Real-Time Systems

Prof. Dr. Rajib Mall

Department of Computer Science and Engineering

Indian Institute of Technology, Kharagpur

Module No. # 01

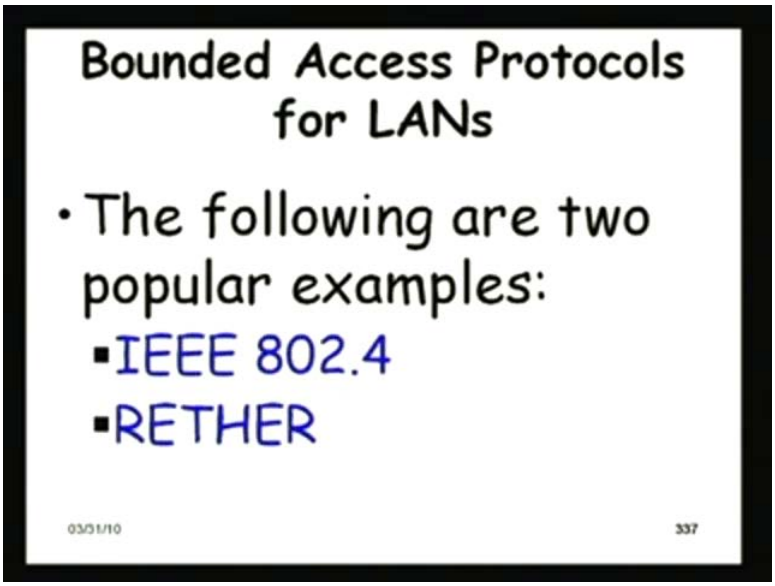
Lecture No. # 35

Real-Time Communication in a LAN (Contd.)

ok Good morning; so, just last class we were discussing about real-time communication in a LAN environment and we could not complete our discussion last time. So, let us continue from where we left last time.

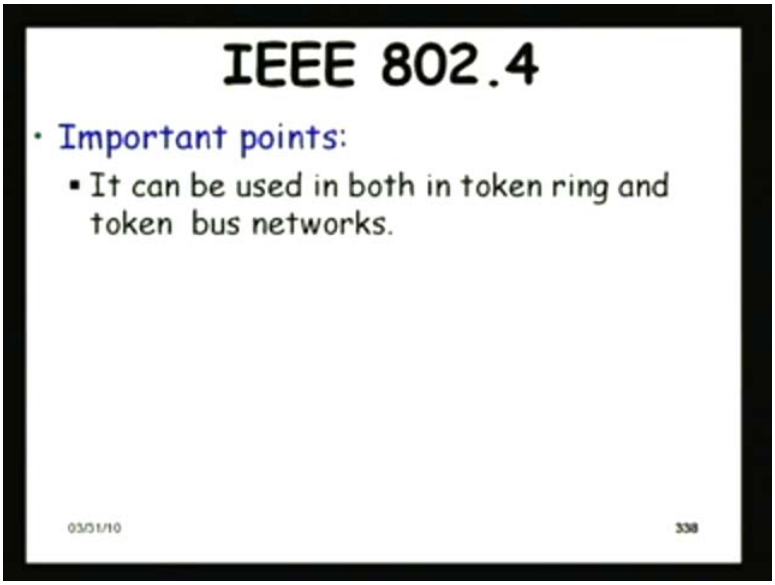
(Refer Slide Time: 00: 34)

(Refer Slide Time: 00: 42)



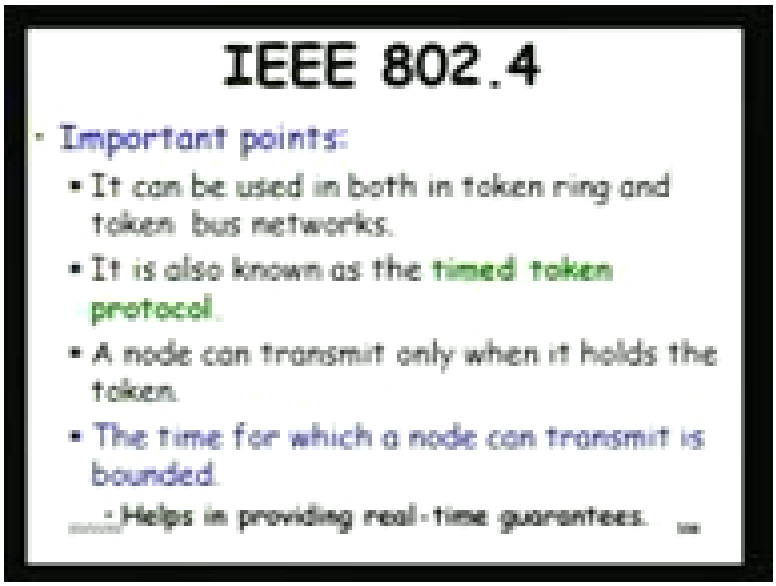
So, last time we were discussing about bounded access protocols for LAN and 2 popular examples of bounded access protocols are I triple E 802 point 4 and Rether. First, let us discuss about the I triple E 802 point 4 protocol.

(Refer Slide Time: 01: 00)



Some important points in the working of the protocol are that, it is a protocol that is used either in a token ring or a token bus network and we were seeing that, the token ring token bus network have a special advantage in the manufacturing situations; have been in use in manufacturing automation for quite some time. The 802 point 4 is also popularly known as the timed token protocol. Here, a node is allowed to transmit only when it is in position of the token.

(Refer Slide Time: 01: 40)



So, the token visits every node and only when a node holds the token it can transmit and once it starts transmitting then, the duration for which it can transmit is bounded. So, that is fixed apriori and it is not necessary that every node transmits for similar amounts of time. Depending on the traffic at a node, the time that a node is permitted to transmit can vary but, still once it has been assigned we know that which node is going to transmit maximum for what duration and based on that, we can do some computation; to find out what will be the maximum delay before which a node will be allowed to transmit. So, we can compute the maximum priority inversion time and which is very important in real-time applications. Because, based on that we can design a system. If we need the priority inversions in message transmissions to be less than some amount we can design the network and then, based on that guarantees can be given. But, we had seen that in some networks it is very difficult to guarantee the bound and the priority inversion.

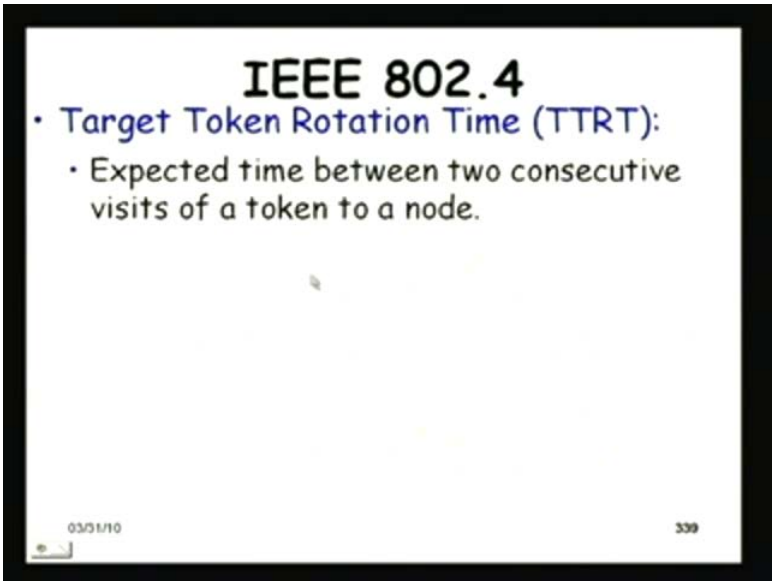
Sir

Yes

802 actually, 822.5 we had already used…

Yes, 822.5 we had already discussed; that was a priority based protocol. This is a bounded access protocol. So, here we,…the time taken to transmit it, is pre-decided, transmitted one after another; whereas, in a I triple E 822.5 which is a priority based protocol the different nodes bid or they there is arbitration; there is a reservation and a mode bit; all those we had discussed in the last class.

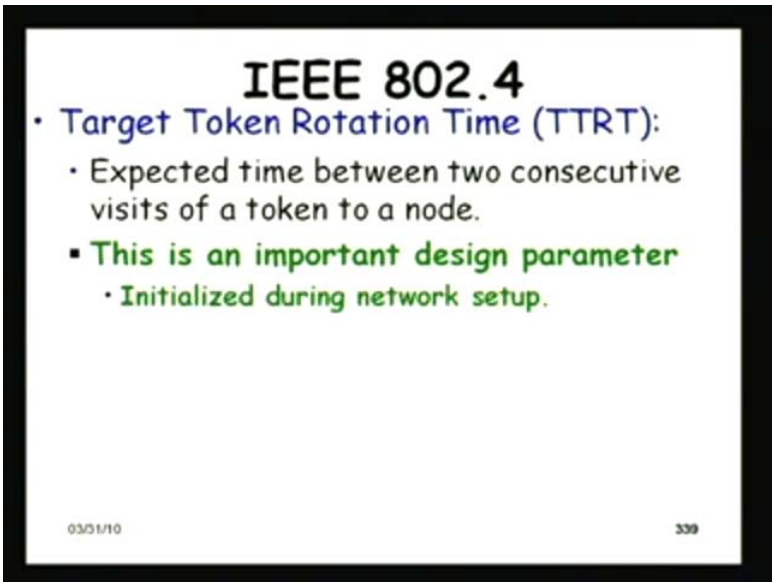
(Refer Slide Time: 03: 33)



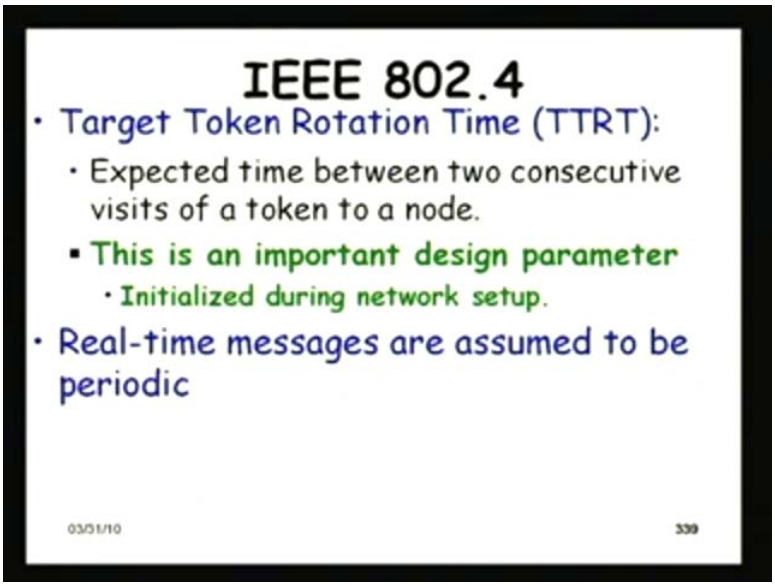
That each time a node gets a message it tries to capture the token by writing the priority of the message on the reservation field and then the token is set into the free mode so that is about 802 point 5 but, here it is not a priority based protocol; it is not a global priority protocol; but it is a bounded access protocol. So, here one important parameter is the target token rotation time (Refer Slide Time: 04: 51) or T T R T. The T T R T is the expected time between 2 consecutive visits of a token to a node (Refer Slide Time: 05: 00). So, it is not the exact time the 2, of 2 visits of a token to a node to the expected time. It can be less than this or more than this. We will see the situations in which the actual token rotation time can be less than T T R T; it can even be more than T T R T; but, T T R T is the expected time between 2 consecutive visits of a token to a

node. And, this happens to be the most important design parameter. We will,… based on some situations that nodes have certain type of messages, we will try to design a I triple E 802 point 4 network where we would first try to fix the T T R T- most important design parameter and this is initialized during that network setup (Refer Slide Time: 05: 52).

(Refer Slide Time: 05: 32)

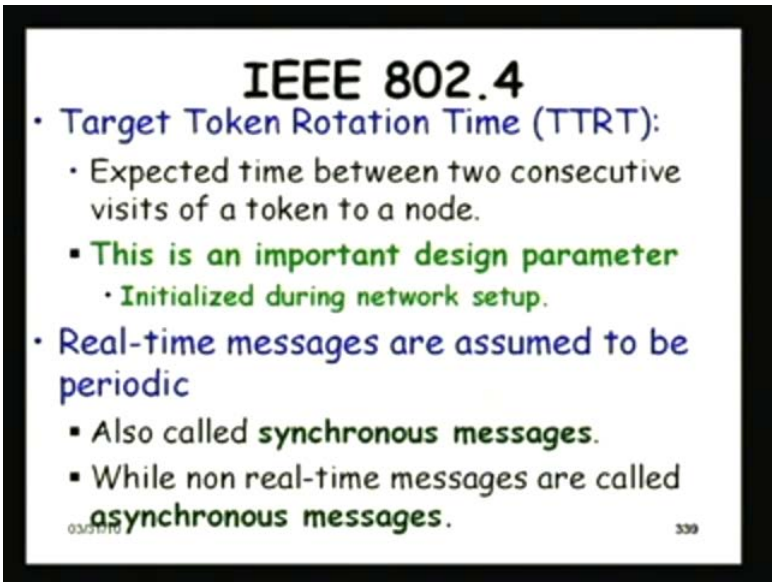


(Refer Slide Time: 06: 00)



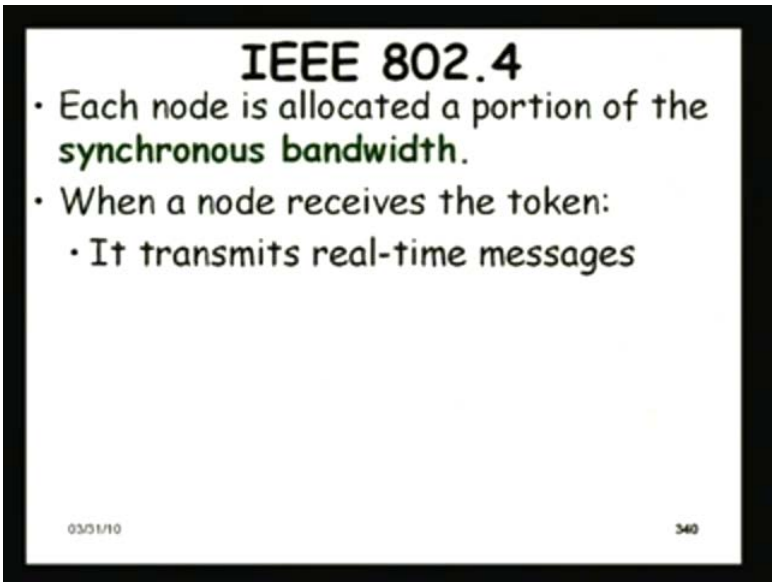
Here, 2 categories of messages are handled. Real-time messages are assumed to be periodic. So, if we have some aperiodic or sporadic messages which are important, we will have to convert them into real-time, into periodic messages where there will be a slot reserved for them each time and they might or might not arise like that. So, all the real-time messages are assumed to be periodic and these are referred to as the synchronous messages.

(Refer Slide Time: 06: 30)



So, the synchronous messages are the real-time messages which are periodic and when synchronous messages do not exist with a node for some token visit they can start transmitting the other messages that they might have the non real-time messages which are called as the asynchronous messages. So the transmit the synchronous messages first and if there is time they would transmit the asynchronous messages and we will see that sometimes the token might arrive at a node early, less than the T T R T time and then that is the opportunity to transmit the asynchronous messages.

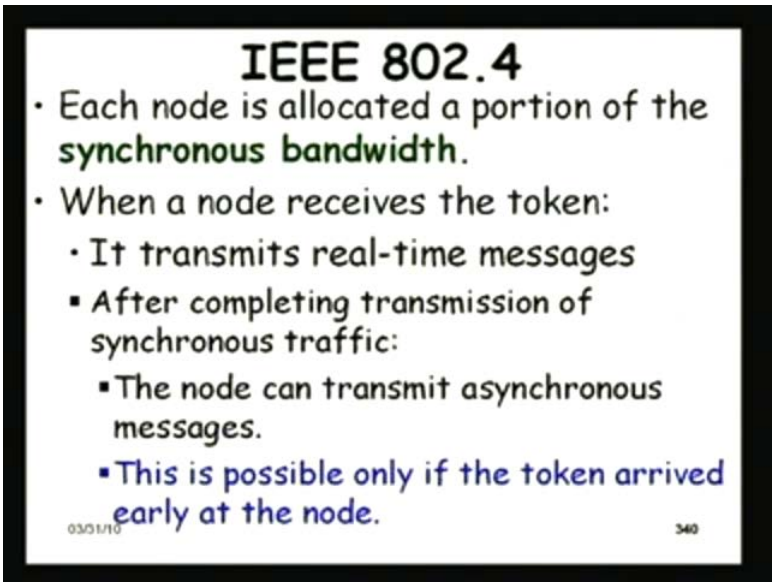
(Refer Slide Time: 07: 19)



So, the T T R T is the time it takes for the token; the expected time of the token to visit each node and the T T R T is the total bandwidth available I mean, we can say that in one turn all the nodes together transmit for T T R T, right? So, the T T R T duration all the turns all the nodes take turn and when each one transmit once for the turn to complete that is equal to T T R T(Refer Slide Time: 08: 00), right.

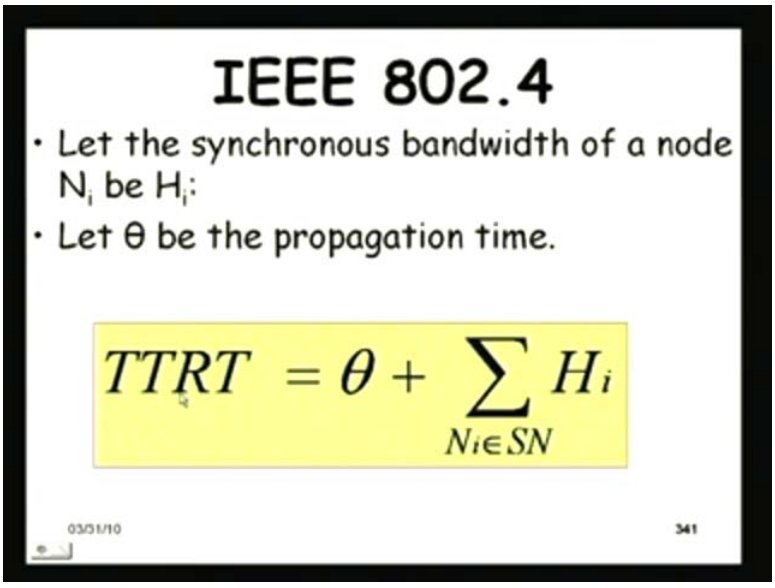
Now, the T T R T is distributed between the different nodes so some node might get T T R T by 10 some node T T R T by 5. So, that is the duration for which they will be allowed to transmit (Refer Slide Time: 08: 20). Now, when a node receives a token it starts to transmit its real-time messages and after transmitting these messages if there is any time left out they can transmit the asynchronous messages.

(Refer Slide Time: 08: 37)



This we will see is that usually, possible if the token arrives early at a node. A token might arrive early at a node because some node has released the node, the token early some nodes node or nodes have released it early because they had possibly (no audio 09: 02 to 09: 30) visit. Let it be h i. So, for node N i each time the token visits it can transmit for a maximum of H i and let theta with the propagation time in the network.

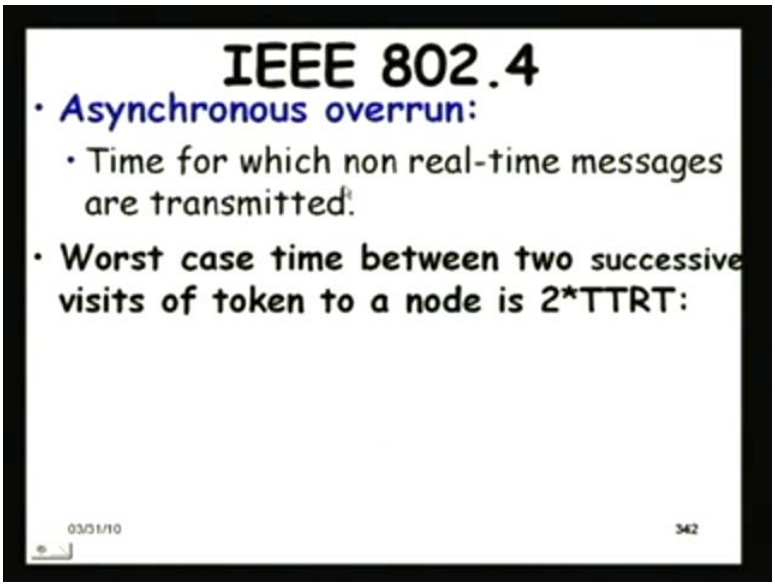
(Refer Slide Time: 09: 45)



So, then we have this expression the target token rotation time the T T R T is equal to the propagation time plus the times for which all the nodes transmit, right? It looks very simple here. The time for which all the nodes transmit plus the token rotation, sorry, the propagation time is equal to the target token rotation time; this is the expected value of the token to arrive 2 consecutive visits to a node. Is this appearing ok? See, very simple T T R T is theta plus all the synchronous bandwidths for every node. Sorry (( )).

Yes, n is all the nodes the nodes at let us say the set of all nodes that are there in the network ok? So, for all the nodes that are there in the network which share this bandwidth the sum of their synchronous bandwidths plus the propagation time will be equal to T T R T. Now, once we have allocated the synchronous bandwidth to the nodes, we have to be aware that there can be asynchronous overruns. Here, the token arrives early and a node starts to transmit non real-time messages. So, that we will call as the asynchronous overrun where the token arrives early or may be the node does not have itself real-time messages to transmit for its entire allocated duration and it is starts to transmit non real-time messages that we will call as the asynchronous overrun. Now, given this situation the worst case time between 2 successive visits of a token is 2 into T T R T. We had said that the expected time of a token visit 2 successive visit of a token is T T R T but, due to asynchronous over run it can be 2 into T T R T; why is that? Because, in the worst case let us assume that none of the nodes had anything to transmit either the synchronous neither the asynchronous messages in the worst case. So, then it arrives at a node almost one T T R T time early right.

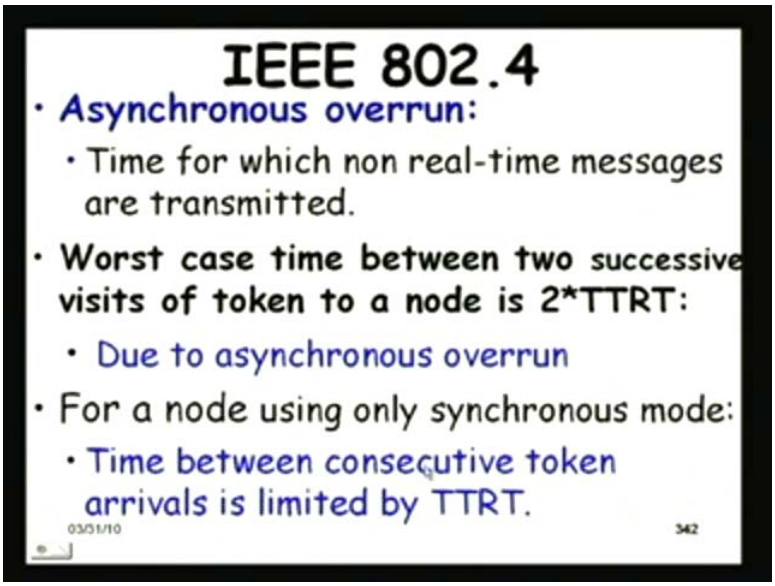
(Refer Slide Time: 11: 00)



T T R T minus time minus theta time early, alright? Almost T T R T time early it can arrive at a node and now suppose, in the next iteration, every node had synchronous and asynchronous messages to transmit and they keep on transmitting, so, then the time for the token to visit a node will be 2 into T T R T. does that appear alright? Ok, we will just check that again with a diagram; but, for the time being let us just assume that. Then, the worst case time between 2 successive visits of a token is 2 into T T R T.

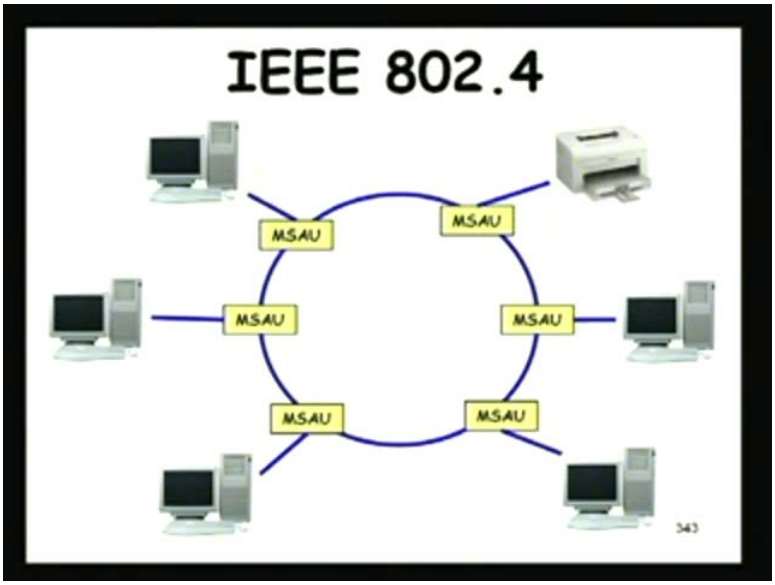
Now, again this is due to the asynchronous overrun because the synchronous messages they are their size is fixed so the asynchronous messages when need to transmit them when the token arrives early that leads to this 2 into T T R T. But, when in a network we have only the synchronous messages then of course, the worst case time between consecutive token arrivals will be T T R T, right?

(Refer Slide Time: 13: 36)



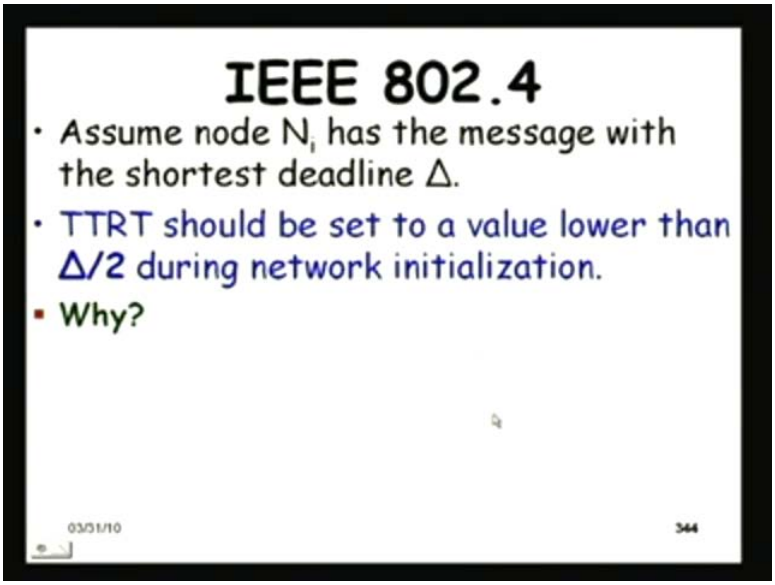
Assume that there are no asynchronous messages only the real-time messages then, it will be T T R T. So, it becomes 2 into T T R T when we have asynchronous messages in the node in for the nodes to transmit. Do you agree with that, yes or no? Yes, it is a simple thing.

(Refer Slide Time: 14: 19)



So, this is just an example suppose the token once it lift this node none of these had any synchronous or asynchronous messages to transmit so within theta it will arrive back here right? It has arrived early; almost T T R T theta is very small. So, it has arrived almost T T R T time earlier. Now, let us say this node and this node (Refer Slide Time: 14:51), each one had both synchronous and asynchronous messages to transmit lot of asynchronous messages to transmit. Since the token has arrived early they will start transmitting the asynchronous messages and then also the asynchronous messages. So, let us say the synchronous messages are transmitted for T T R T and asynchronous messages for another T T R T. So, for the next visit of the token to this node it will be 2 into T T R T

(Refer Slide Time: 15: 29)

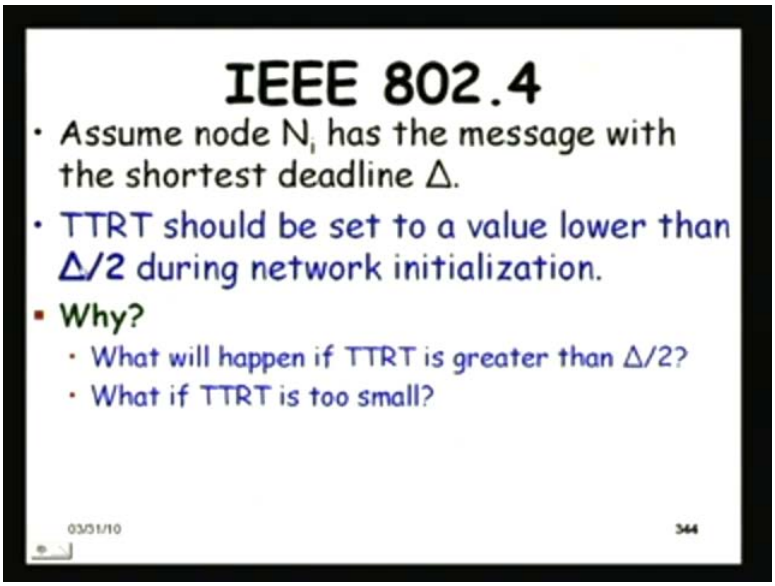


Now, let us look at, how do we design given a situation, where nodes have certain messages. How do we design the network? Now, let us assume that each, we know, what are the synchronous messages the real-time messages with every node. Now, let us assume that of all the nodes, the node N i has the message with the shortest deadline the other nodes have messages where the deadlines are more relaxed (Refer Slide Time: 16: 00) - the larger and delta is the shortest deadline and node N i happens to have that message.

Now, we need to set the T T R T because that is the first parameter. We need to fix so the T T R T should be set to be either delta by 2 or lower than that; why is that? (( )) yeah, because a node

the token can arrive at a node in the worst case 2 into T T R T time later. So, it can arrive delta time later and then it start transmitting. If we keep T T R T greater than delta by 2 then, by the time in the worst case the node arrives the deadline would have already expired now.

(Refer Slide Time: 17: 00)

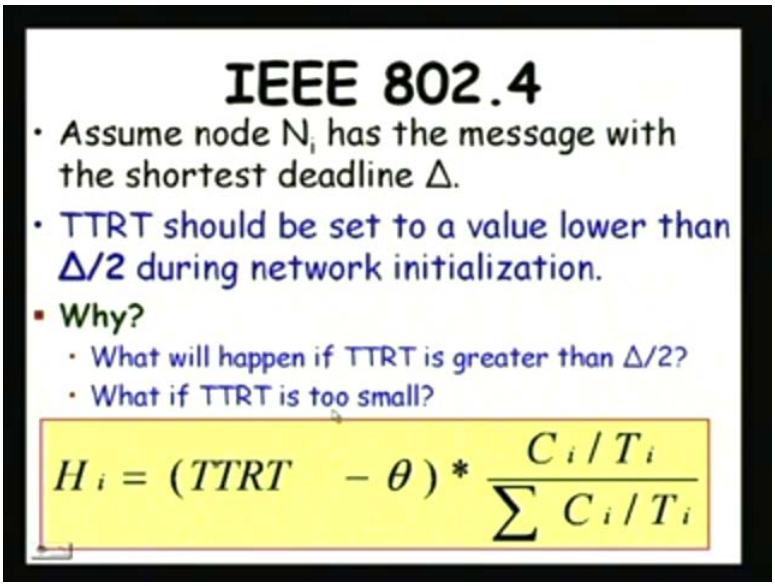


What if the T T R T is set too small, let us say delta by 10? What do you think will it be a good idea to set it to be delta by 10 or delta by 100, (( ))? No, it will transmit to the next digit ok. So, you are saying that a real-time message.

(( )) ok, a real-time message it may not be able to transmit in that duration but that is not the major problem because again it will visit after delta by 10, right?

So, the main problem is that there will be too much of overhead here. Each time theta is lost out is not in propagation. So, the node is rotating and sorry, the token is rotating and each time only very few transmissions is taking place and more time is wasted in just propagating across the network. So, the token is propagating more 10 times and the transmission is only few right so the overhead will become too much. And, also as he was pointing out, that in each visit it may not be possible to transmit all the real-time messages that it might have. So, it is possible that if the entire synchronous messages has the bandwidth which is larger than delta by 10 then there will be problem, is it not?

(Refer Slide Time: 18: 48)

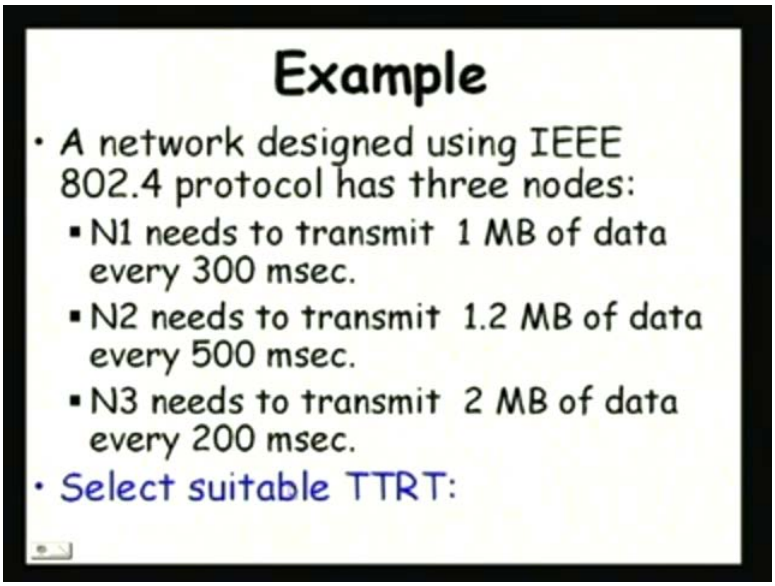


So, having known that we have to fix the T T R T to be equal to delta by 2 if possible or slightly lower than delta by 2, never higher than delta by 2 but, either equal to or slightly lower than delta by 2. Now, the next thing we will have to decide is, for how much time each node can transmit. So, that is given by this formula here (Refer Slide Time: 19:23). See here, the total time that a node can transmit in one T T R T that is during one visit of the token is T T R T minus theta. This is the total time for which all the nodes can transmit right and now let us assume that each node has messages that keep on arriving T i. T i is the period of the messages and C i is the size of the messages means C i by T i is the utilization due to the message i. So, C i is the messages for a node i size of the messages for node i and T i is the period then C i is C i by T i is the total utilization of the channel due to the node i, right. So, divided by the total utilization for all the node, you have proportionally distributed the available bandwidth among the nodes depending on what is the utilization they are going to achieve; is that ok?

So, the total band width that is T T R T minus theta for which the transmissions can occur during one visit of the token is proportionally distributed among the nodes depending on how much traffic they have. Now, let us try to do a small example let us try to consider a network which is designed using I triple E 802 point 4 protocol it has 3 nodes; node n 1 needs to transmit 1 MB data every 300 milliseconds; n 2 needs to transmit 1 point 2 MB data every 500 millisecond and

n 3 needs to transmit 2 MB data every 200 millisecond. So, the first thing is we will have to select a suitable T T R T. T T R T would be very simple to select. What do you think will be T T R T? Yes, what would be T T R T? (( )).

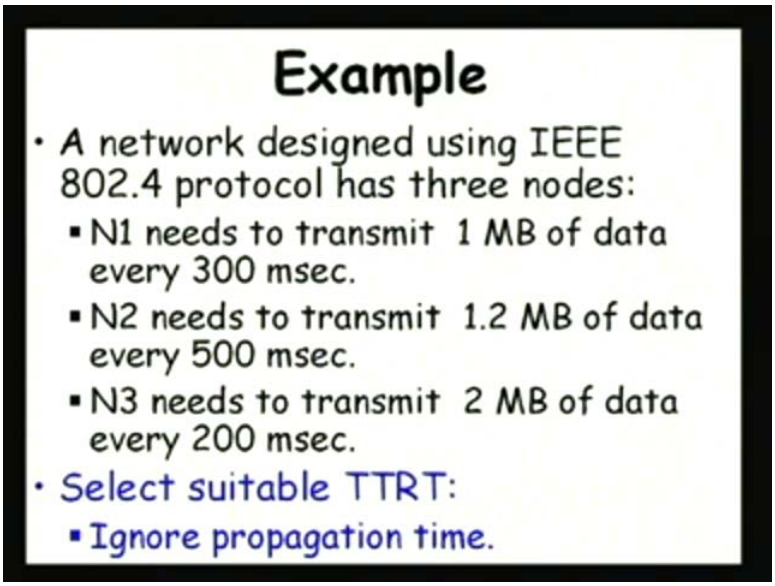
(Refer Slide Time: 21: 15)



So, what will be in this case? For this example look at the example, so these are the situation given 3 nodes are there: n 2, n 1, and n 3 and the size of messages and over what duration they need to transmit is given (( )).

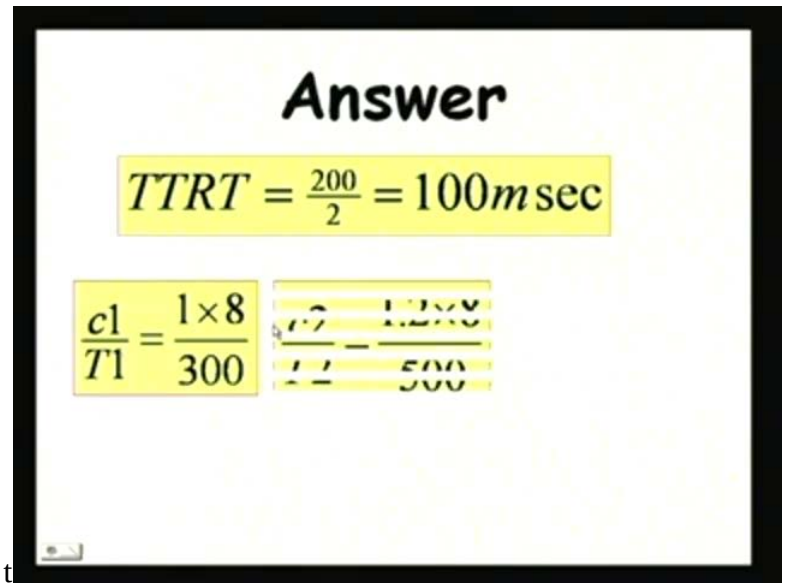
Now, what is their value, tell me? Here, 100; 100, yes so we should set T T R T to be equal to 100. Do everybody agree with that? Is it 200 by 2? Exactly, so delta by 2; delta have seen, the node n 3 has messages with the shortest deadline right? So, we need to set delta is equal to 200 by 2, sorry, T T R T is equal to 200 by 2, ignoring the propagation time, ok.

(Refer Slide Time: 23: 00)



Now, what about the synchronous bandwidth? How do we allocate the synchronous bandwidth? Yes, how can we allocate the synchronous bandwidth? Just tell, how do we allocate (( )). We have to find the utilization due to each node - the channel utilization due to each node or how much traffic is there in each node per unit time and then, based on that we need to proportionally distribute the T T R T between all the nodes, right?

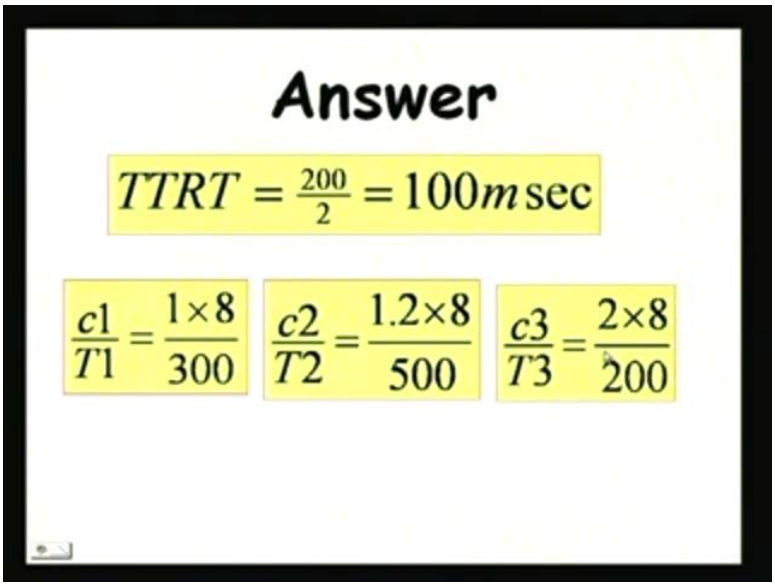
(Refer Slide Time: 23: 45)



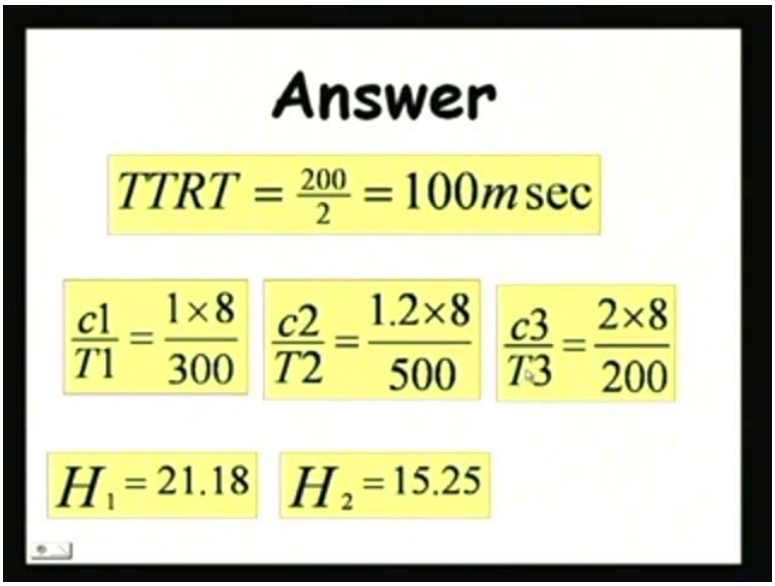
t

So, let us do that; T T R T is 200 by 2 is 100 millisecond. These we have already said; now the utilization due to node one will be one into 8 by 3 100 MB per millisecond right? For node 2 it will be 1 point 2 into 8 by 500 MB per millisecond because, it was byte I think 8 is multiplied here; so you have to convert it to bits sorry (( )) seconds. Seconds, is it ok? So, then it will be bits per second and c 3 by T 3 will be 2 into 8 by 2 100 bits per millisecond or second, whatever was given there.

(Refer Slide Time: 24: 40)



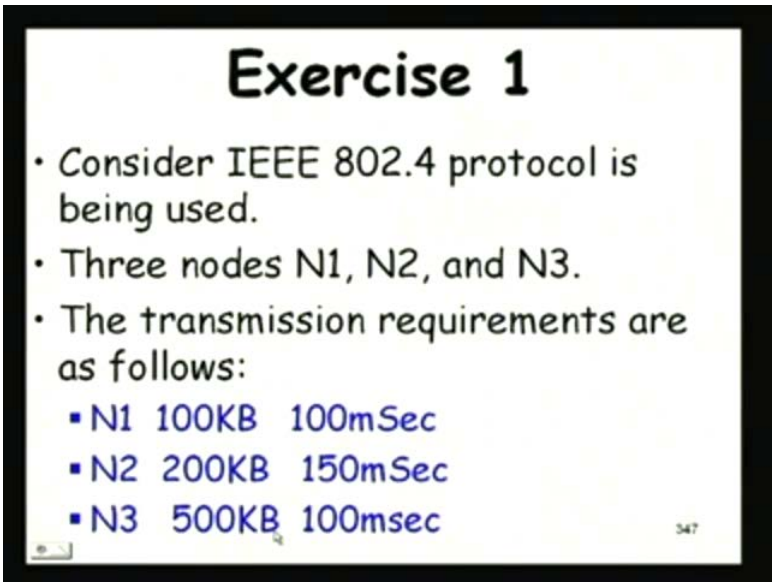
(Refer Slide Time: 24: 54)



Now, based on that we can find the total utilization due to all the nodes which is the sum of all this and then, we can proportionally distribute among the nodes. So, if the summation of all these 3 items (Refer Slide Time: 25:15) becomes let us say sigma, then T T R T 100 millisecond into 1 point 8 by 3 100 divided by sigma will be h 1 which is 21 point 18 millisecond. So, the node n one can transmit up to 21 point 1 8 millisecond h 2 that is the synchronous bandwidth of node 2 will be 15 point 2 five millisecond and n 3 the synchronous bandwidth for n 3 will be 63 point 56 millisecond does it appear ok? Anybody has any difficulty? Ok, all right so…(( )) yeah…

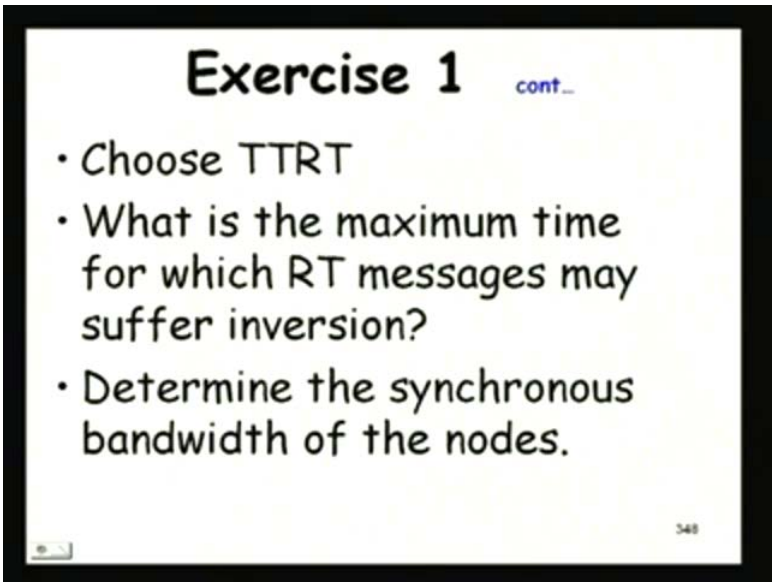
That is what is given in the problem then the propagation time can be ignored. (( )) yeah, so the propagation the network is small. So, theta we have ignored the theta is given then we will have to consider that so I hope if we give let us say, theta is equal to 1 millisecond the propagation time is 1 millisecond. You can easily do it also just a small modification on this right? This is another example; ok, this is an exercise for you to try so again 3 nodes n 1, n 2, n 3 and the transmission requirements of nodes is as follows:

(Refer Slide Time: 26: 00)



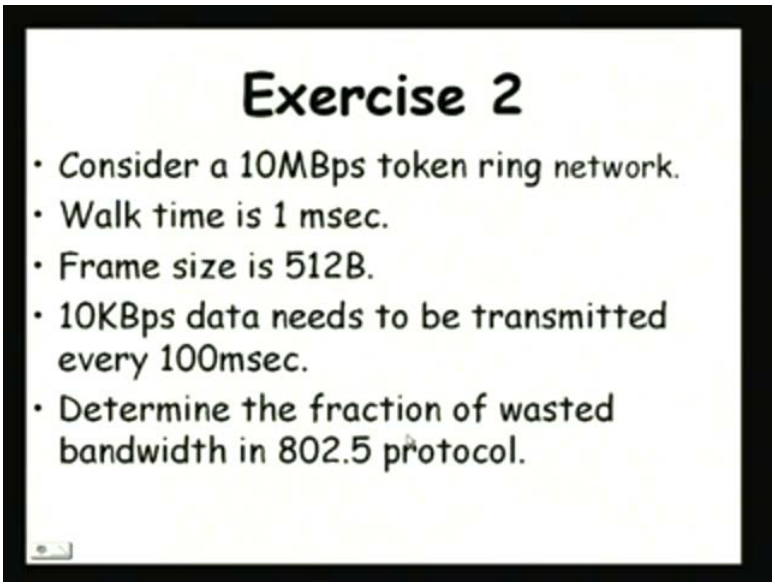
Node n 1 is 100 kilo bytes every 100 millisecond; node 2 200 kilo bytes every 150 millisecond and node n 3 500 kilo bytes every 100 millisecond. Please take that down; I will describe the other part of the problem. So, this is the situation that we were discussing that 802 point 4 is used and there are only 3 nodes in the network and the transmission requirements n one needs to transmit 100 kilobyte every 100 millisecond node 2 200 kilobytes every 150 millisecond and node n 3 500 kilobytes every 100 millisecond (Refer Slide Time: 27: 45). The first thing is we will have to choose the T T R T; how much will be the T T R T? it is 50, because 100 is the lowest deadline we can consider. So, 100 by 2 is 50; now what is the maximum time for which the real-time messages may suffer inversion? Yes, what do we think ? (( )) ok so, it is simple. The maximum time for which the messages suffer inversion is twice the T T R T that we had already discussed. So, T T R T is said to be 50; so the maximum inversion that is a higher priority message exists but, lower priority messages are getting transmitted that is limited to 100 2 into T T R T.

(Refer Slide Time: 28: 55)



Determine the synchronous bandwidth of the nodes; so this we will have to proportionally distribute the T T R T among the 3 nodes based on the utilization of the channel for each of this nodes; hopefully you can do that is it not? It is a simple thing. Now, let us look at another exercise so here it is given to be a 10 MBps token ring network

(Refer Slide Time: 29: 25)



The walk time is 1 millisecond; walk time is basically the token propagation time frame size is 500; 12 bytes 10 KBps. Data needs to be transmitted every 100 millisecond determine the fraction of wasted bandwidth in the 8 o 2 point 5 protocol (Refer Slide Time: 30: 00). So, this is a priority based protocol so here the data is that the total bandwidth of the network is given the propagation time is given the frame size is given so this is the size of the payload and a token and then the amount of data that needs to be transmitted every 100 millisecond is given and a fraction of the wasted bandwidth have to be identified. What is the fraction of the wasted bandwidth? Yes, how do we determine the fraction of the wasted bandwidth? Ok, how does waste occurs let us assume let us first discuss how does waste occur? Yes, can anybody tell how in 802 point 5 protocol bandwidth wastage occurs?

Sir, it may occur (( ))

yes

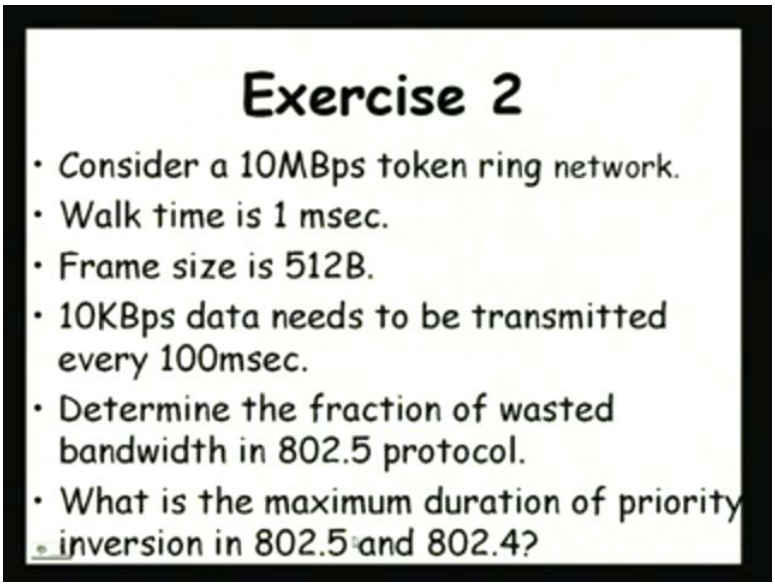
(( )) the time takes to travel from one node to other

ok So, basically what he says is that the time it takes to transmit a packet it is less than the propagation time, right. So, here a node does not transmit until it sees the head or back only then it knows that either reservation is made or not made so until it receives the header back a node does not start the next frame. So, let us say the transmission completed very fast because the bandwidth is 10 MBps and only 500 12 bytes needs to be transmitted right but it takes 1 millisecond for the token to the header to reach at the node back. So, if this time is more than the transmission time right then the node has already finished its transmission of the frame but it cannot transmit the next one waiting for the header of the token to arrive right. So, that is the wasted bandwidth; so the wasted bandwidth can be found out by determining theta minus the time it takes to transmit one frame. So, how much time it will take to transmit one frame (( )) this will be 512 divided by 10 to the power 6; right, 10 to the power 6.

So, 1 millisecond minus 512 by 10 to the power 6 will be the time that will be wasted and that waste occurs every frame of transmission, right. So, from that we can find out the fraction of wastage bandwidth in the protocol; does that appear ok? I mean, would you like to do and get the answer or you think that the computations are already… you can do it, ok.

So, if you understood that that we need to compute the time for the frame to be, one frame to be transmitted and check if it is more than 1 millisecond then, find out what portion of the time the node will be sitting idle without able to transmit the next frame. So, that is the wastage and that occurs during every frame transmission time so that is the fraction of the wasted bandwidth.

(Refer Slide Time: 34: 00)



But, what about this part of the question what is the maximum duration of priority inversion in 8 o 2 point 5 and 8 o 2 point 4 how do we do this (( )) time transmission

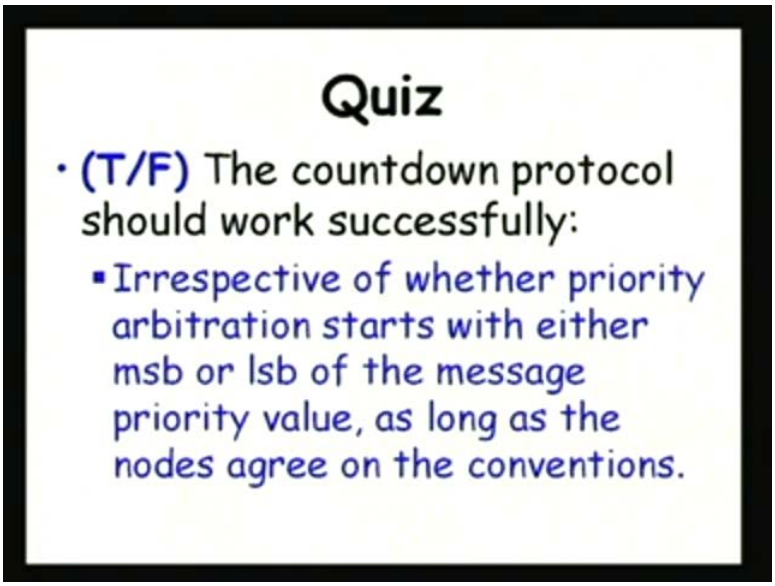
yeah here it will be 2 times theT T R T but what about this one?

(( ))

yeah

(( )) yes 800, for 800 2 point 5 it will be twice of either theta or whichever is larger. So, depending on the type of the network and our choice of T T R T we can find out which has the highest or more priority inversion possibility of higher priority inversion Now, what about this (Refer Slide Time: 35: 10) quiz? Please try this one; so, please identify whether the following statement is true or false. Now, we know the countdown protocol, right? We had discussed that; do you remember the countdown protocol?

(Refer Slide Time: 35: 40)



Here, there is arbitration period followed by transmission and the arbitration occurs by the nodes start transmitting their ID and when a node finds that it is there is a higher priority or a ID node with a higher ID then it just drops out and then we had discussed the example that we had taken that time that the priority arbitration start with transmitting the MSB that we had discussed the example that if you remember the nodes starts transmitting from its MSB most significant digit most significant bit and then they just check out which has higher. Now, the question is that, will it work? If they start from the LSB start transmitting from the LSB side and as long as the nodes all nodes agree the convention that each node its going to transmit from the LSB side will it work?

(( ))

No, what about let us say he says yes let us check others, would you…

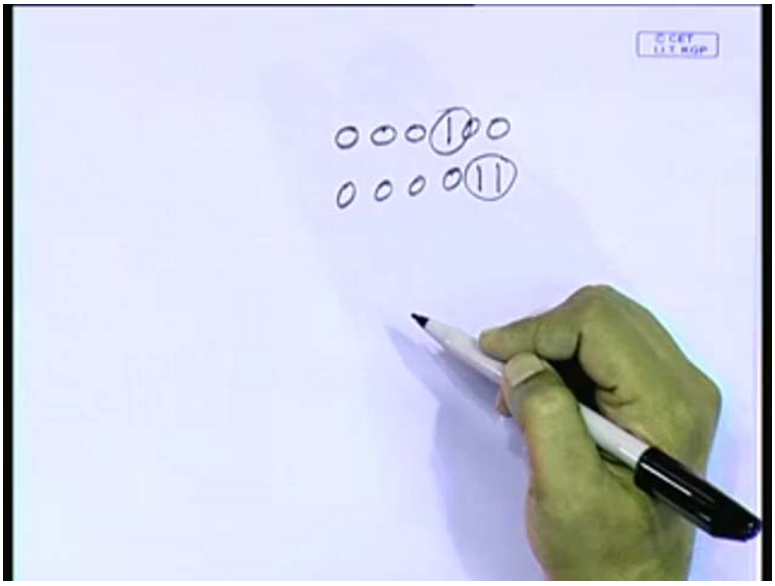
(( )) number of stations are limited (( ))

why will it work on number of stations are limited…

(( )) see each one transmitting the LSB, will it work what about others? Anyone else can answer; he said yes right, anybody saying no? Ok, it will not work actually since no one is coming

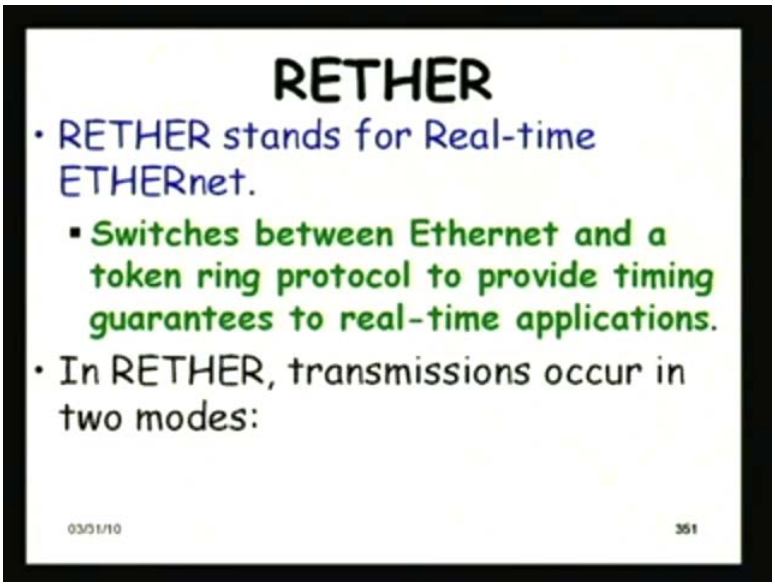
forward just look at this suppose a node has ID 0 0 1 0 0 and another one has 0 0 0 let us say 0 1 1.

(Refer Slide Time: 38: 00)



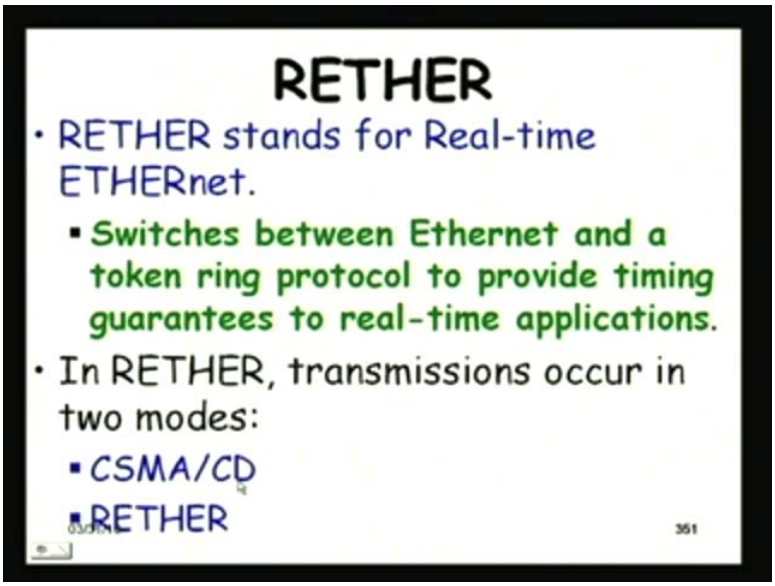
So, this is much lower than this one right; this node is lower than this node but, they cannot reason that out and you can just construct an example where let us say, if you look at another example like let us say 1 1 0 1 and another one has let us say 0 0 1 1, how do you reason about that? So, it becomes very difficult to reason the ID of a node by just looking from the LSB side. Ok, so you can always construct an example where it is very difficult to identify which one has higher or lower ID by just starting to examine from the LSB side it will not work. So, that is why you had given the example of MSB. Ok, now let us look at another protocol - the Rether. The Rether stands for real-time Ethernet. So, from the name we can guess that it is based on the Ethernet.

(Refer Slide Time: 39: 37)



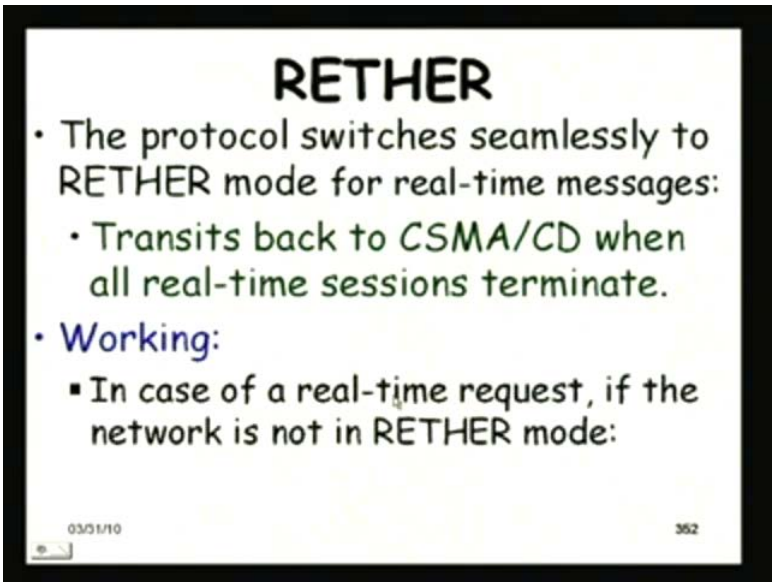
Actually, this protocol switches between ethernet and the token ring protocol. The ethernet have some advantages for non real-time messages ethernet is good; leads to higher channel utilization for non real-time messages. For, a token ring protocol does not really work well when only for non real-time messages; do you agree with that? Yes, do you agree with that? That the ethernet is more efficient or it is a better protocol for non real-time messages and the token ring protocol is not as good to transmit non real-time messages and then I gave the hint that the ethernet can lead to higher channel utilization compared to the token ring. So, when will that occur let us give an example. Not very hard; see, the fraction of bandwidth is allocated to different nodes and let us assume that a node does not have any data to transmit but it will be you know there will be a wastage of the bandwidth (Refer Slide Time: 41: 10). So, this protocol switches between the ethernet and token ring protocols for a depending on whether the transmission is for non real-time messages or for real-time messages.

(Refer Slide Time: 41: 29)



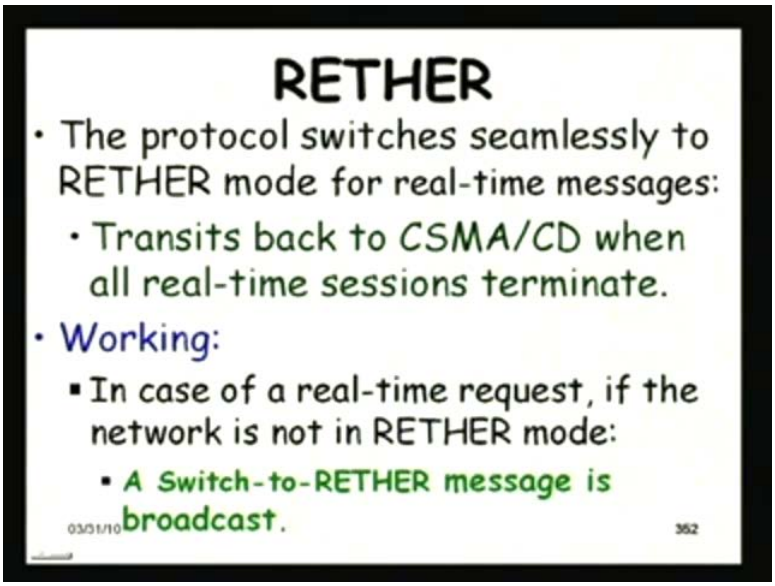
The transmission occurs in ethernet mode in CSMA/CD protocol and in for the real-time messages in the Rether and it switches to Rether mode when there are real-time messages to transmit and once the real-time messages are over it switches back to the ethernet the CSMA/CD.

(Refer Slide Time: 41: 49)



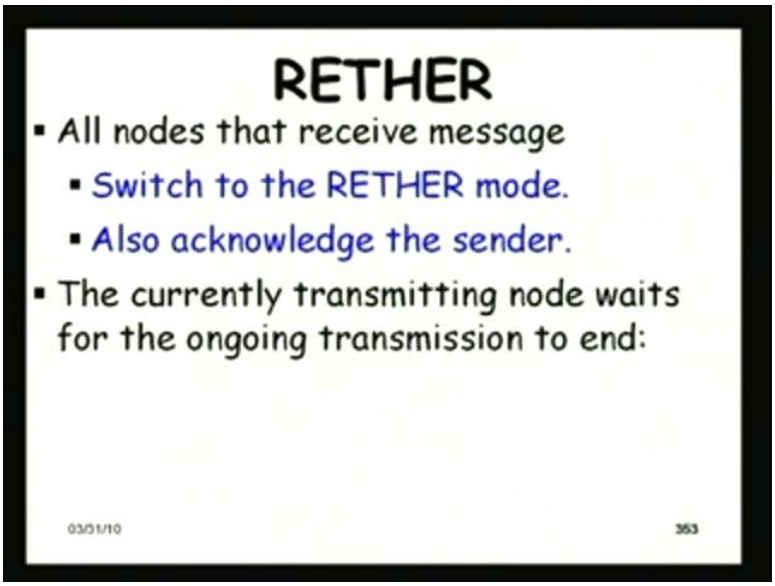
So, when a real-time request comes from a, for a node if it is already in the Rether mode, no problem, its request is registered and it transmits its message but if it is not in the Rether mode it is in the ethernet mode then, it broadcasts a switch to Rether mode.

(Refer Slide Time: 42: 23)



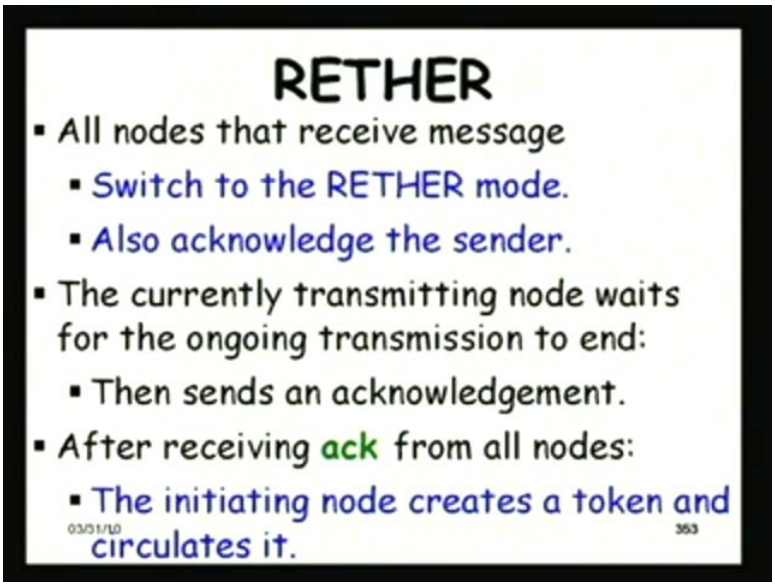
Switch to Rether mode is broadcast and all nodes that receive the message switch to Rether mode waiting for a token and they also acknowledge the sender saying that they have switched their mode; they are not going to transmit anymore non real-time messages. Only when they receive the token they will start transmitting. So, that is the mode that switch to and then that node which was transmitting some message, it just completes that, right.

(Refer Slide Time: 42: 39)



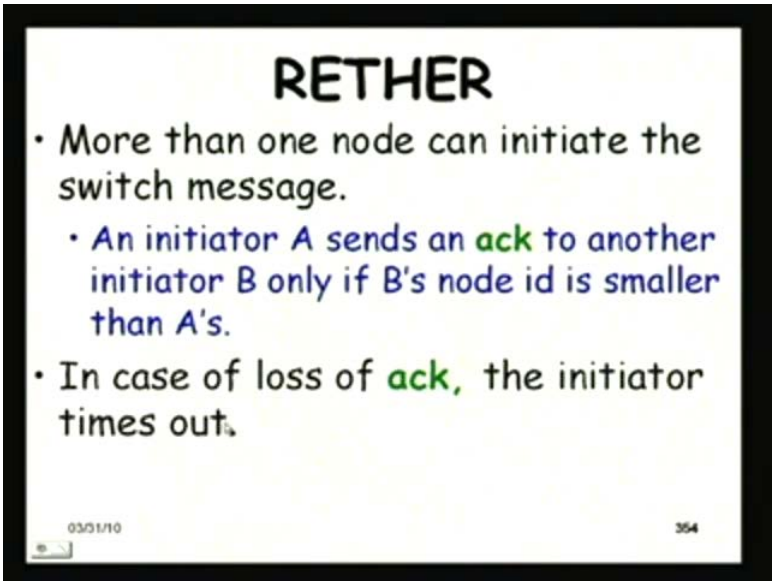
It was sending, trying to send a packet or something or a frame, it just completes transmitting that frame and then after transmitting the frame, it sends an acknowledgement and changes to the Rether mode. After the node initiating the Rether mode gets acknowledgement from all the nodes, it creates a token and starts transmitting and then circulates it.

(Refer Slide Time: 43: 36)



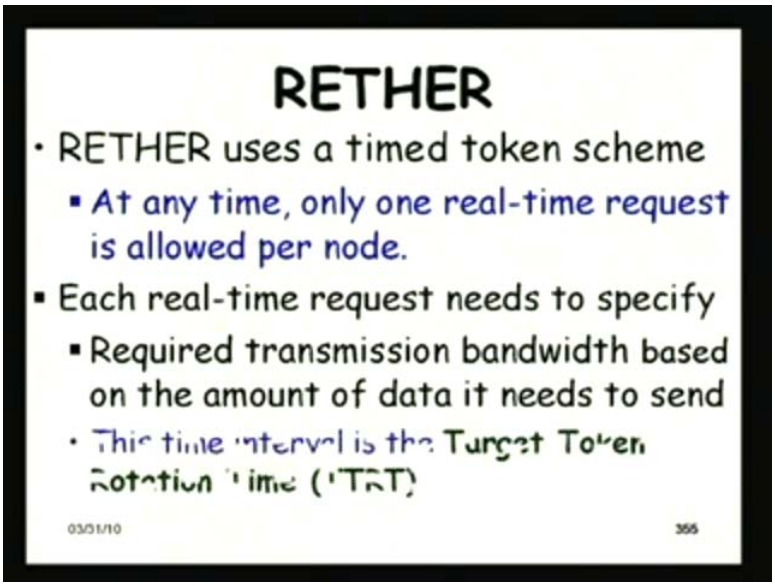
Finally, when there are no real-time messages it does not have any more real-time messages, it sends a broadcast for starting the ethernet mode. But, one complication is that it is possible that 2 nodes get real-time message at the same time and both of them they start - want to start the real-time mode. Then, there will be a problem, is it not? So, let us see how the protocol handles this. When more than one node tries to initiate the switch mode, the switch message, switch to Rether mode then there will be a problem if both of them generate the token, right. So, here the way it is reserved is that an initiator A will send an acknowledgement to another initiator B. so, b had also transmitted switch to Rether it will send to B only if B is node ID is smaller than A; so B is higher priority to a then only it will acknowledge otherwise it will withhold the acknowledgement and then B will understand that it cannot really generate a token.

(Refer Slide Time: 44: 00)



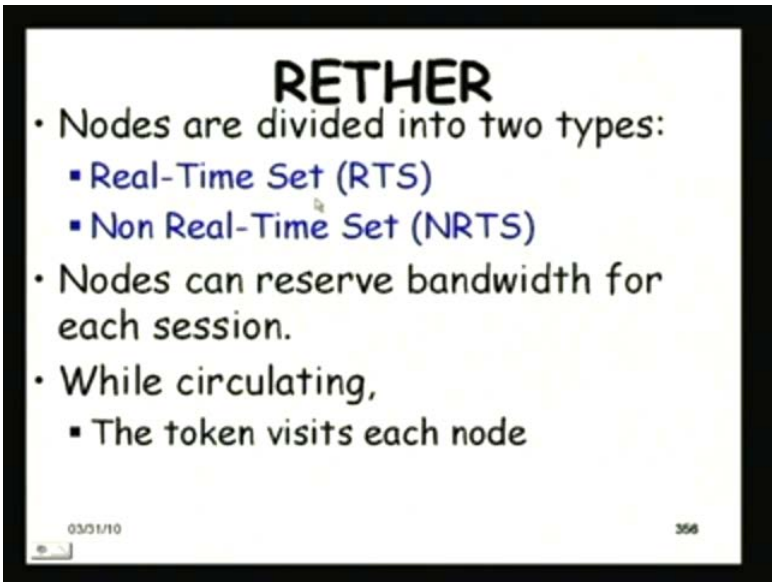
Then, we have to look at another possibility is that, what if there is a loss of acknowledgement/ some node sent an ack but lost; then the initiator would timeout and then it would retry again send a switch to Rether mode message.

(Refer Slide time: 45: 35)



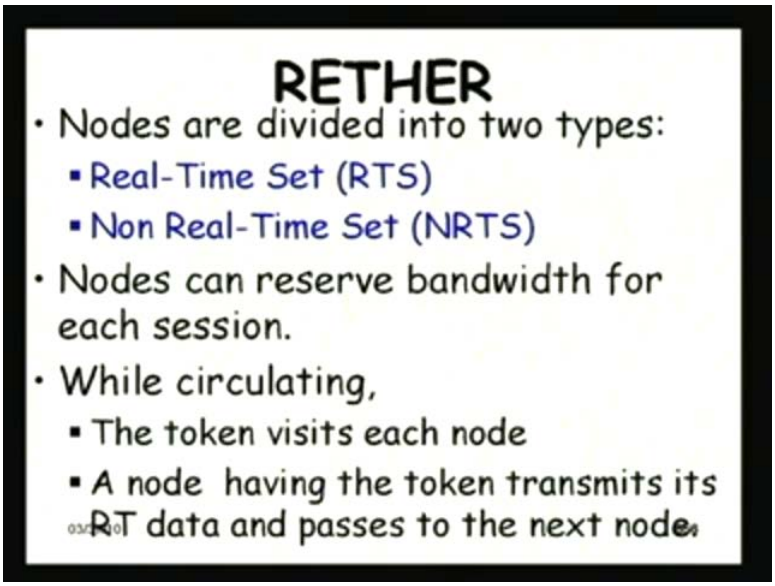
Once in the Rether mode uses a timed token scheme and at any time only one real-time request is allowed per node so that is the time for which it can transmit. it can say, that I have 2 3 messages with different characteristics; each it has to convert that into one request and once for every request needs to specify the total bandwidth that it would not need to transmit the total data that needs to transmit and based on this the T T R T time is framed. The T T R T is set and one thing that we need to remember is that all nodes need not have real-time messages at any time

(Refer Slide Time: 46: 25)



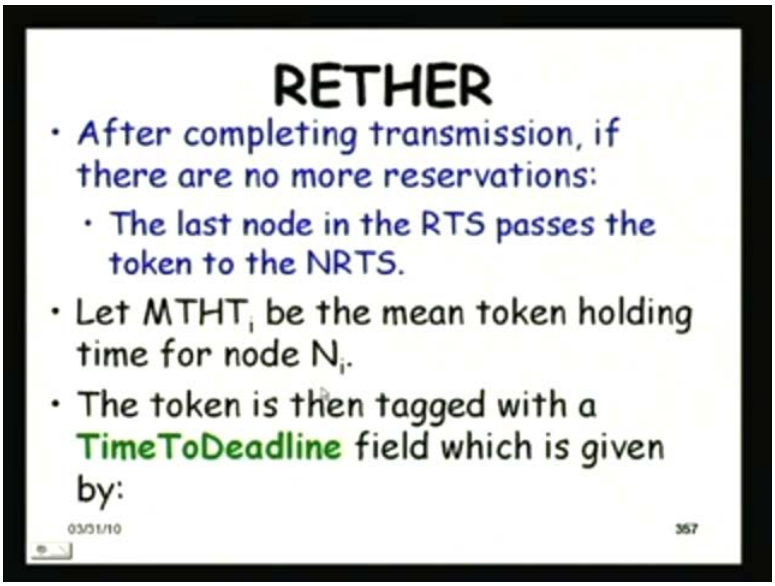
Some nodes have real-time messages and it is possible that some nodes at that instant have only non real-time messages. So, then how will it work? So, we have to also consider this case. Now, the real-time messages the nodes having real-time messages the reserve bandwidth and also there is some time for the non real-time set. This will be decided based on how much data the real-time messages would take over the T T R T so the, what is their deadline and how much data they need to transmit before the deadline based on that the T T R T is set and how much the time amount of time that is left out will be given to the non real-time set to transmit. So, while the token is circulating it actually visits the nodes in the RTS - the real-time set.

(Refer Slide Time: 47: 30)



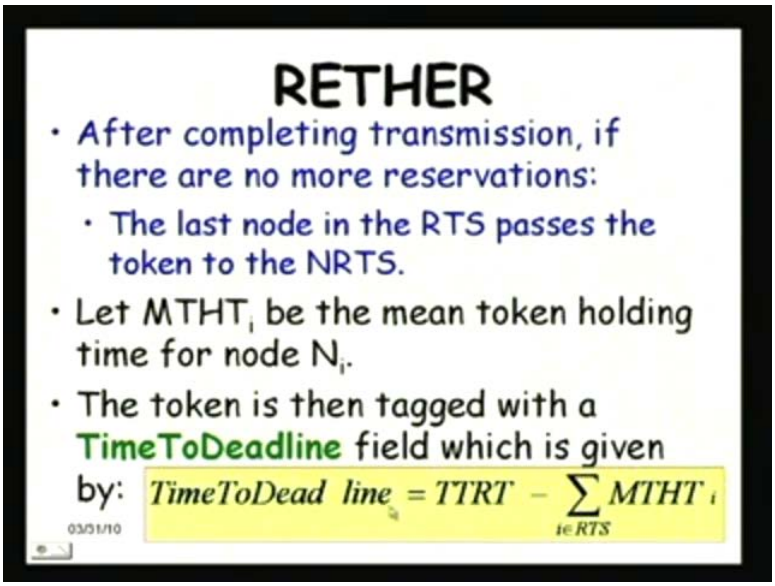
While circulating, the node visits the real-time set and they transmit node having the real-time or the synchronous data they transmit and then pass to the next node which transmits and after completing the transmission of all the real-time messages, if there is still time left out then the token is passed to the non real-time set the nodes in the non real-time set and they start transmitting.

(Refer Slide Time: 48: 00)



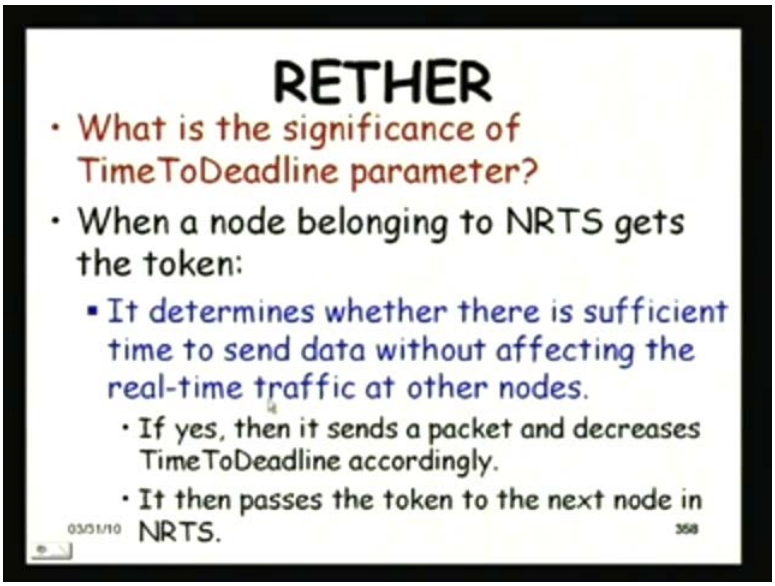
If m T h T i is the token holding time for the node N i and so this is a value let us define as a variable, MTHT i is the mean token holding time for node N i and the token itself is tagged with a variable. Here, time to deadline field basically it says how much of the T T R T is left out. The time to deadline in the token means it is a indicator of how much time is left out before the T T R T would expire.

(Refer Slide Time: 48: 55)



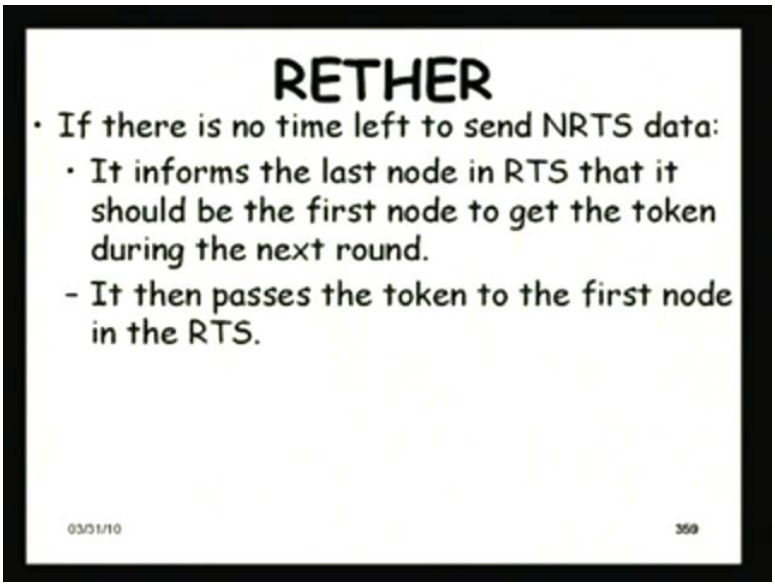
So, if you think of it in terms of MTHT i and T T R T, we can express the time that the non real-time set can transmit is T T R T minus MTHT i. The summation of all these, all the nodes having real-time messages to transmit the summation of the token holding time for them T T R T minus this will be the time that the other nodes can transmit. So, that you have seen that the time to deadline parameter indicates how much time for the token holding for the token to T T R T is to expire.

(Refer Slide Time: 49: 28)



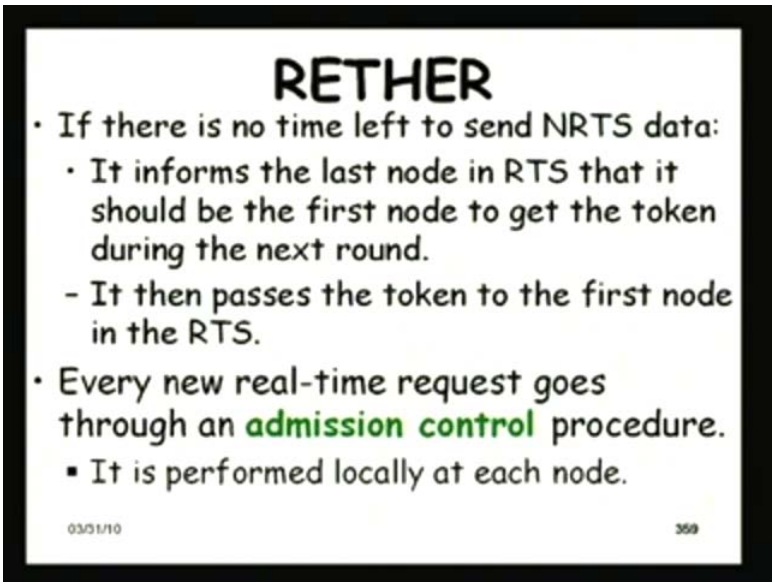
Based on the time to deadline the non real-time set nodes in the non real-time set, they know how much they can transmit so once the time to deadline becomes 0 they will have to release the token (Refer Slide Time: 50: 00) and each time they transmit time to deadline decreases and once one node completes transmitting its asynchronous messages transmits, it passes the token to the next node in the NRTS and this happens until the time to deadline parameter becomes 0 and at that moment it has to go back to the real-time set.

(Refer Slide Time: 50: 30)

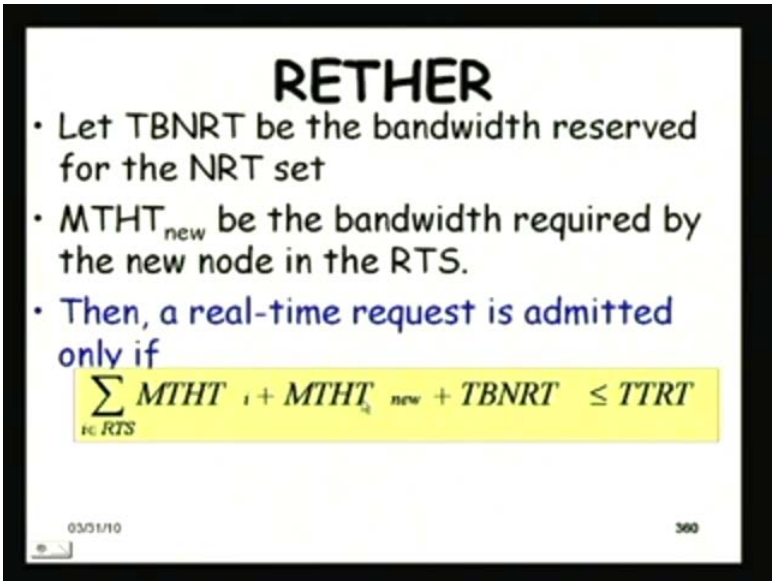


Also, to prevent a node from monopolizing while transmitting the non real-time messages somehow it has to be remembered that which token had last got the chance to transmit the non real-time messages. Because, it might again come to that node and then that node will monopolize. It will have lot of data to transmit; all other nodes will be stormed. So, which node was, last got chance to transmit the non real-time messages is remembered and next time the real-time (Refer Slide Time: 51: 25) messages are over, it comes back to the next node which had transmitted the asynchronous messages. And, then one final is so that you need to consider is the admission control procedure. So, what if a node gets a real-time message? Will it just start transmitting? In that case, if there are too many messages then the messages will start missing deadline in a unpredictable manner, the admission control procedure is as follows each node locally determines that whether it will be possible to honor its request but how will it do that very simple.

(Refer Slide Time: 52: 00)



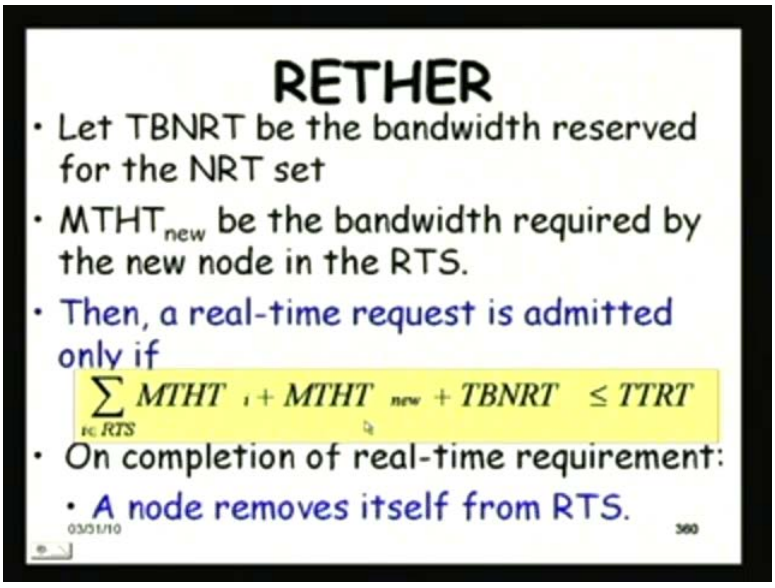
(Refer Slide Time: 52: 10)



If it knows the MTHT or the different nodes and the T T R T, so if the, so this is the time for (( )) its reserved for the asynchronous messages ok. This will not just starve and this is the one which has already been allocated. Now, if the new request that has come in, if all these sum up to less than T T R T, then this request can be honored that is the admission control (Refer Slide Time: 52: 47).

If this happens to be larger than the sum happens to be larger than T T R T then, some messages will start missing their dated line is not it? So, finding the admission control or finding whether a new request can be honored is also easy to find out; is it not? I hope all of you can able to do that is, given a situation. What is the mean token holding time for each node and the new nodes requirement and the reservation for sending the non real-time messages based on that? Can reason out whether a new message request can be honored?

(Refer Slide Time: 53: 42)



Finally, when a node completes transmitting its real-time messages there are no further messages we kept. It will send a request to remove it from the real-time set so that the token will not visit it during the transmission of the synchronous messages. So, this is also popular protocol. We would like to have done some problems on this also but considering the time that we have will not do that I just leave it to you as an exercise and will stop this discussion. Now, will continue for the next lecture next time ok; thank you.