 protocol. So, this is the new one which is used for bonded access protocols in LANs. So, another protocol is RETHER, RETHER stands for a real time Ethernet.

And besides these two, we will also look out little bit about the basic concepts of switched real time Ethernet. Because what happens nowadays, the modern Ethernet networks, they are using more and more in this direction. They are using what? They are using switches. This modern Ethernet networks they are more and more using switches instead of using hubs or coaxial cables.

So, that leads to this concept switched real time Ethernet. That we will see towards the end of this lecture a little bit on the switched real time Ethernet.

(Refer Slide Time: 2:52)

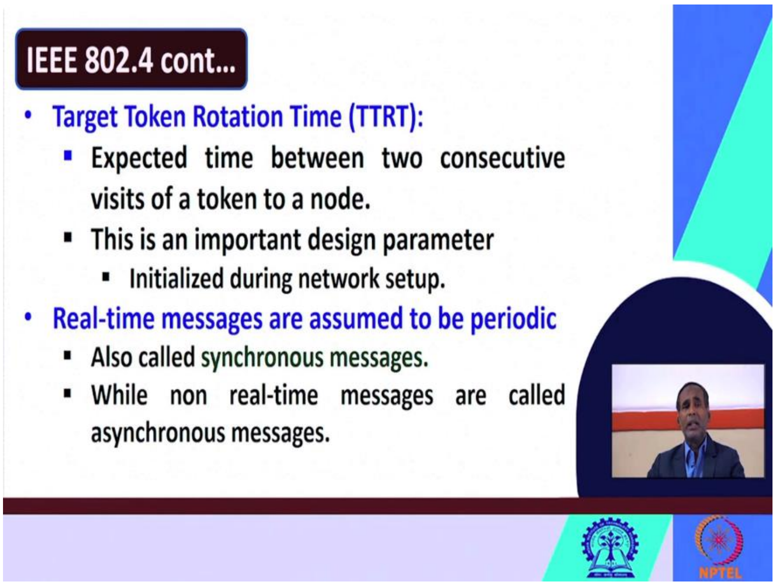


We will start with the IEEE 802.4. Let us see some of the important aspects of IEEE 802.4. First of all, this protocol IEEE 802.4, it can be used both in token ring networks as well as token bus networks. This protocol IEEE 802.4, it is also known as a timed token protocol. So, this protocol is also known as time token protocol. And let us see, how this transmission occurs using IEEE 80.4. So, according to IEEE 802-point protocol, a node can transmit only when it holds the token.

So, only when a node token holds the token then only it can transmit. Now, why it is name is bounded access? I have already told you. Because here the time for which a node can transmit it is bounded. The time for which a node can transmit is it is bounded. Why this is bounded?

What advantage you will get by what making bound this thing? This helps in providing real time guarantees. So, since the time for which a node can transmit is bounded, this helps in providing real time guarantees.

(Refer Slide Time: 4:02)



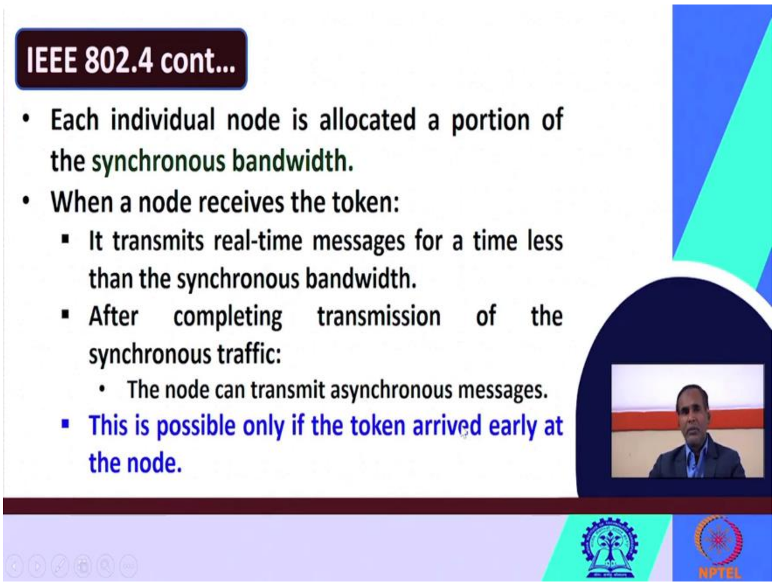
So, in IEEE 802.4 one important design parameter is used what is that? That is special time called as target token rotation time. So, this target token notation time is an important design factor and this is initialized during the network setup. So, this is a very important design parameter. Now let us see how it is defined? This target token rotation time TTRT, it is defined as the expected time between two consecutive digits of a token to a node. So, if a particular node N1 is there, so first time when this token comes, and when the next time the token same token comes to the same node. Then we say what is the time difference that we call is as this TTRT.

I am repeating again TTRT is defined as what is the expected time between two consecutive digits of a token to a node. That means to a node. Suppose, first time a token has arrived at T1, then again it will circulate among all the nodes and when again it will come to the same node N1 say at the time T2. So, this is the difference, so this T2 - T1 this you can say that what is the expected time. So, this difference may be treated as this what TTRT.

So, this is the expected time between what two consecutive digits of a node of a token to a node, when the first time the token arrives to the node when the second time it arrives. So, what is the expected time between these two consecutive digits of a token to the same node, this is known as TTRT. So, normally in IEEE 802.4, the real time messages are assumed to be periodic. We assume that the real time messages they occur periodically, the real time messages are assumed to be periodic and these real time messages they are also known as synchronous messages.

And you know that in a real time communication there can be also some non-real time messages. So, these real time messages are called as synchronous messages while the non- real time messages are called as asynchronous messages.

(Refer Slide Time: 06:21)

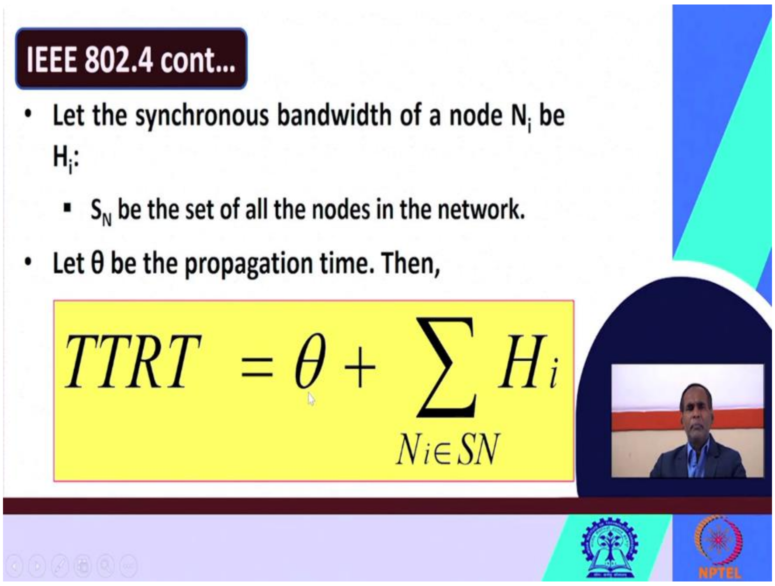


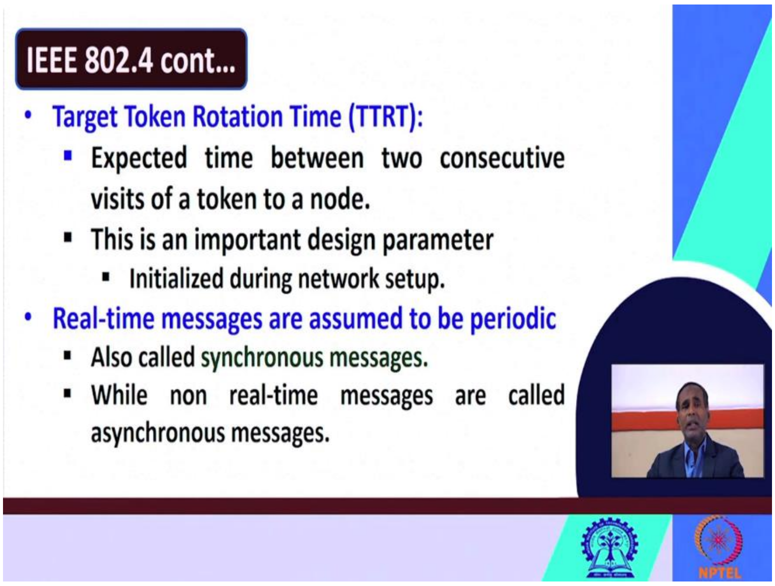
Now, let us see how does this IEEE 802.4 this protocol it works. So, first each individual node is allocated a portion of the synchronous bandwidth. I have already told you what is a synchronous message, what is an asynchronous message. So, first the individual, each of the individual node is allocated a portion, a fraction of the synchronous bandwidth. Now, when a node receives a token, so I have already told you this is based on what a node can transmit only when it acquires the token. So, normally that token will be rotating among the nodes. When a node receives the token then it transmits first, what? The real time messages.

So, whenever a node receives the token, it transmits the real time messages first for a time which is less than the synchronized bandwidth. I have already told you each node is allocated a portional fraction of this synchronous bandwidth. So, whenever the node receives the token, it transmits first the real time messages for a time which is less than this what allocated synchronous bandwidth. Then after completing transmission of this synchronous traffic that means after it has transmitted all the real time messages, all these synchronous messages then what it can do? Then only it can or the node can transmit the asynchronous messages.

So, that means in other words I can show that the node can transmit the asynchronous messages only when or if the token arrived early at the node, so the asynchronous messages can be transmitted, this will be possible only if the token it is arrived a little bit early at the node, then only the asynchronous messages can be transmitted.

(Refer Slide Time: 8:16)

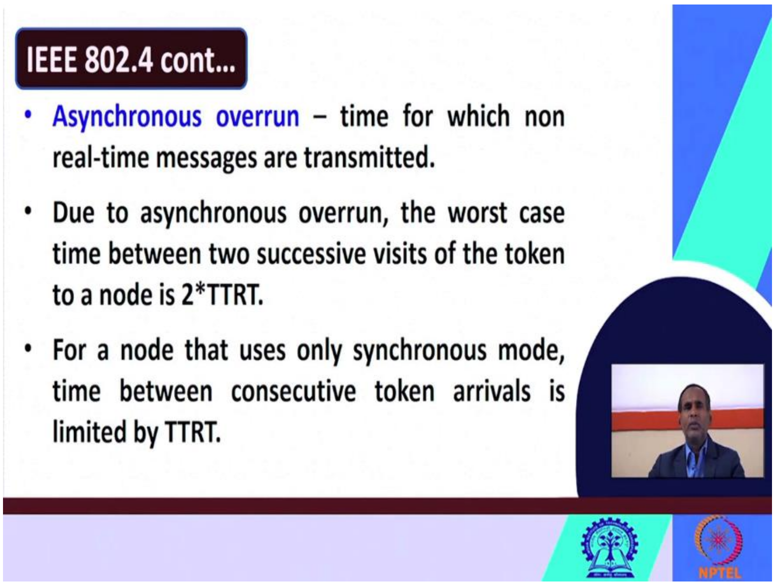


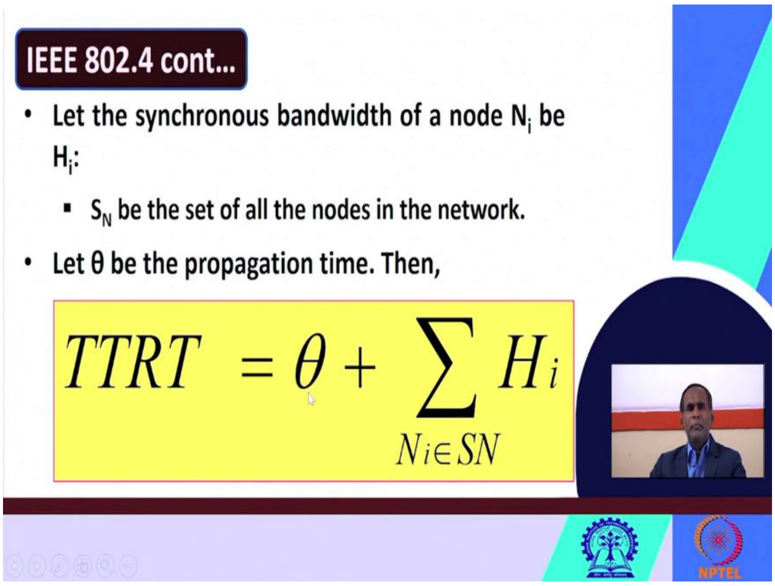


Now, let us derive a mathematical formula that what will the synchronous bandwidth of a node Ni, we call that the token holding time you can say that. Now, let us assume that the synchronous bandwidth of a node Ni be Hi and Sn be the set up all the nodes present in the network. Let us assume that the theta is the propagation time. Then we can define TTRT, what is TTRT I have already told you. TTRT stands for target token rotation time. How much time it will take that the expected time between the two consecutive digits of a token to a node. This is called TTRT or target token notation time.

So, mathematically you can say that TTRT = ϴ + $\sum_{Ni \in Sn}$ Hi where ϴ is what? ϴ is the propagation time. What is Hi? Hi is the bandwidth, Hi is the synchronous bandwidth of a node Ni or in other words I can say that Hi is the token holding time or the node Ni. What is Ni? Ni is a particular node. So, the TTRT, this can be defined as the propagation time plus what the token holding time or the bandwidth of what bandwidth or you can see the summation of the bandwidth of all the nodes, here Ni it belongs to Sn. In this way you can define TTRT.

(Refer Slide Time: 9:48)





Now, let us see how this asynchronous message can be transferred and what is the overrun, what will the worst case time that will be required. So, asynchronous overrun it is defined as the time which the non-real time messages are transmitted. Asynchronous overrun means the time for which the non-real time messages they are transmitted. So, I have already told you the formula for TTRT.

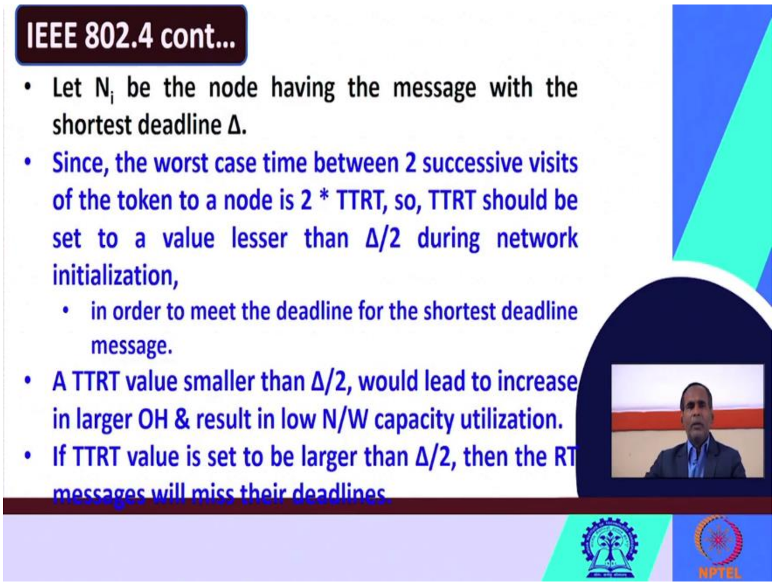
Now, if you will consider the asynchronous messages, it you will consider the time for transmission of these non-real time messages or the asynchronous messages then what will happen? Due to the these asynchronous or due to this synchronous overrun, the worst-case time between two consecutive visits of the token it will be increased by how much? It will be twice of TTRT.

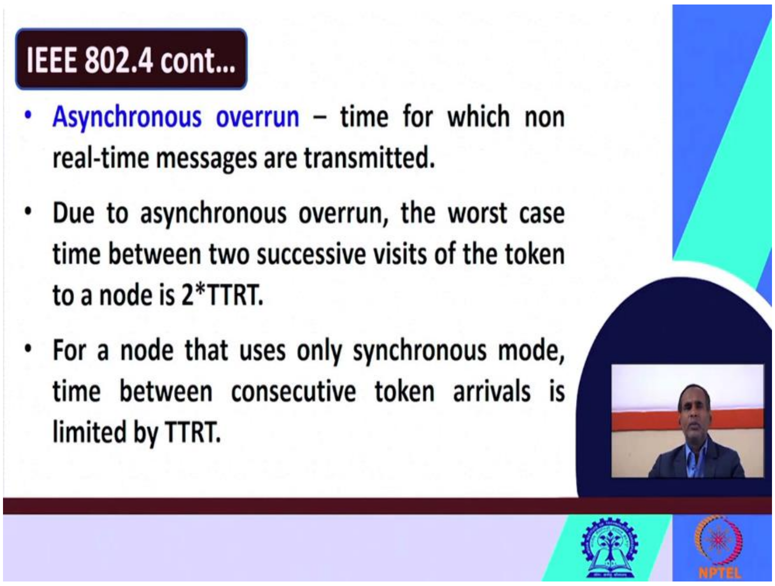
If you are not considering asynchronous overrun then the time is bounded by TTRT. If you are considering this asynchronous overrun, due to this asynchronous overrun the worst-case time between any two consecutive visits. I am saying worst case, the worst-case time between any two successive visits of the token to a node is bounded by how much, 2 * TTRT that means twice of TTRT. If you will consider it will take into account asynchronous overrun or this non real time messages then the worst-case time between two successive visits of the token to a node is becoming how much, twice of TTRT.

Now, for a node that uses only synchronous mode, I have already told you if it uses only synchronous mode that means only it is transferring real time messages then the time between the consecutive token arrivals is just limited by TTRT.

So, two conclusions for a node that uses only synchronous mode that means only it transmits the real time messages then the time between the two consecutive token arrivals is limited by each bounded by only TTRT. But if you will consider the non-real time messages which we call as asynchronous messages due to this asynchronous overrun then the worst-case time between two successive visits of the token to a node each how much, twice of the TTRT.

(Refer Slide Time: 11:59)





Now, let us find out what another mathematical formula. But before going to that mathematical formula let us see the required fundamentals. Suppose Ni be the node having the message with the shortest deadline delta. Suppose, how many nodes are there, suppose Ni is a node which is having the message with the shortest deadline. So, it contains the message having the shortest deadline delta. I have already told you the worst-case time between two successive digits of the token to a node is how much? 2*TTRT just I have told you, is not it. That due to asynchronous overrun, the worst-case time becoming 2 * TTRT.

So, since the worst-case time between two successive visits of the token to a node is becoming twice of TTRT. So, the TTRT value, how much it should be set? So, TTRT should be set to a value lesser than delta by 2. Where delta is what? Delta is the shortest deadline. So, since the worst case time between two successive visits of the token to a node is 2 into TTRT or twice of TTRT. So, the value of the TTRT, so the value of the TTRT should be set somewhat lesser than delta by 2 during the network initialization, what is delta? This is shortest deadline.

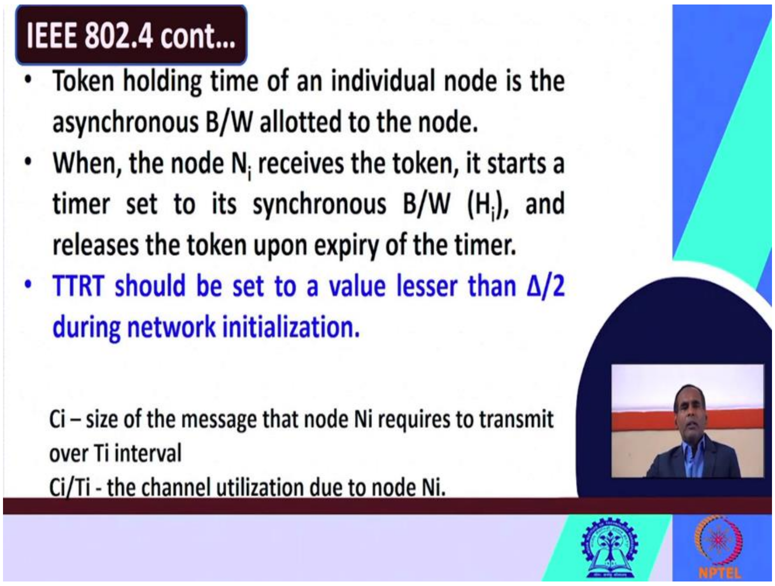
Why, you will set the value lesser than delta two? Because you have to meet the deadline for the shortest deadline message. So, if you want to meet the deadline of the shortest deadline message which is here delta then you have to set the value of TTRT is equal to delta by 2. So, since the worst-case time between two successive visits of the token to a node is 2 into TTRT.

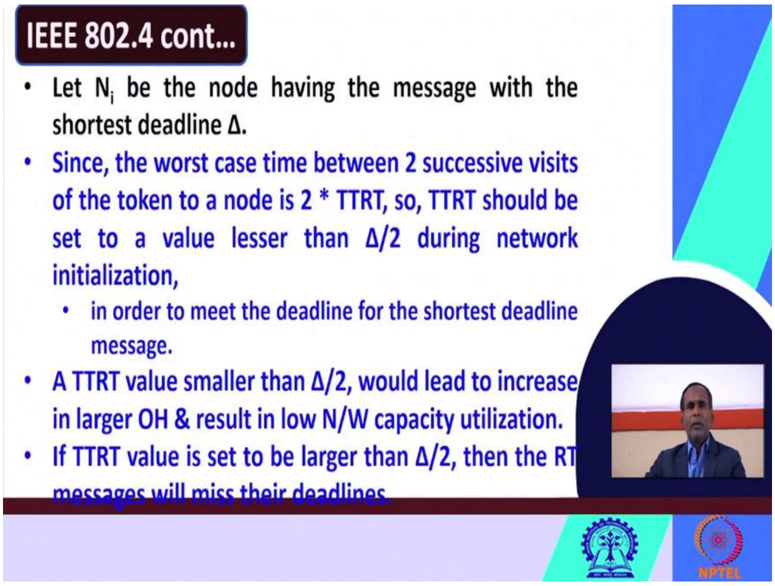
So, TTRT should be set to a value lesser than delta by 2 during the network installation, why? In order to meet the deadline of the shortest deadline message. Now, let us see the consequence, if you will support set a value smaller than delta by 2 or if you set a value that means you set the TTRT value greater than delta by 2 then what will the consequence? So, if you are setting the value lesser than delta by 2 what will happen? This will lead to increase in large overhead, overhead will be large and this will result in low network capacity utilization.

Similarly, if you will set the TTRT value greater than delta by 2 what will happen? Obviously, the real time messages they will meet their deadlines. So, that is why we should set the value of delta TTRT equal to delta by 2. A TTRT value smaller than delta by 2 would lead to increase in larger overhead and it would result in low network capitalization or the other hand if TTRT value is set to larger than delta by 2 then the real time messages will miss their deadlines.

So, that is why we should set the value of TTRT is equal to what, it should be lesser than delta by 2. It should not be much smaller than delta by 2. It is just closure to delta by 2 or even you can say just equal to delta by 2. I have written lesser that means just closure to delta by 2. It should not be much smaller than, it should not be much higher than the delta by 2.

(Refer Slide Time: 15:20)

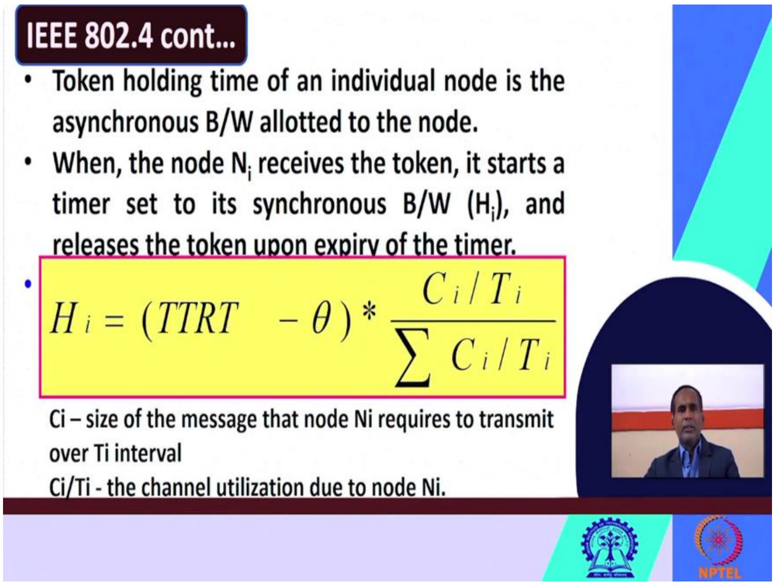


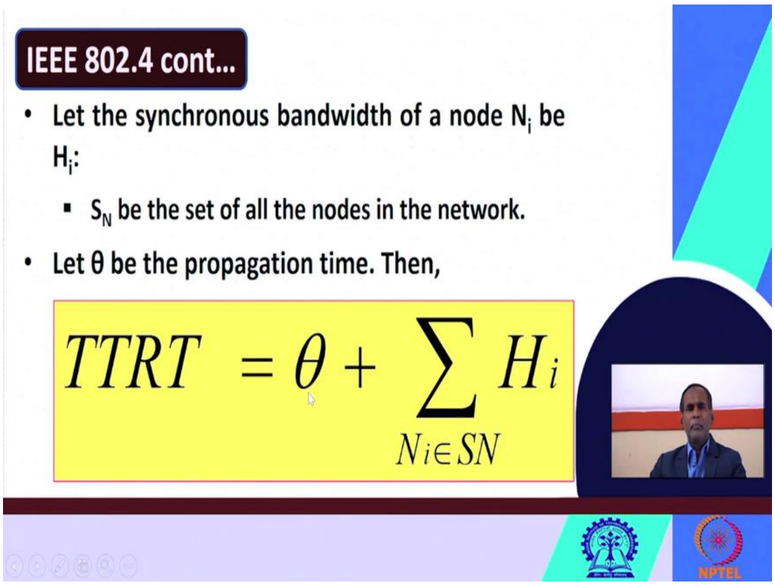


Now, we will find out another mathematical equation, I have already told you here Ni be the node having the message with the shortest deadline delta. The token holding time of an individual node is the asynchronous bandwidth allocated to the node. You know the token holding time of an individual node is equal to the asynchronous bandwidth allotted to the node.

Now, as soon as the node Ni it receives the token then what does it do? It starts a timer; it starts a timer set to its synchronous bandwidth. What is the synchronous bandwidth? You have known this synchronous bandwidth is normally used in notation Hi. So, it starts a timer to set it synchronous bandwidth Hi and then it releases the token upon expiry of the timer, when the timer expires it releases the token. I have already told you the value of TTRT should be set to a value just lesser than delta by 2 not much less than or much smaller than, so it should be set to a value just lesser than delta by 2 during network initialization. So that what will happen? The deadlines of the shortest deadline can be met.

(Refer Slide Time: 16:36)



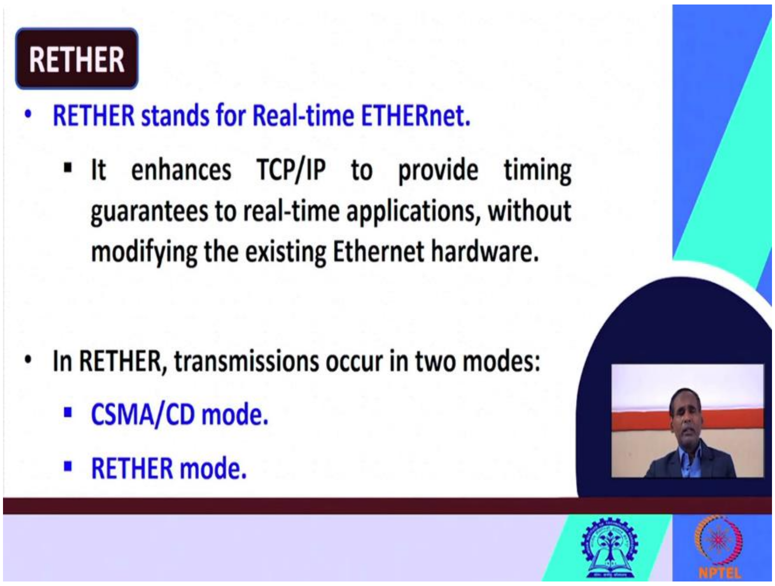


So, now we can write down the mathematical equation how you can find out the value of Hi? Hi can be found out TTRT - ϴ *Ci / Ti / Ci - Ti. We want to find out the value of this what synchronous bandwidth Hi or the we can say that token holding time. So, the value of this what synchronous bandwidth Hi can be expressed that value of TTRT minus theta, what is theta I already told you. Theta is what? Here we have already written theta is the propagation time.

So, Hi = TTRT - ϴ * Ci / Ti / ∑Ci /Ti. So TTRT, already we have known, theta we have already known, Hi is the synchronous bandwidth only remaining Ci and Ti. What is Ci? Ci is the size of the message that node Ni required to transmit over Ti interval. Ci is transferred the size of the message that the node Ni it requires it needs to transmit over the time interval Ti and what do you mean Ci / Ti? It represents the channel utilization due to node Ni. Ci / Ti it represents the channel utilization due to node Ni. So, now you may be asked to similar types of questions in the examination.

So, you may be asked to find out a suitable synchronous bandwidth given the value of TTRT, given the value of theta, given the value of Ci and given the value of Ti, you can easily compute the value of the synchronous bandwidth Hi by using this formula. In the examination similar types of problems might be asked.

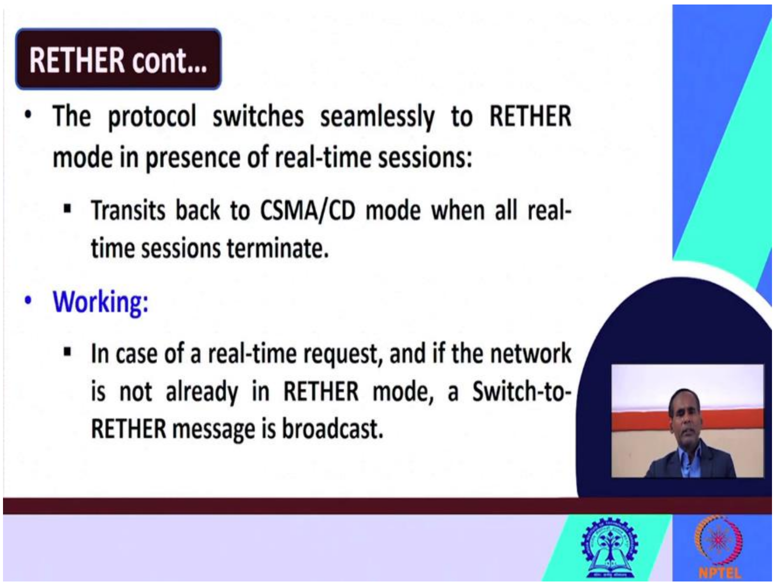
(Refer Slide Time: 18:27)



Now, we will go to the next protocol that is RETHER, I have already told you RETHER stands for real time Ethernet. What does it do? This protocol it enhances TCP, IP already newer computer network paper. So, it enhances the TCP, IP protocol that to provide the timing guarantees, to provide the real time guarantees to what? To the real time applications without modifying, without changing the existing Ethernet hardware.

So, in RETHER, the transmission occurs in two modes. One is the CSMA, CD mode and the other in the RETHER mode. CSMA, CD already you have known, now let us quickly look out this RETHER mode or how data transmission occurs in this RETHER.

(Refer Slide Time: 19:14)



This protocol RETHER, it switches to a RETHER mode in the presence of real time sessions. So, whenever real time sessions are there, this protocol switches to what, to the RETHER mode.

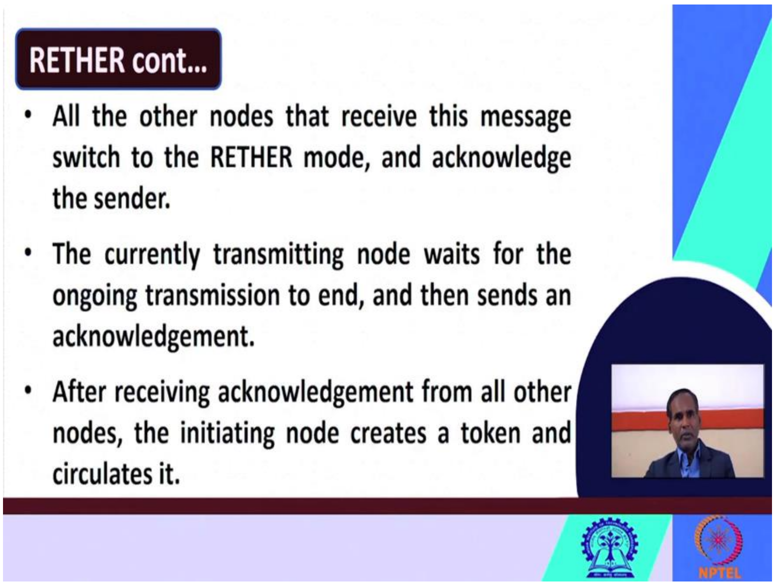
So, this protocol switches to RETHER mode in the presence of real time sessions and then when it will come back to the CSMA mode because I have already told you there are two modes CSMA, CD mode and RETHER mode. This protocol it when it switches to RETHER mode? It switches to RETHER mode in the presence of real time sessions and it comes back, it transits back to CSMA, CD mode when all real time session terminates.

So, when real time sessions are containing, when real time systems are present, the protocol switches to RETHER mode and it is transmitted back to the CSMA, CD mode when all the real time sessions they terminate.

It is very important please remember, this RETHER protocol switches to the RETHER mode in the presence of real time sessions and it transits back to the CSMA, CD mode when all the real time sessions they terminate. Now, let us try to understand the working of this RETHER protocol. In case of a real time request, suppose the real time request is arriving in case of a real time request and at that time, if the network is not already in RETHER mode, if at that time network is not already in the RETHER mode, then what will happen? Immediately it will switch, the protocol switch to the RETHER message.

Because I have already told you the protocol switches to RETHER mode when in the presence of real time sessions or when real time requests, they arrive. So, in case of a real time request and if the network is not already at that time in RETHER mode, then a switch to RETHER message is broadcasted. At that time switch to RETHER message is broadcast or broadcasted.

(Refer Slide Time: 21:09)



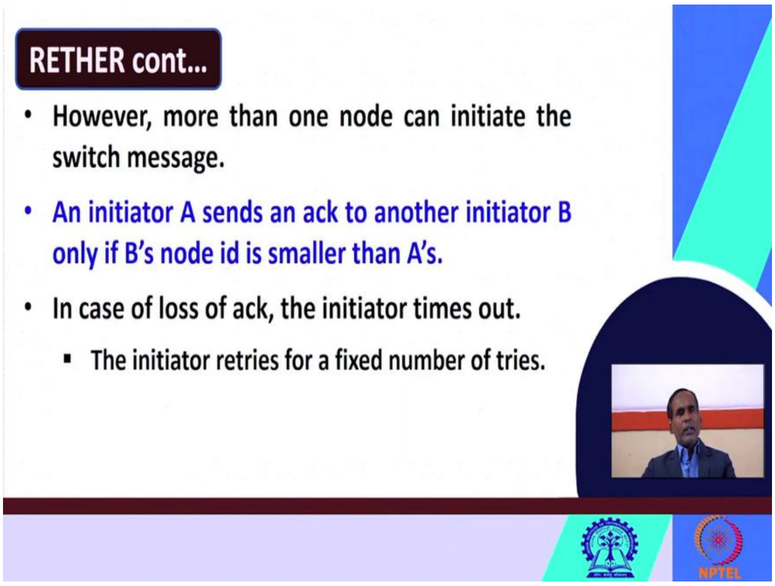
Now, other nodes will receive that broadcast message. So, all the other nodes that received this broadcast message or they will do, they will simply switch to the RETHER mode and they will acknowledge to this sender, yes, we have switched to the RETHER mode. Now we are in RETHER mode, then all those nodes, those will receive this message immediately they will switch over to the RETHER mode and they will acknowledge the same, they will send an acknowledged message to the sender, yes, now we have switched to the RETHER mode.

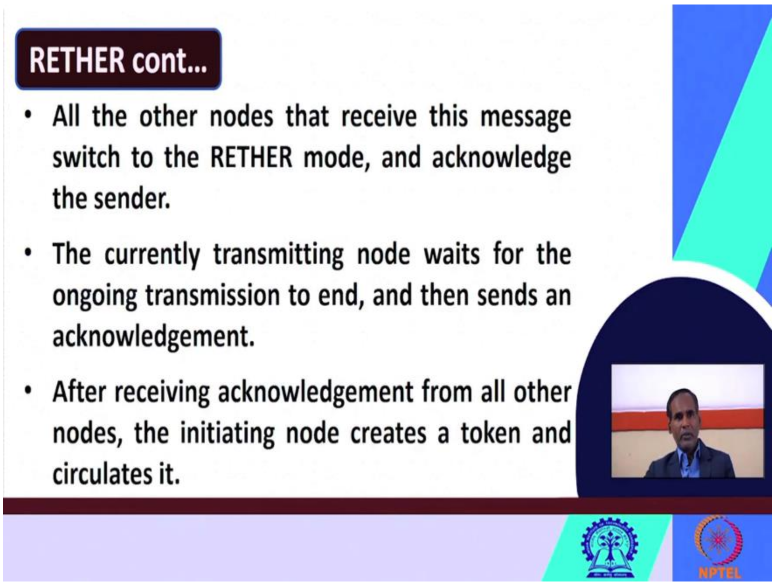
Now, suppose if any currently node it is transmitting something, then what it will do? The currently transmitting node now, immediately it cannot switch to the RETHER mode, it will just wait for the ongoing transmission to be completed.

Because some message is now under transmission. So, the currently transmitting node now, it will wait for the ongoing transmission to be end or to be completed and then only it will switch to the RETHER mode and then it sends an acknowledgment to the sender, yes, my transmission is completed and now I am ready to switch to the RETHER mode. So, then now the sender, the initiator it will receive the acknowledgments from all the other nodes as well as the current transmitting node.

Now, let us see what is happening? After receiving acknowledgement from all other nodes as well as the currently transmitted node then what the initiating node will do? The initiating node it will create a token and circulate it. Because here it is based on the circulation of a token. So, when the initiator node it receives the acknowledgement node, acknowledgement messages from all the other nodes including the currently transmitting node then it will create a token and it circulates it.

(Refer Slide Time: 22:54)





Because I have already told you when the node will get the token acquired, the token it can circulate. Now, it has got acknowledgment from all other nodes including the current transmitting node. So, that means all the transmissions are over. So, now the initiating node it can transmit some message. So, first the creates a token and then circulates it. So, after receiving acknowledgment from all other nodes, the initiating node creates a token and then it circulates it. So, after acquiring the token it can send the message it can transmit the message.

Now, what will happen? It is possible that more than one node can initiate this switch message. Here we have seen that only the initiator can send this what message or it can transmit the message when it will receive the acknowledgment from all other nodes. It will create a token then it can transmit the message. But it is also possible that more than one node can initiate this switch message that is also possible.

At the same time, two nodes can initiate this switch message then what will happen, let us see. However, more than one node can initiate the switch message. An initiator A sends an acknowledgment to another initiative B only if B’s node id is smaller than A’s. I am repeating again, so if it is the case that is the more than one node can initiate this switch mode, switch message then what will happen? An initiator A, it may send an acknowledgement to another initiator B because now two nodes they are what say trying to send message.

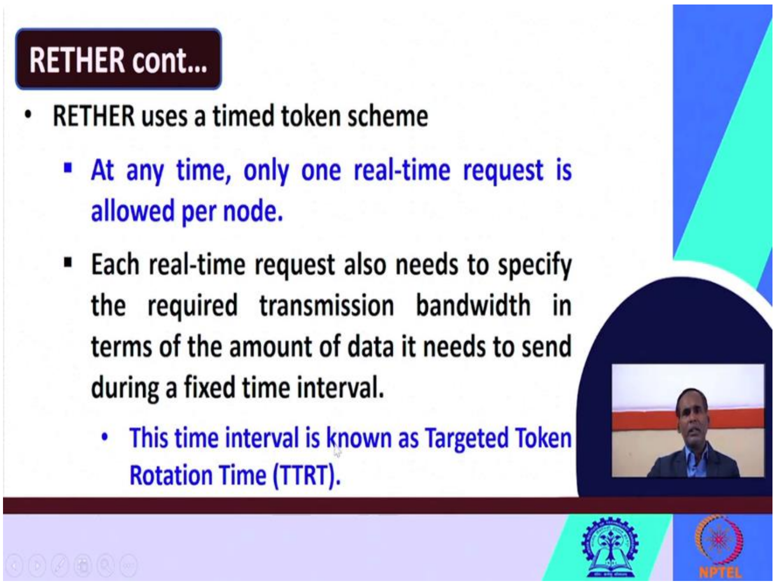
An initiator A sends an acknowledgement to the other initiator B only if this B's node id is smaller than the node id of A. Then only a will send an acknowledgment to the initiator. So, in this way, this type of confusion or this type of ambiguity can be overcome. So, now finally, only one initiator will be able or only one node will be able to transmit the message. There is another possibility that in case of a loss of acknowledgment, so some of the nodes while they are sending the acknowledgments, some of the acknowledgments may be lost during transmission.

So, in case of loss of acknowledgment what is the possibility the initiator times out. In case of loss of acknowledgement, the initiator it times out and the initiator retries for a fixed number of tries. So, these how many attempts you can take?

So, it is supposed fixed at ten, so like that or whenever initial times out again you will try, again you will try so, it may be three, four, five or ten whatever that it is a fixed. So, whenever in case of loss of acknowledgement, the initiator times out then the initiator retries for a fixed number of tries.

Suppose, let us fix it three then the initiator retries for sending the acknowledgement for three number of tries and if he does not get anything else any response then you say that then it will assume that the nodes are dead. Then it will if it does not get any response even after sending the fixed number of tries maybe three number of tries then it will assume that the nodes are dead.

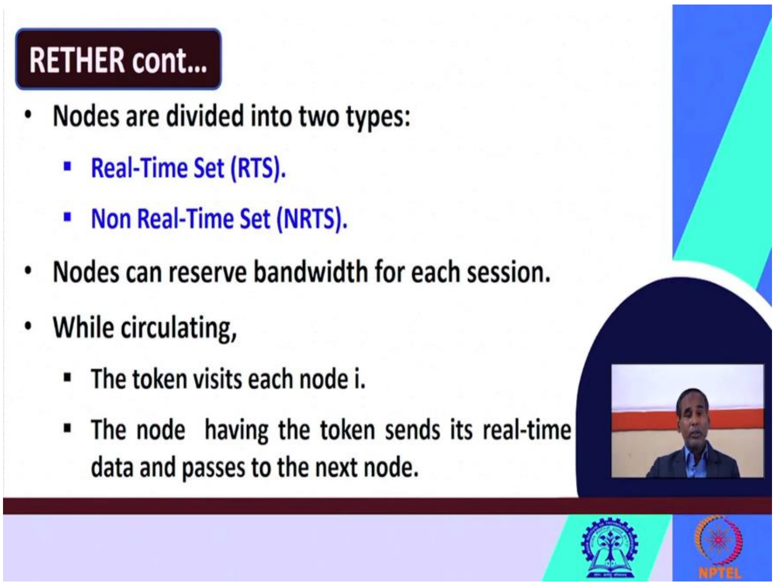
(Refer Slide Time: 25:59)



RETHER uses a time token scheme; this protocol RETHER uses a time token scheme at any time only one real time request allowed for node. Please remember, RETHER it uses a time token scheme and at any point of time only one real time request is allowed to be transmitted for a node. Each real time request, it also needs to specify the required transmission bandwidth. The real time request also it has to specify the required transmission bandwidth, in terms of what? In terms of the amount of data it needs to transmit during a fixed interval time.

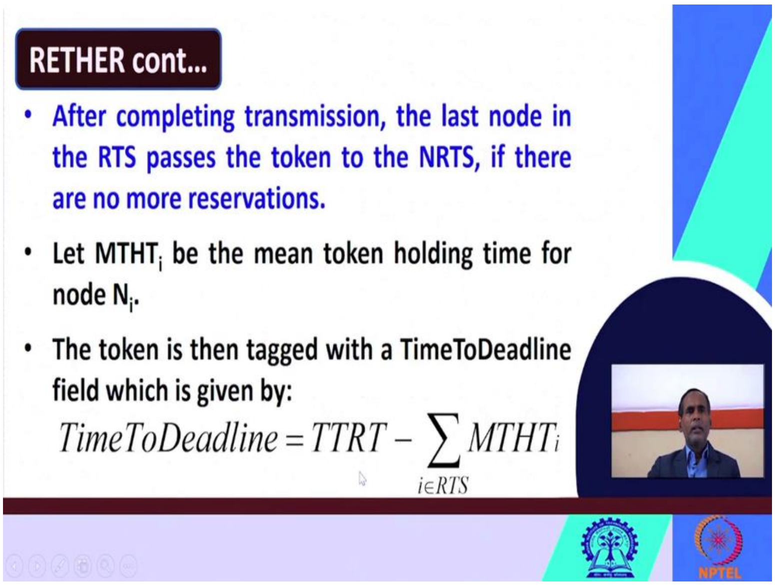
And, this is nothing but the TTRT or targeted toke rotation time. Each real time request, it also needs to specify the required transmission bandwidth in terms of what, in terms of the amount of data it needs to send during a fixed time interval during a fixed time period. And this time interval is known as what TTRT, this is also known as TTRT or targeted token notation time TTRT. What is TTRT, we have already known earlier.

(Refer Slide Time: 27:02)



In RETHER, the nodes are divided into two types, one is real time set or RTS another is non real time set or NRTS. The nodes can reserve bandwidth for each session, the nodes they can reserve the bandwidth for each session. While circulating the token visits each node i. So, when the token, it is circulated it visits each node i, while circulating the node the token visits each node i. Then the node having the token it sends its real time data and passes to the next node. So, we have already told you, the node can send something only when it is having, it acquires the token. So, the node having the token, it sends its real time data and it passes to the next node.

(Refer Slide Time: 27:47)

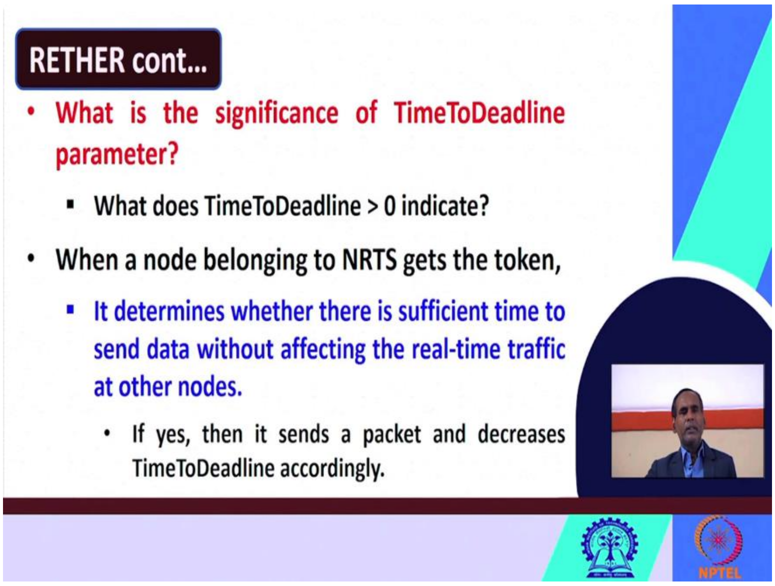


Then what will happen? After completing the transmission, the last node in the RTS passes the token to the NRTS, after completing. So, I have already told you the node having the token it will send its real time data and passes to the next node. After completion of this transmission, then the last node in the RTS set, what it will do? It will pass the token to the set NRTS, if there are no more reservation.

So, if there are no more reservations, then it will after completion of the transmission, the last node in the RTS set, it will pass to what? It will pass the token to the NRTS, what a set. Let MTHTi, HTi be the main token holding time for node Ni. Let us assume that for node Ni, MTHTi represents the main token holding time then the token is then tagged with another field called us time to the deadline field which is given by the following formula.

The token is then tagged with a field called as time to deadline field which is given by the following equation that is time to deadline is equal to. How to compute it? The value of TTRT minus summation of MTHTi, i belongs to RTS. What is MTHTi? MTHTi is the mean token holding time for node i. So, how many such nodes are there? Take the summation of the MTHTi values for all the i's. So, then take out the summation then subtract it from TTRT, what it will get that is known as time to deadline. In this way compute this time to deadline and then tag this time to deadline field with this token.

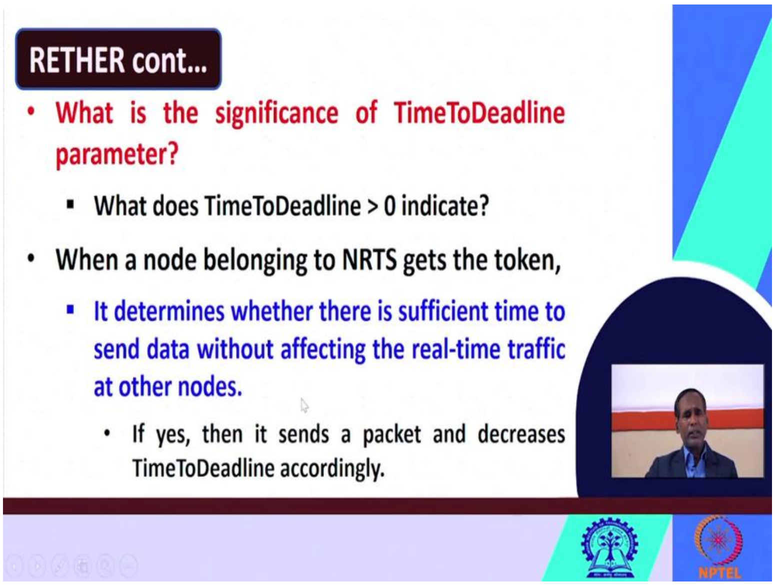
(Refer Slide Time: 29:35)



Now what is the significance of time to deadline perimeter? Why do you compute? What does the time to deadline? If the time to deadline is greater than 0, what does it indicate? So, when a node belonging to NRTS gets the token, when any node belonging to this NRTS set, it gets the token, what does it do? It determines whether that is sufficient time to send data without affecting the real time traffic at other nodes.

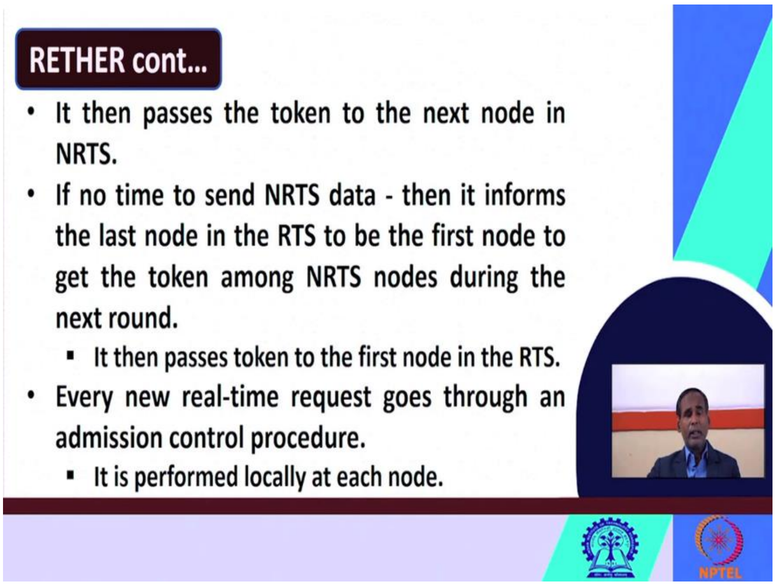
When a node it is belonging to NRTS gets this token what it will do? It determines whether there is sufficient amount of time is there to send the data without affecting the real time traffic, without affecting the real time messages at other nodes. If it is, yes, then only it will send a packet and then it will decrease this time to deadline accordingly. So, this is the significance of time to deadline parameter.

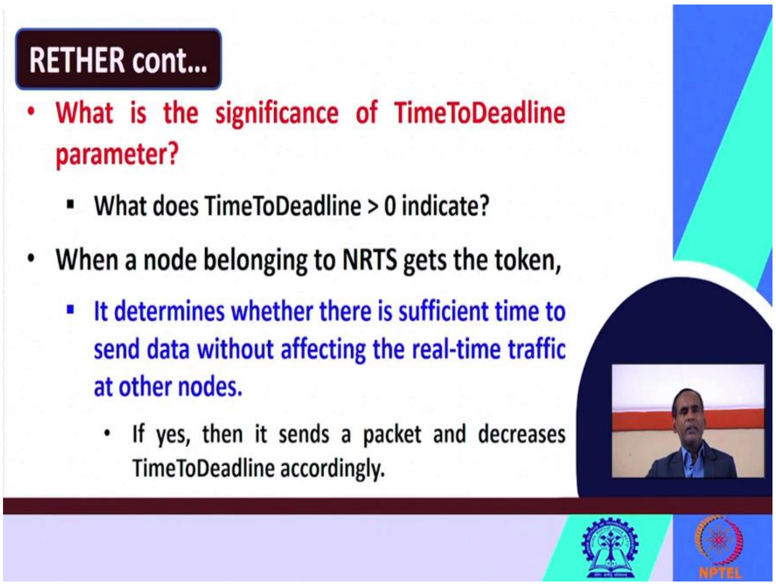
(Refer Slide Time: 30:34)



So, now, it will see I have already told you, if this will be yes that means, if sufficient time is there to send the data without disturbing the real time what the traffics at other nodes. If it is yes then it sends a packet and this time to deadline accordingly.

(Refer Slide Time: 30:51)





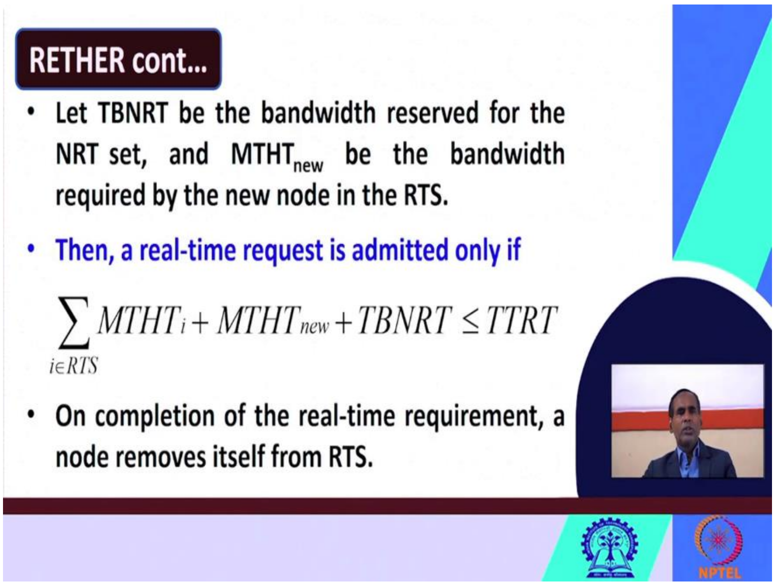
Then it passes the token to what, to whom it will pass the token? Then it passes the token to the next node in this NRTS set. Now, there are two possibilities, if time is there then it will send the packet and decrease the time to the deadline accordingly. And then it will pass the token to the next node in NRTS.

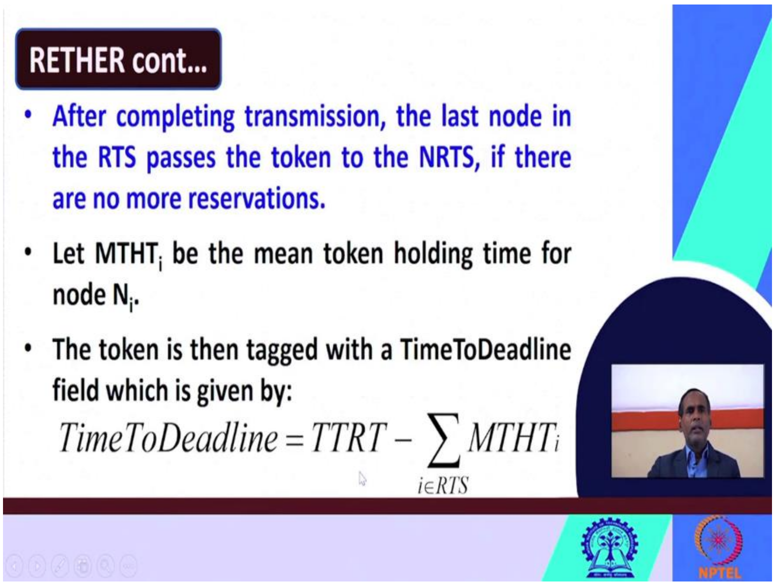
But if no time is there if this is yes part, if there is no time, if no time to send NRTS data then what will happen? It informs the last node in the RTS set to be the first node to get the token among NRTS nodes during the next round. So, if yes, we have already seen that it can send the packet and decrease the time to deadline accordingly.

Then it will pass the token to the next node in NRTS, if no time to send NRTS data is available, then it informs the last node where it informs the last node in the RTS set to be the first node to get the token among the NRTS nodes during the next round. It then passes the token to the first node in the RTS, then it passes the token to the first node, where? In the RTS.

Every new real time request goes through an admission control procedure. So, whenever any real time request comes, it passes through an admission control procedure which is performed locally at each node. Each real time, each new real time request when it comes, it goes through an admission control procedure and this is performed locally at every node.

(Refer Slide Time: 32:24)





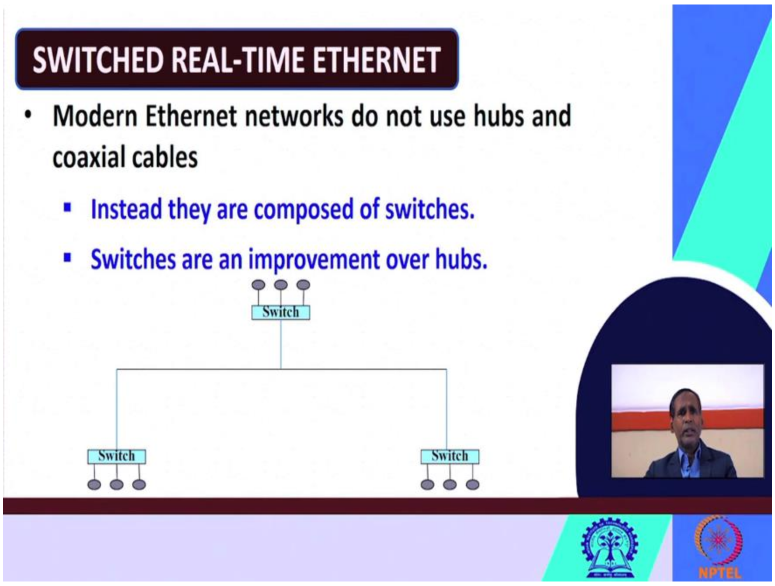
So, let TBNRT be the bandwidth reserved for the NRTS set and MTHT new with the bandwidth which is required by the new node in the RTS set. Then a real time request can be admitted only if the following condition is satisfied. What is the condition? That is $\sum_{i \epsilon RTS} MTHTi + MTHT_{new} + TBNRT \leq TTRT$.

So, if this word sum is less than or equal to TTRT then only a real time request can be admitted else not. So, on completion of the real time requirement, a node removed itself from the RTS. So, on completion of this, on completion of the real time requirement in node remove itself from the RTS.

So, in this equation all the terms you have already known, TTRT you have known earlier, TBNRT I have told the bandwidth result for the NRTS set. MTHT you have already known we have known MTHT, MTHT I have already told you, what is MTHT, we have already told MTHT be the mean token holding time for node Ni and similarly MTHT new be the bandwidth required by the new node.

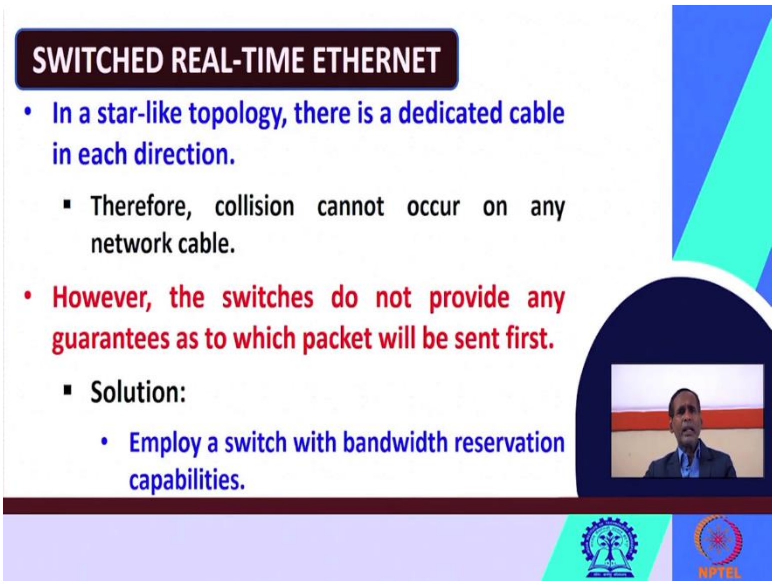
So, all those terminologies you have known. So, if this sum is less than equal to TTRT then a real time request can be admitted else not. On completion of the real time requirement, a node removes itself from the RTS.

(Refer Slide Time: 33:56)



Then quickly look what this switch real time Ethernet I have already told you the current age Ethernet networks that do not use these hubs and coaxial cables rather they are composed of switches. And switches they are the improvement over hubs if it to some extent these switches can be integrated to the hubs. So, this is how when the switches are connected, they may look like this.

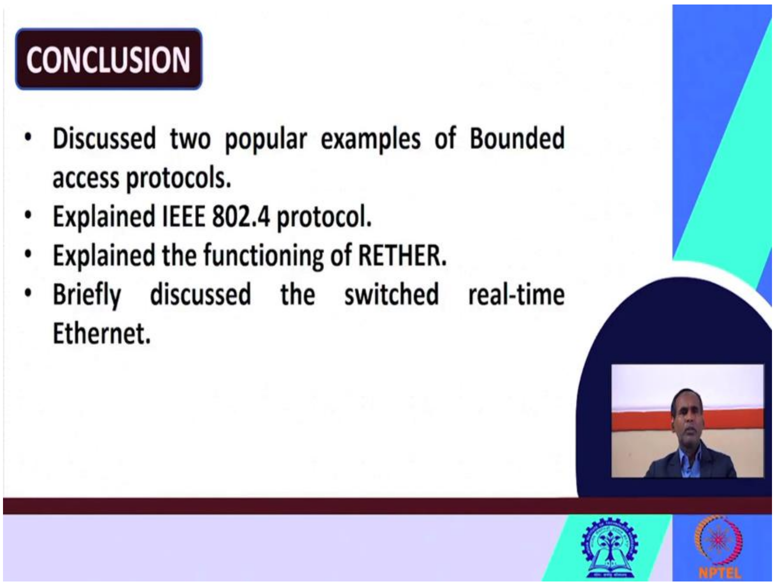
(Refer Slide Time: 34:21)

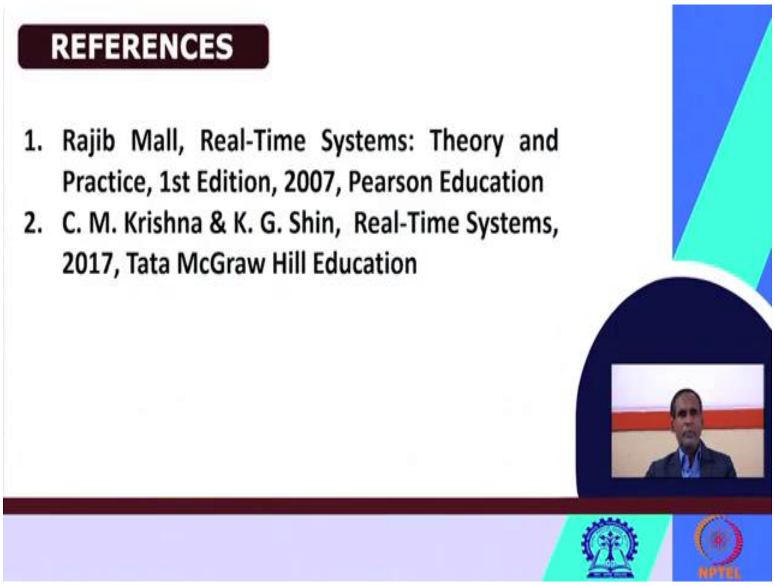


In a star like topology, there is a dedicated cable in each direction. So, if you are using a star like topology, in a star like topology there is a dedicated cable in each direction. Therefore, collision cannot occur on any network cable. So, if you are using this star like topology, since there is a dedicated cable in each direction, so therefore collision cannot occur on any network cable. However, the switches do not provide any guarantees as to which packet will be sent first. This is one of the drawbacks of this switches.

However, the switches they do not provide any guarantees as to which packet will be sent first that is not given by the switches. Switches cannot provide this guarantee. So, to overcome this problem a solution is there, let us see the solution. The solution is as follows. To overcome this problem, you may employ a switch with the bandwidth reservation capability. So, if you can employ, if you can use this switch with or which has bandwidth reservation capabilities, then this problem can be overcome. This is how we have seen little bit basics of the switched real time Ethernet.

(Refer Slide Time: 35:27)





Today we have discussed two popular examples of bounded access protocols IEEE 802.4 and RETHER. We have also briefly discussed the switched real time Ethernet. We have taken from these books. Thank you very much.