Real-Time Systems

Prof. Dr. Rajib Mall

Department of Computer Science and Engineering

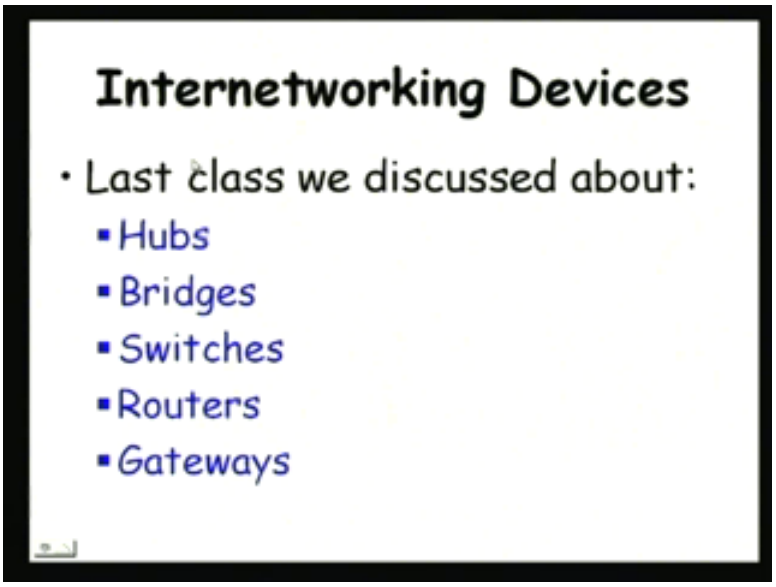
Indian Institute of Technology, Kharagpur

Lecture No. # 34

Real-Time Communication in a LAN

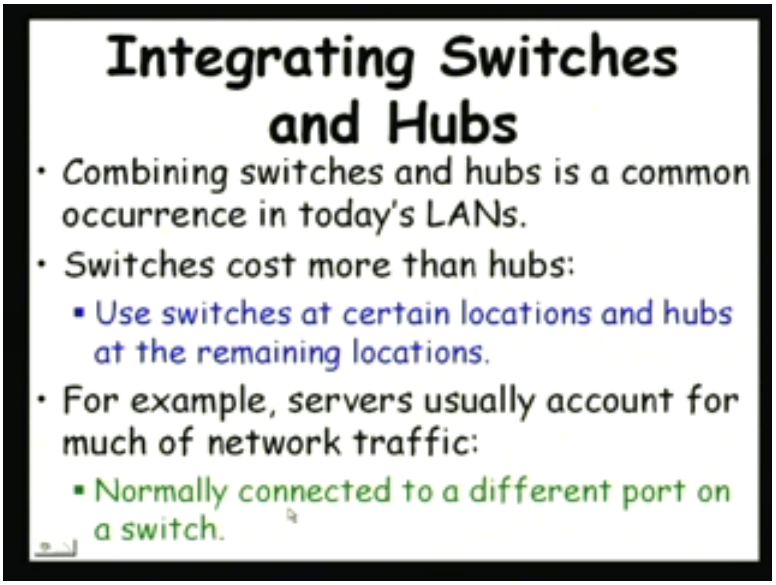
Good morning. Let us get started over the last few classes we were discussing some basic concepts in real-time communication and we reviewed our knowledge in computer communication. Now, let us look at how real-time communication can occur in a LAN environment and we will see that there is no standard protocol that has yet emerged, but there are many competing protocols which are being used in different situations. Let us first finish what we are discussing last time. So, let us small part left out and then we will proceed towards discussing about real-time communication in a LAN environment.

(Refer Slide Time: 01:09)



Last time we were discussing about the internetworking devices .We discussed various types of devices that are used in internetworking Hubs, Bridges, Switches, Routers etcetera.

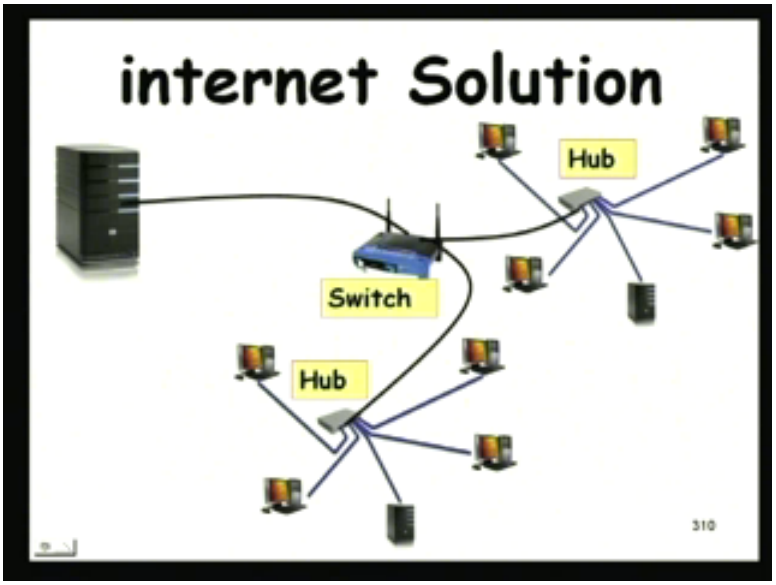
(Refer Slide Time: 01:29)



But one thing we did not mention is that when we have an interconnection network where we have the option of using switches and hubs where to use which one we did not discuss about this issue. Let us just complete this issue then we look at the real-time communication in a LAN environment. If you look at any network nowadays you will see that there are many switches and hubs in the LAN and we know that switches cost more than hubs we were saying that hubs are very inexpensive can buy in couple of hundred rupees switches are more expensive. We need to restrict switches to certain locations from cost considerations and need to use hub at the other places. So, where to use which one.

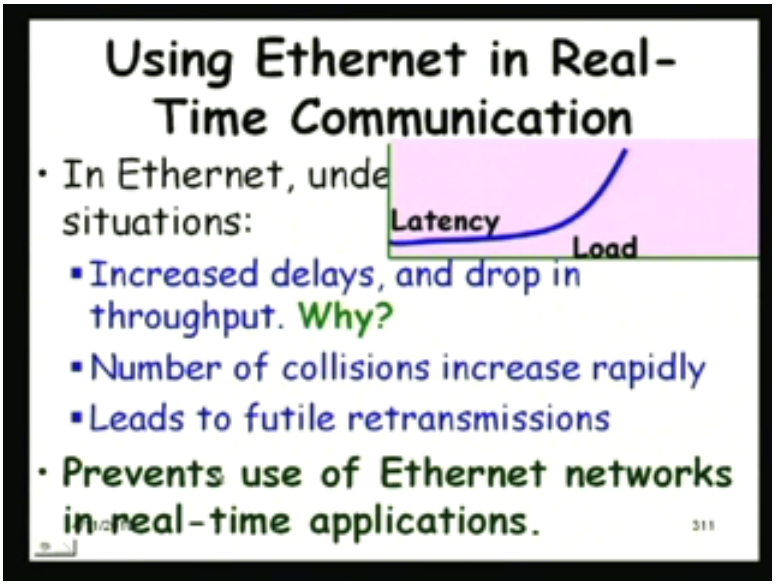
Now, a common sense is that we need to use a switch port for computers which are heavy generators and consumers of network traffic and we know that servers account for much of the network traffic and therefore natural that we connect a server to a switch rather than a hub. Because if it is connected to a hub it will increase collision and it will cause collisions with other computers on the hub and it will reduce the throughput.

(Refer Slide Time: 03:12)



You might find that desktops and low end servers etcetera these are connected to a hub and the hub in turn is network through a switch where as a server which is heavily used is connected to a switch port. It does not participate in the LAN activities here and collisions do not occur here.

(Refer Slide Time: 03:53)

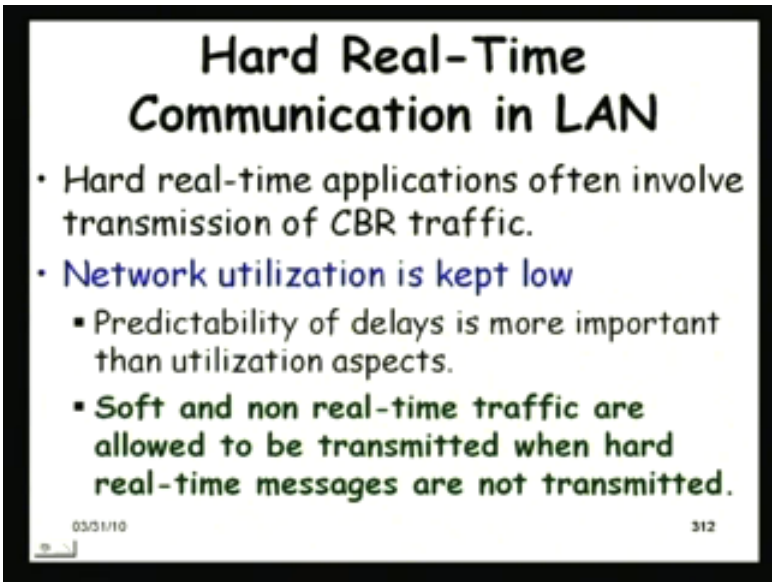


Now, let us proceed towards discussing ethernet in real-time communication. Can we use ethernet in real-time communication .What do you think. What can be the problem using ethernet in a real-time situation. Ethernet whenever a collision occurs there is an alteration that is called (( )) performance. That will can lead to any amount of time there is no perfect bound on an algorithm. The Ethernet does not guarantee.

By which a transmission can complete and theoretically the communication may never take place if it know just keeps on experiencing collisions, but let us look at it in more detail closer. One of the main problem is that under high load situations we have the delays and drops that the delays in the communication increases very rapidly as the load increases the delays would increase very rapidly and also there will be a drop in throughput if you look at it as the load increases here it increases slowly .Then suddenly becomes almost unusable latency increases very rapidly. Typical performance of any ethernet you can even experiment by generating a large traffic and larger traffic. See that there are some threshold values after which it becomes almost unusable. Why is this situation. There are more number of collisions.

The main reason is that number of collisions increase rapidly as the load increases. So, there is no collision just gets transmitted then, if there is a collision few nodes collide. There are chance that as the load increases they collide again and again leads to futile transmissions. They just keep on transmitting resulting in a collision and if the load increases further there may be zero throughput just keep on colliding and. This is one of the main reason why the ethernet networks cannot be used in a real-time application ,Because consider the situation where there is a alarm avalanche or something. Then the load drastically increases and the throughput reduces the delays increase. This is a common knowledge that we need to have before we discuss further.

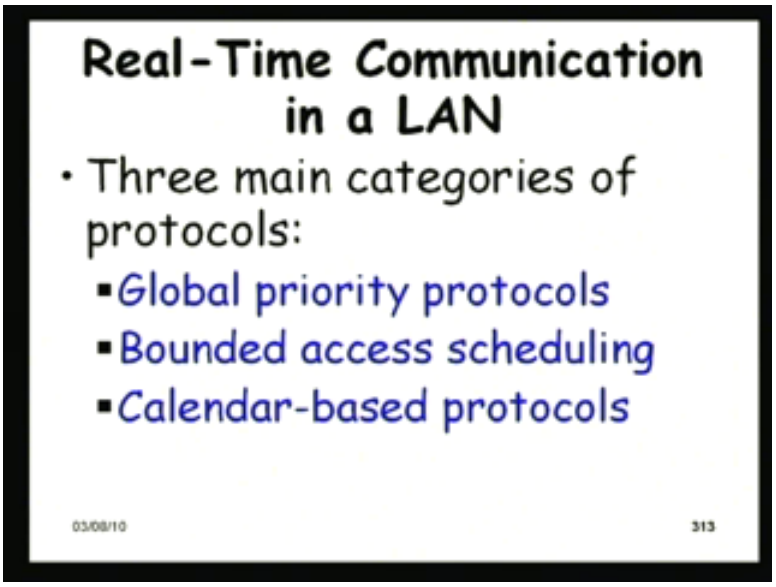
(Refer Slide Time: 06:58)



Now, let us see how hard real-time communication can occur in a LAN. Many times there are constant bit rate traffic, but there can be other types of traffic and one thing that

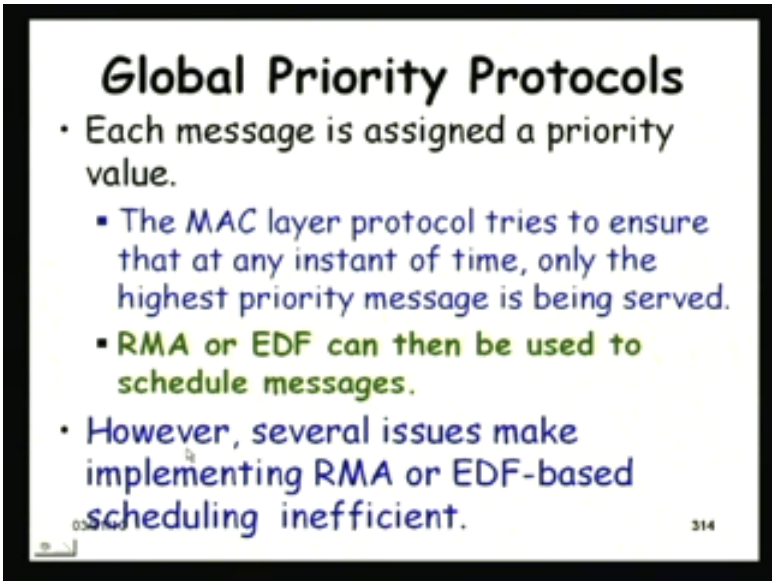
we need to understand is typically the network utilization is kept low .Only when there are alarm avalanche and sudden generation on traffic the load increases, but typically they operate under low utilization and low load and the predictability is more important than the utilization aspects . Since the load due to the hard real-time messages is low whenever these messages are not there the soft and non real-time traffic are allowed to be transmitted .Only when hard real-time messages do not exist.

(Refer Slide Time: 08:02)



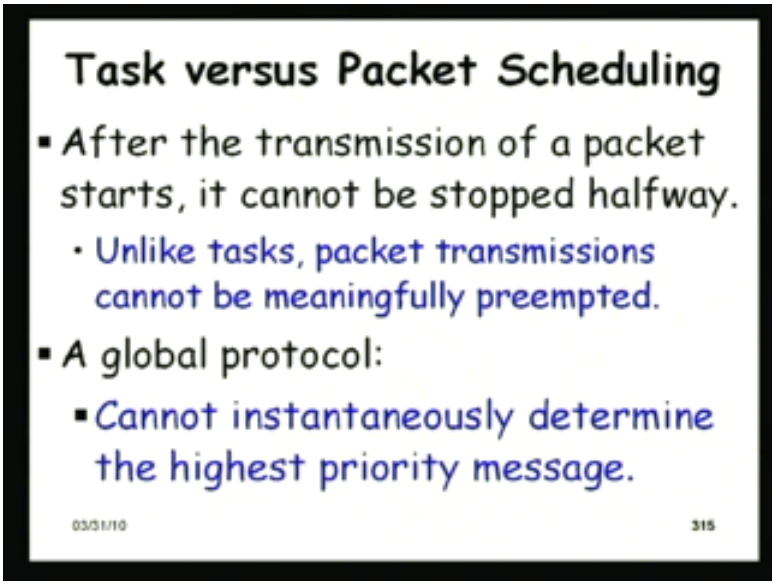
We will examine three main categories of protocols: One goes by the name Global priority protocols here, all the messages are assigned priority unique priority values and based on the priority values they are assigned slots to transmit. A Bounded access scheduling where they priorities are not really the messages, but if the stations they are given sometime to transmit and within that they can locally schedule the messages. Then we have the Calendar-based protocols where they just reserve sometime time slots are reserved and then they transmit.

(Refer Slide Time: 09:00)



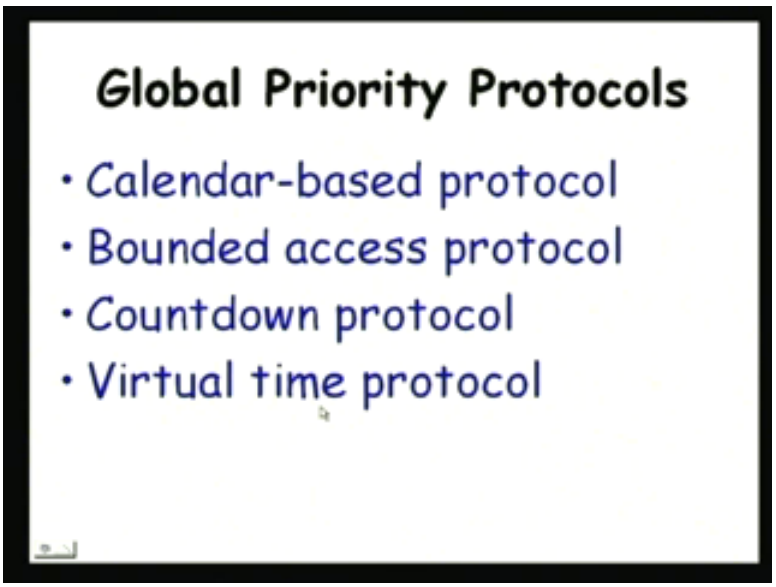
Now, let us look at the global priority protocols here, each message is assigned a priority value respective of the irrespective of the node at which the message arises all are assigned priority values . The MAC layer protocol tries to ensure that at any instant of time .Only the highest priority message is served and once we can achieve this that all the messages are assigned priorities at anytime only the highest priority message will be solved we can think of using RMA or EDF to schedule messages. It basically boils down to something like a task scheduling .There also we had the concept of priorities to tasks and then we found that RMA and EDF are the optimal algorithms to schedule the processor. We can think of the network here as processor and the messages are assigned priorities which are analogous possibly to the tasks and then we schedule them using RMA or EDF. Unfortunately there are many issues. We will just discuss couple of them which makes implementing RMA or EDF based solution rather inefficient is not really analogous to a task scheduling on a processor.

(Refer Slide Time: 10:40)



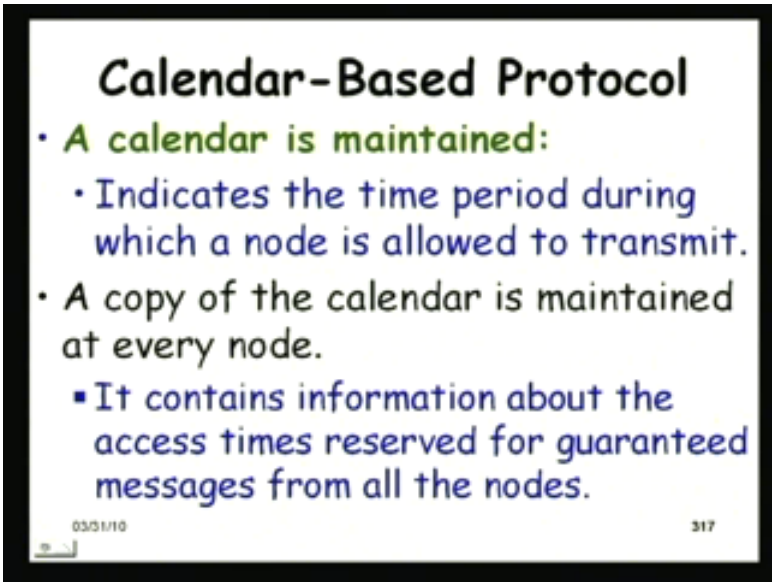
Let us see the complications. One is that we can stop a task when a higher priority task comes you can preempt it and then we can restart it later, but what about a packet either you transmit it fully or you do not you cannot just stop it half way .Then continue later does not work that way .Unlike tasks packet transmissions cannot be meaningfully preempted and also here the stations the nodes are distributed .It is difficult to have a global knowledge of which node has the highest priority message at a instant. Cannot instantaneously determine the highest priority message, because we do not have a global knowledge .They are distributed on different nodes network and network transmission do does take time. Because of the delay we do not have a global knowledge.

(Refer Slide Time: 12:02)



Now, we will look at Calendar- based protocol, Bounded access protocol, Countdown Protocol and Virtual time protocol.

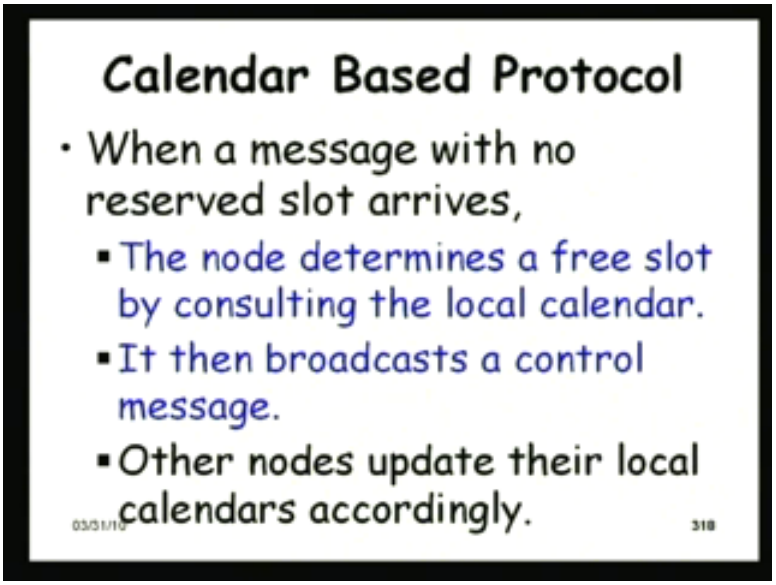
(Refer Slide Time: 12:13)



Let us look at that. The calendar-based protocol using very simple applications where there are just few sets of messages and few nodes .We will see or just observe that it does not work and tell me why it do not work when the load on the network. I mean there are many nodes and many types of messages to be transmitted .The calendar -based protocol may not be a good solution. Let us look at that, please observe that.

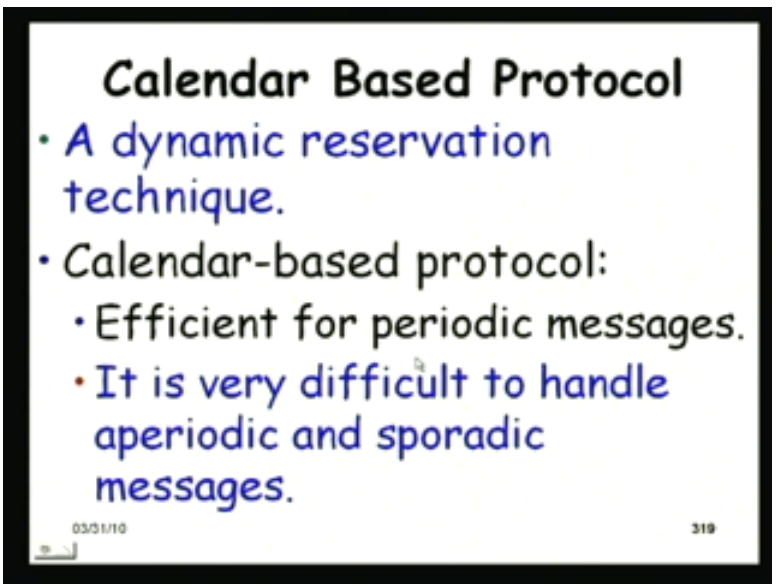
Now, let us first see how the protocol works as the name of the protocol suggests there is a calendar which is maintained at the at every node a copy of the calendar is present in every node and the calendar indicates the time period during which a node is allowed to transmit. So, for every node there are slots which are assigned to them during which they will transmit .Copy of the calendar is maintained at every node and contains information about the access times reserved for guaranteed messages from all nodes. So, these are the hard real-time messages. The transmission times are reserved for guaranteed messages.

(Refer Slide Time: 13:50)



But it is possible that there are many other types of messages other than the hard real-time messages which arise at specific nodes .When a message with no reserved slot arrives the node where the messages arrives it looks at its local calendar finds a free slot and then broadcasts a control message for reserving that slot. It cannot just start transmitting on a free slot, because there might be other nodes which might try to transmit at the same time. So, it needs to reserve that slot and once the control message is successful then all nodes update their calendars accordingly and this transmits, but of course, it would not be a permanent slot just transmit for that instant.

(Refer Slide Time: 15:01)



So, we can think of it as a dynamic reservation technique for some categories of messages there are static reservation and for other messages the dynamic reservation. Now, if the periodic messages are not predictable depend on which node they will arise etcetera. Then it would become inefficient, but we know that the hard real-time messages they mostly restricted. I mean the CBR kind of traffic they are predictable from where they arise etcetera. So, such a messages its efficient, because slots are reserved from the beginning by the designer and they just keep transmitting on other nodes do not use that slot. But for other categories of messages a periodic and sporadic messages it becomes difficult, because each time reservation has to be done and then it needs to transmit in a slot.

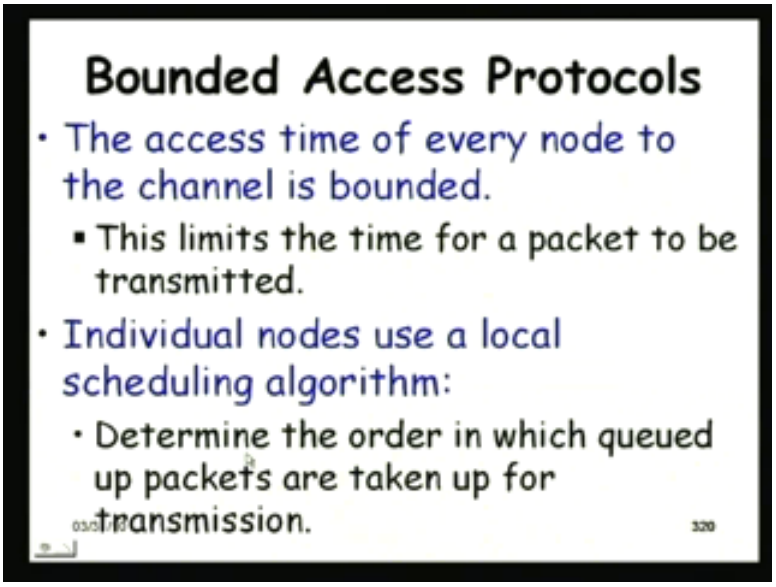
So, lot of delay will be involved and also if the periodic messages are not predictable. I mean they can arise anywhere then it becomes difficult to handle those. In this calendar based protocol we have security static for some messages. If it is static even if one message on a worst case in basic slot then will it have effect on the remaining things there this will get postponed …Not really. See the static reservation is made for constant bit rate traffic. Let us say every hundred millisecond need to transmit some packet of data.

So, it will make a reservation for every hundred millisecond some slot of data and those are predictable traffic, but if the traffic is unpredictable then the reservation cannot be done that is what I have just saying. Only for those which have predictable traffic for that reservation is meaningful for others it just temporarily they will try to book a free slot and transmit. Even if it is predictable. Tend to happen that a message is corrupted… Or that we have missed its slot dead line in worst case like. Then it becomes complicated. Here it is assumed that a slot is assigned to it will transmit on that time. If in case if it is missed it then it becomes complex. But what about the problems with this protocol ?.I mean I said that it gets used for very simple applications.

So, what do you think?(( )).That we know that for periodic traffic it can be used and for others it becomes inefficient lot of delays will be incurred. We have to know which packet is sending at what time everything should be ready to work through the … So, very simple situations where the messages are almost totally predictable it can be used in that situation. The network inflation may not be in done. See if these are messages which are predictable then appropriate reservations can be made and utilization can be high for

some applications utilization can be high if the set of messages can be scheduled and we can find a schedule for that. So, reservation will be made based on that.

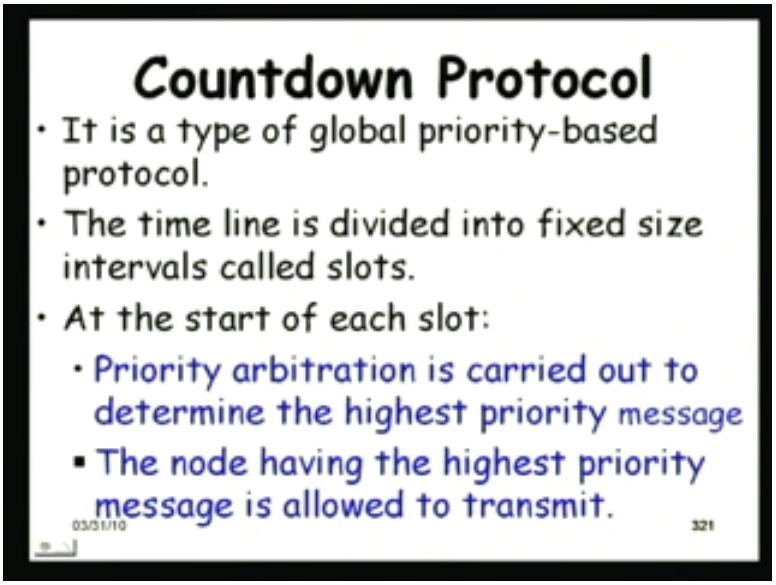
(Refer Slide Time: 19:05)



Let us look at the bounded access protocol. Here the access time of every node is bounded. So, they can transmit only for certain duration of time. So, this limits the time for which a node can transmit the packets and since every node has a time for which it can transmit .The individual nodes can use a local algorithm to find out which are the higher priority messages at that local node and during its slot it will transmit. Each node in the network will be given some slot and they have to restrict themselves to using only that slot that is the bounded access protocol.

But the thing is that all nodes need not have the same slot similar slot. The size of the slot need not be same for all the nodes in the network. It might vary depending on the kind of traffic they have amount of data needs to be transmitted. The individual nodes once their slot comes the determine the order in which the queued of packets are to be transmitted and they use that to use their slot.

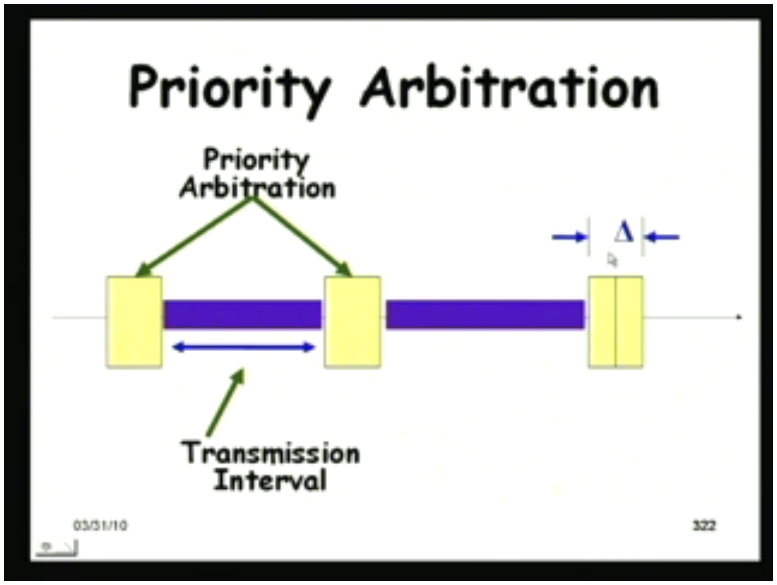
(Refer Slide Time: 20:39)



So, we look at this protocol the bounded access protocol in more detail we will solve some problems and look at some examples, because used in manufacturing situation quite frequently. We will also examine what is good about it and bad about it later we will do a performance comparison we will see that where the bounded access protocols score over the other kinds of protocols. So, let us look at another protocol the countdown protocol we had seen this in some context in our previous discussions in the context of controller area network that we discussed. That was actually countdown protocol even though we did not tell it that name that time.

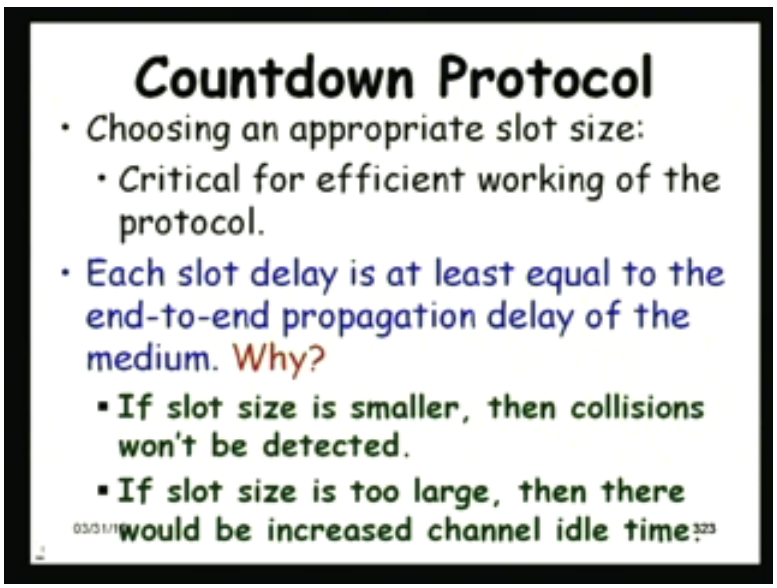
So, let us look at it is a global priority-based protocol. That means that every message needs to have a priority value assigned to it. So, here the time line is divided into fixed size intervals called as slots. So, if we let us say every ten millisecond is a slot .At the start of each slot an arbitration is carried out to determine the highest priority message at that instant and after the arbitration period. So, there are slots and every slot to start with has a arbitration period and at the end of the arbitration period it will be known which node has the highest priority and that node will be allowed to transmit.

(Refer Slide Time: 22:33)



So, If we have fixed sized slots here. Fixed sized slots the timeline is divided into fixed size slots .Does not look like fixed size, but these are fixed sized slots and each slot is preceded by an arbitration interval its exaggerated here, if the arbitration interval takes so much of time then it will be inefficient .The arbitration interval is much much smaller compared to the transmission interval. So, at the start of every slot priority arbitration takes place and once it is decided which node has the highest priority message at that instant it utilizes the channel .Until, the start of the next slot and the duration for which the priority arbitration occurs is typically very small denoted by delta.

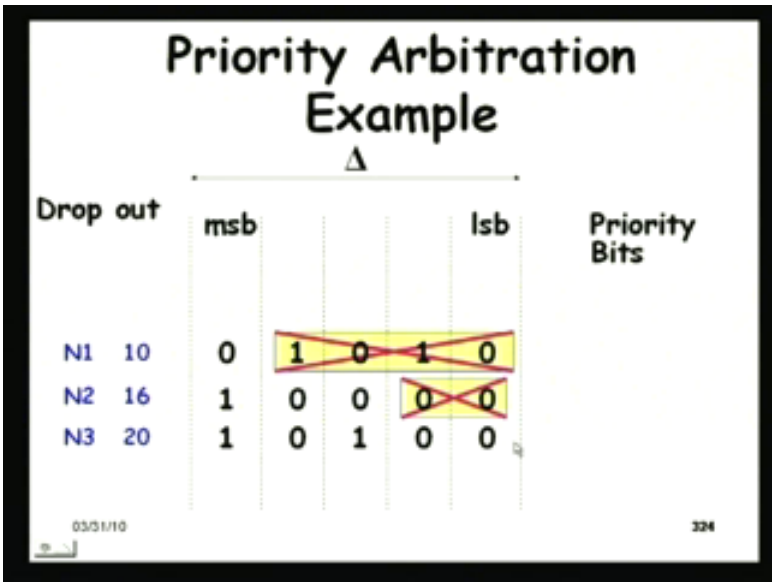
(Refer Slide Time: 23:41)



Choosing the size of this arbitration interval and the slot size is very important and in each arbitration interval should be equal to the end- to- end propagation delay of the medium. So, the size of the arbitration interval should be equal to the end- to- end propagation delay of the medium .Why is that?(( )).Not really .See if it is less than that what will happen. Within that time an another slot may begin and another node may start sending messages. Not really. If it is less then that time the propagation time then, there may be nodes trying to transmit where the collision will not be detected.

The actual priority value will not it would not be possible to determine. So, the arbitration interval here actually the terminology is not correct it is not slot delay it is the arbitration interval should be at least equal to the end- to- end propagation delay of the medium .If the arbitration interval is smaller then collisions would not be detected. So, initially the nodes keep on transmitting and if there is a collision they know that there is a higher priority message which is getting transmitted. So, if the slot size is too large or the arbitration interval is too large then there would be increased channel idle time.

(Refer Slide Time: 25:42)

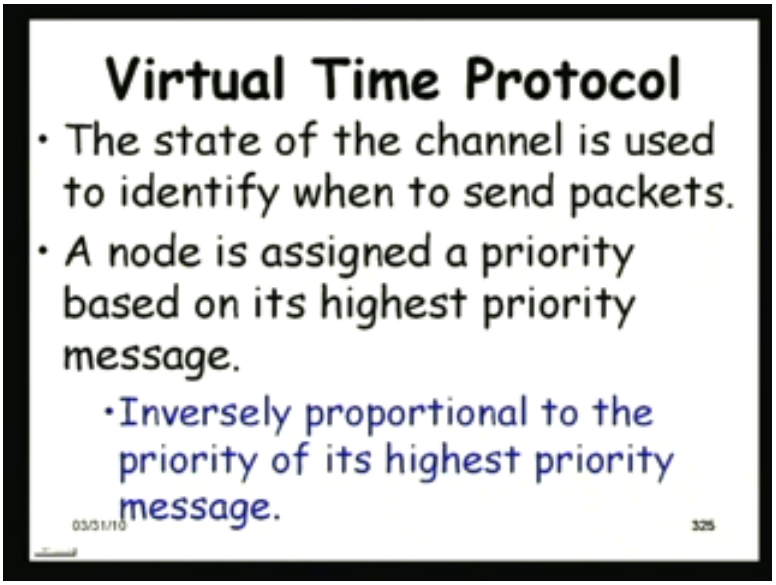


This is an example of the priority arbitration. So, to start with every node starts transmitting their priority value. So, N 1 transmits zero assuming that higher priority is indicated by higher values of the priority .Sometimes we have the situation where the higher priority is indicated by lower priority value but here, it is the higher priority value just check here that N 1 which has priority ten is a lower priority compared to N 2 which is priority sixteen and N 3 has priority twenty. So, N 1 at the start of the arbitration

interval or at the start of the slot the same thing start of the slot is the start of the arbitration interval.

The start transmitting from the msb and N 1 transmits zero and finds that one is been transmitted. So, it drops out does not transmit its rest of the priority values .Then N 2 finds that it writes a zero it could sends a zero continues and next transmits a one and sends as a zero drops out. So, that is the way it works and then N 3 is the winner at the Arbitration and it continues transmitting. So, if this becomes too small then the protocol would not work, because nodes will falsely assume about their priorities.

(Refer Slide Time: 27:52)

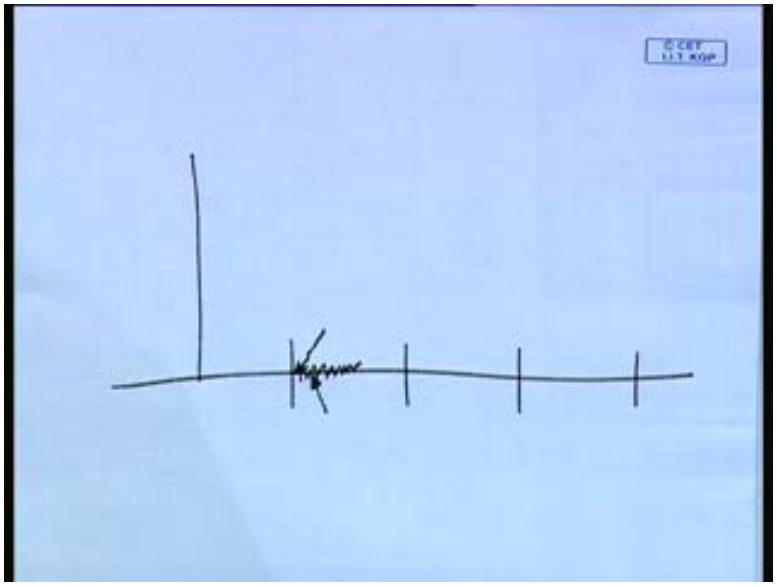


Let us look at the virtual time protocol again a popular protocol here the state of the channel is sensed and that is used to identify who can send a packet. So, again this is a global priority protocol each node is assigned a priority based on the highest priority message it can transmit, but again at a particular instant it may not have the highest priority message. So, it is the node responsible for a high priority message does not mean that at every instant it has the highest priority message. So, that is a shortcoming of this protocol.

Now, let us see the working. The node is assigned the priority based on the highest priority message it has and here it should be it waits for a period inversely proportional to the highest priority message that it has. So, what it really means is that, depending on the priority see they keep on sensing at periodic intervals and then a node if it senses that

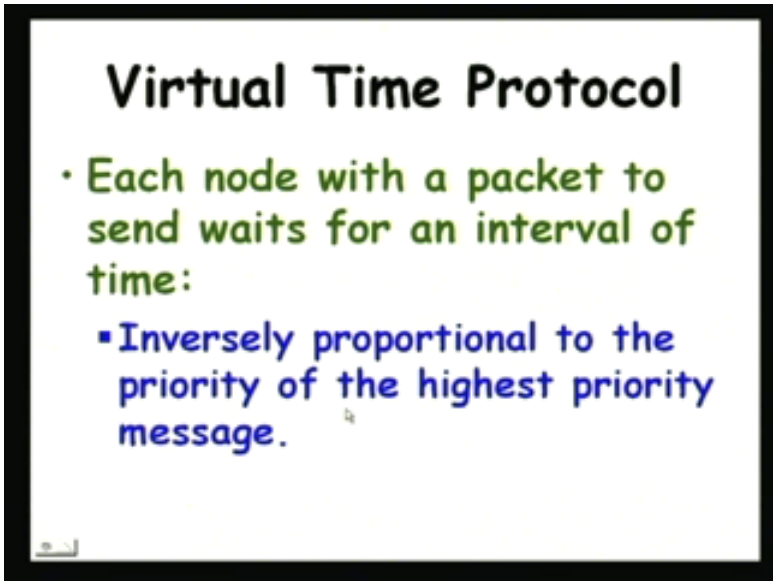
the channel is idle. It does not immediately start transmitting, but it waits for a time that is inversely proportional to its priority.

(Refer Slide Time: 29:40)



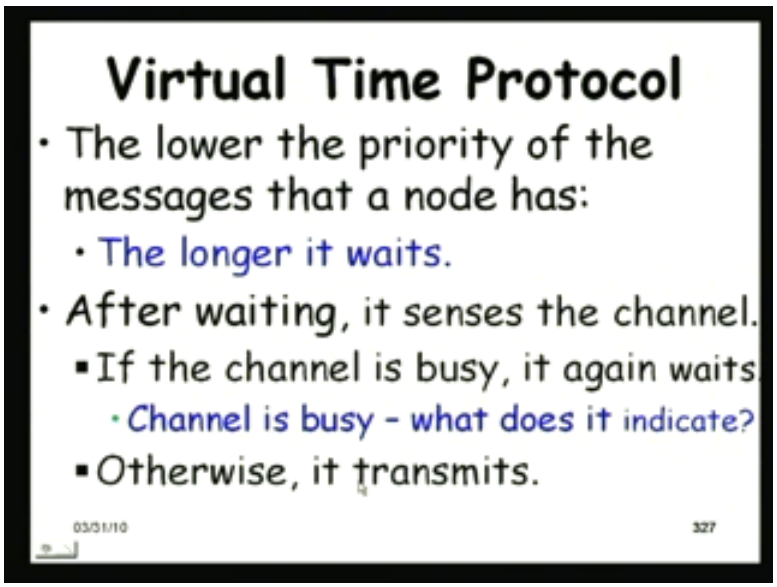
So, the protocol would work as follows again the time has to be divided into slots and at the start of the slot the arbitration would takes place where a node would start sensing here the channel . It would also look at its own priority and wait for a period before which it can transmit the message .If it has the priority message it will immediately start transmitting here, it would not wait much. But, once the transmission occurs then the lower priority the node with the lower priority messages will start sensing slightly later .It will find that the channel is busy and then it will not transmit .It knows that a higher priority message is already getting transmitted. So, the period for which a node waits before it tries to transmit is inversely proportional to the highest priority message that can be there at that node.

(Refer Slide Time: 31:04)



So, each node with a packet to send waits for an interval of time which is inversely proportional to the priority of the highest priority message.

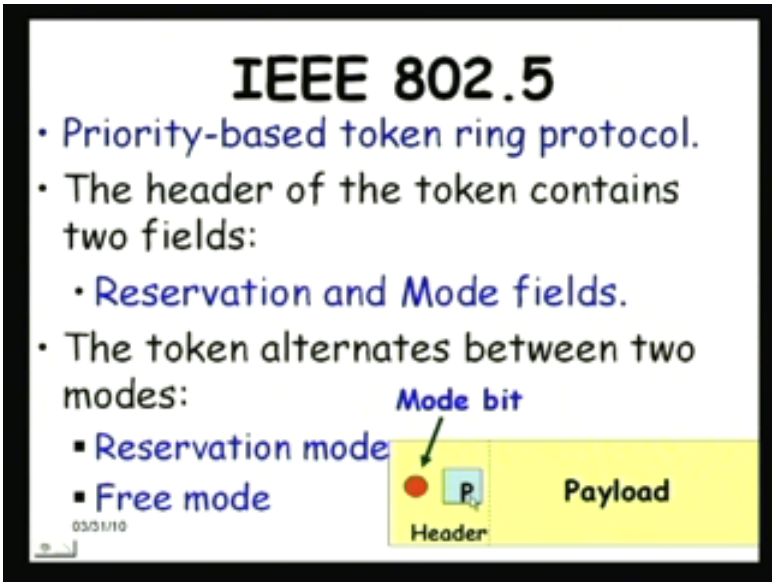
(Refer Slide Time: 31:15)



The lower the priority of the message that a node the longer it will wait at every arbitration interval .It will keep on waiting until other higher priority messages have completed sensing and it can be inferred that there is no higher priority messages for that interval and then it can transmit. So, after waiting for certain time it would sense the channel if the channel is busy it would wait again and we know that the channel is busy if a node finds that the channel is busy by the time it senses what would it indicate. It would indicate that a higher priority transmission is taking place. So, if it finds it idle

knows that there are no higher priority messages there can be lower priority messages, but no higher priority messages and it would start to transmit.

(Refer Slide Time: 32:23)



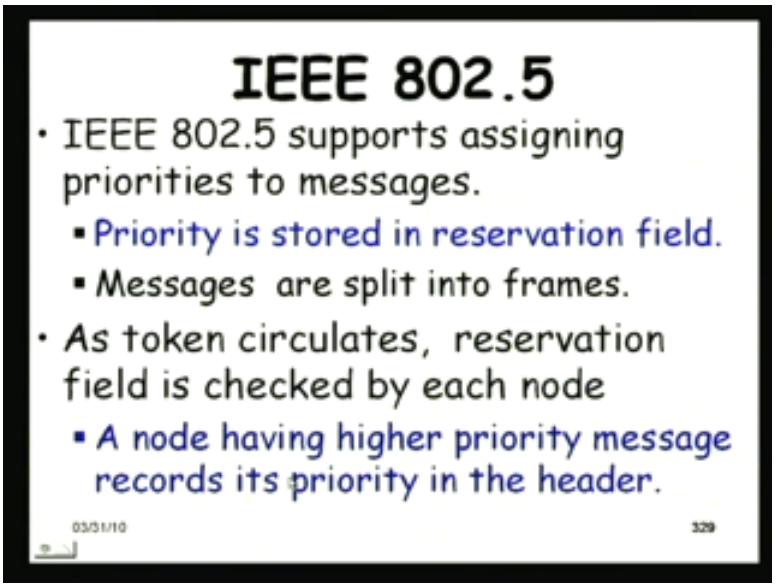
Now, let us look at the IEEE 802.5 which is a priority -based token ring protocol let us seen that token ring has some advantages over a bus protocol. So, here the header of the token contains two fields: a reservation field and a mode field. This was a protocol which was actually designed for a manufacturing automation situation. So, to start with design for real-time applications. So, there are the reservation and mode fields and the token alternates between a reservation mode and a free mode. In the reservation mode the different stations that are on the network the different nodes they reserve based on the priority of the message they have they make a reservation. So, one bit will indicate the mode bit whether it is a free mode or a reservation mode and if it is in the reservation mode then the different nodes on the network as the token rotates they would check the reservation field here and then if they find that a lower value compared to their reservation their message priority is there. Then they would replace this with their own message priority and their own ID.

Now, as this token reaches the sender it would find that a reservation has been made by a node which has a higher priority message and then it would put it in the free mode and without any payload it will just send the token. Then the node which made the reservation will find that its own values are there .Then it will just capture the token is that. The way it works is that the stations make reservation in the reservation mode when the mode bit in the reservation mode they make a reservation, if they find that the

reservation field has a lower priority. Then the priority of the message that they have then they write their own priority value here, but just writing the priority value here does not indicate that they will get the token next time .Because another node might have a higher priority message which will replace it.

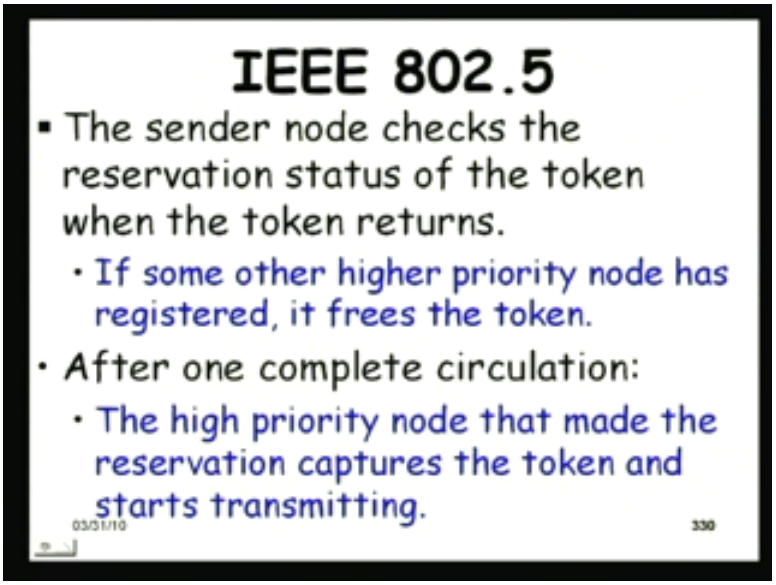
As the token goes around and comes back to the sender if it finds that the reservation field is changed has a higher priority than the priority of the message it has then it will put in the free mode without any payload and transmit the token, but if finds that either nobody has made the reservation or the priority value is less than it the priority of the message that it itself has then it would not change the mode to free mode. It will keep it in the reservation mode and continue to transmit and of course, if it finds that there are no messages to send. It will just put it in the free mode and transmit.

(Refer Slide Time: 36:26)



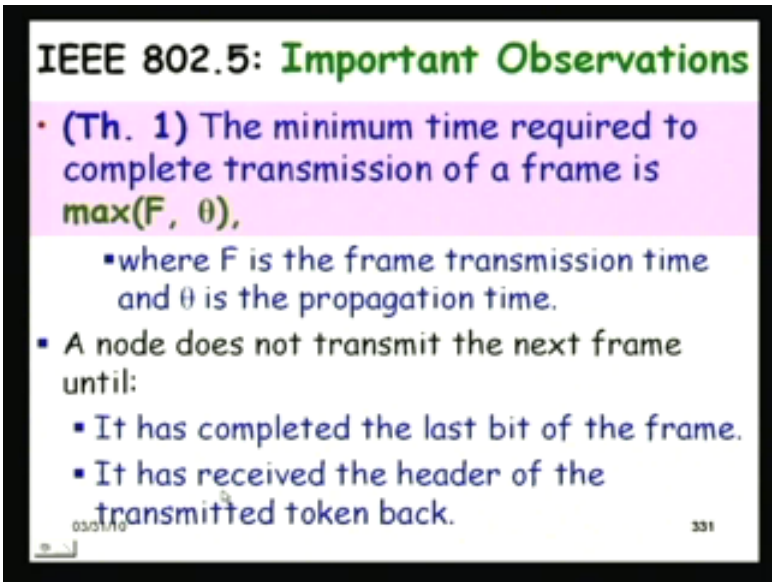
So, here we have assumed that again the messages are assigned priorities. So, global priority protocol and priority value is stored in the reservation field different nodes examine the field and if they have a higher priority they store that in the field. The Messages are split into frames and attached are put in the payload with the token as the token circulates the reservation field is checked by each node and if a node has a higher priority message then it records its priority in the header in the appropriate field the reservation field it will write the priority value.

(Refer Slide Time: 37:32)



And once the token is back to the sender it checks the reservation status and if there is a higher priority node which has registered its priority value in the reservation field then it puts the token in the free mode and circulates .After one complete circulation in the reservation mode it is determined about if there is a higher priority message and then it would be put in the free mode and it would start transmitting is that looks .

(Refer Slide Time: 38:16)

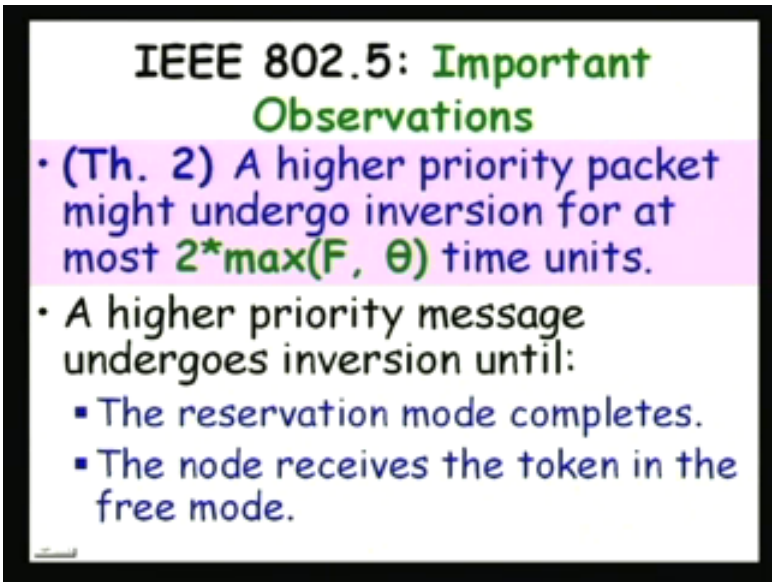


Now, we can make some observations based on which we will derive some results. One is that stated in the form of a theorem. The minimum time required to complete transmission of a frame is maximum of F and theta, where F is the frame transmission time. So, basically a token needs to be transmitted. This is the transmission time and then

theta is the propagation time. So why do we say this, that the time required to complete transmission of a frame is maximum F comma theta is that a node would not transmit the next frame until, it has completed the last bit of the current frame being transmitted. So, that accounts for F.

And also it should have received the header of the transmitted token back if it has not received the header of the token back then it would wait ,because let me just ask you why should it wait? Some other higher high value in a (( )) Exactly. It would not know that if there is a higher priority message waiting to be transmitted and then it should relinquish is not it. So, that is the reason why it should wait for the header to reach back. So, depending on what are the values of F whether it is a very slow network where transmission takes more time bandwidth is very low which dominates the transmission propagation time or whether the propagation time is large it is a large network depending on that the transmission time of a frame will depend how frequently the frames will keep on getting transmitted is that looks.

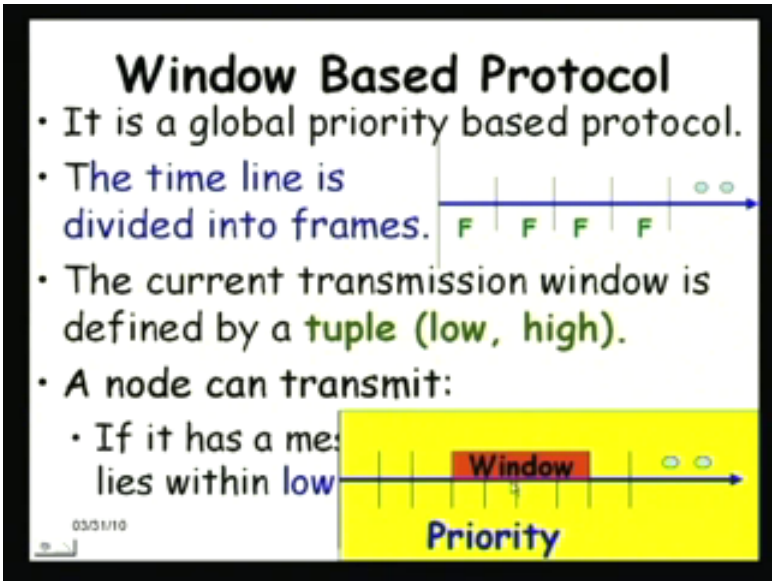
(Refer Slide Time: 40:23)



We will derive some results based on these observations another observation which will state in the form of theorem two is that if a node has a higher priority message which has arisen at a node. Then it might undergo a priority inversion for at most two into max F comma theta time units. The reason is that a higher priority message will undergo inversion until the reservation mode completes and for the reservation mode to complete it will take max F comma theta and also the node receives the token in the free mode. The token needs to be transmitted and in the worst case it can be just away from the node

and it will take max F comma theta. So, since both of these take max F comma theta then the time for which a message can undergo inversion would be two into max F comma theta.

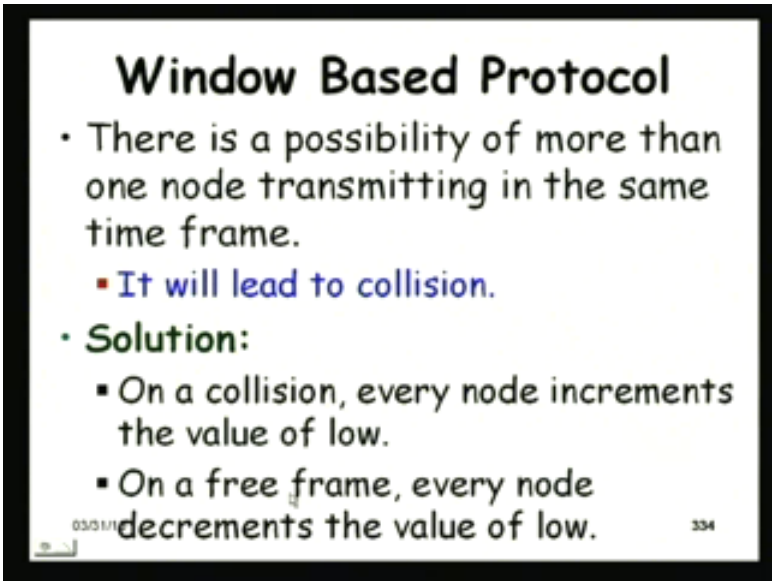
(Refer Slide Time: 41:47)



These are two important results based on which we will derive some results just short while from now we were just looking at the overview of the protocols . We will see the working of 802.5 more detail. Another category of protocol is the window based protocol .Here again it is a global priority protocol and again the time is divided into frames just observe that almost every protocol global priority protocol that we discussed the time is divided into frames and at that the start of the frame there is arbitration where it is decided which is currently the highest priority message and then that is followed by a transmission.

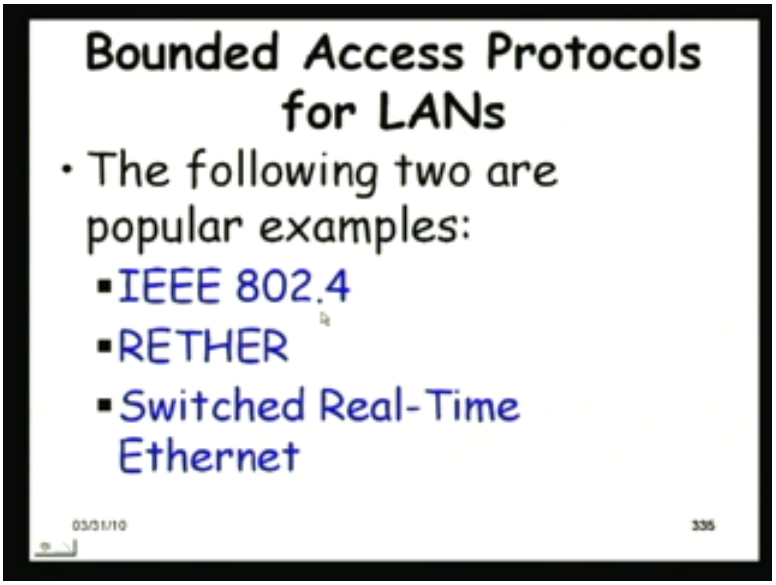
So, here again the time line is divided into frames and there is a concept of a transmission window which is defined by some low and high priority messages. So, every node maintains these two values the low value and the high value every node. So, a node can transmit if it has a message whose priority lies within low and high priorities, but so can be with many other nodes can find at the start of the F frame that they have a message between the low and high priority value. This the window that is defined here . This is the priority values some low value and high value and if a node finds that its priority falls within this window it will start transmitting, but the only problem is that there can be several nodes which might find that they have a message whose priority is fits into the window and then there will be a collision.

(Refer Slide Time: 43:58)



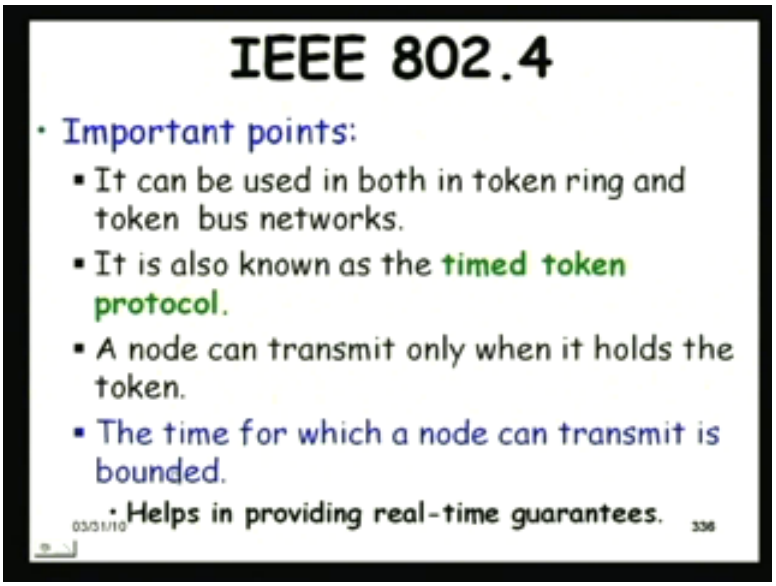
There is the possibility of more than one node transmitting at the same time which will lead to a collision and on a collision each node increments the value of low. So, that some of the nodes will get caught off they would not transmit, but still there can be another collision and then again the low value will keep on increasing until the highest the node with the highest message is found out and on a free frame if nobody is transmitting then the low value will be decremented, because there are no higher priority messages may be some lower priority messages might be waiting we do not know. Each time if there is a free frame each node decrements the value of low of course, each low message will transmission of a low message will be preceded by a free frame, but that is the inefficiency of the protocol.

(Refer Slide Time: 45:11)



Now, we will look at some bounded access protocols mainly three protocol we will look at .One is 802.4. IEEE 802.4 we had seen that 802.5 was a priority based protocol and then the RETHER and a switched real-time ethernet.

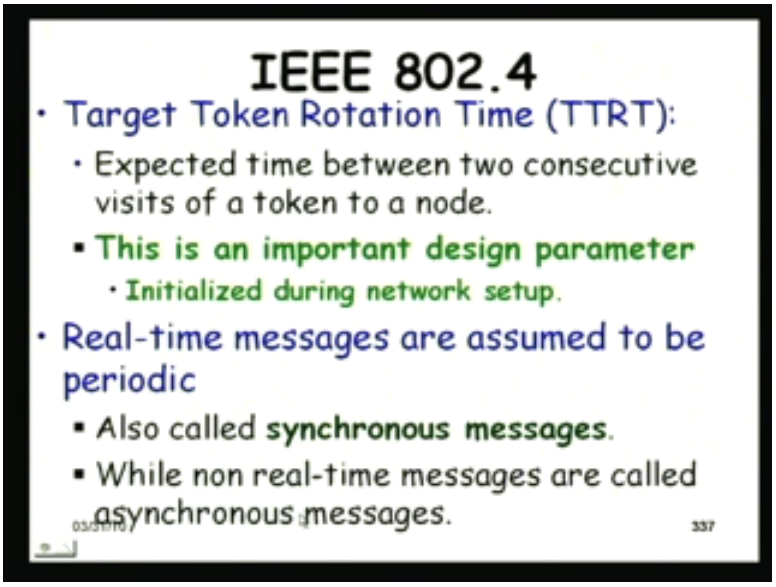
(Refer Slide Time: 45:39)



Let us look at the IEEE 802.4. Now, the way it works let us try to understand the important points in its working. So, the protocol can work both in a token ring or a token bus network and another name for this protocol is the timed token protocol .Here a node transmits only when it holds a token and once it gets the token it starts to transmit, but there is a bound for the time for which it can transmit. This bound helps in providing real-time guarantees, because in the worst case every node will hold for their own

bounds. So, that will give us an indication of how long a message can have to wait in the worst case.

(Refer Slide Time: 46:57)

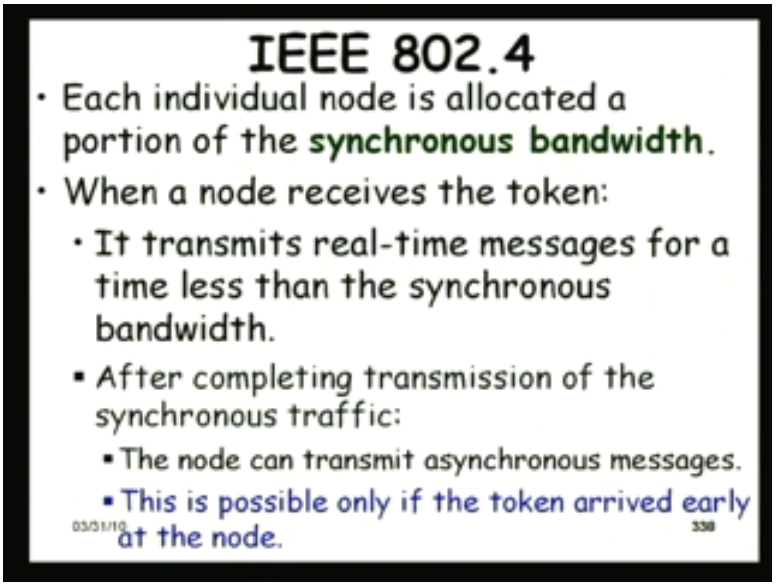


One important parameter here is the Target Token Rotation Time TTRT .The TTRT is the expected time between two consecutive visits of a token to a node. So, when a token arrives at a node starts to transmit and TTRT is the difference in time between two consecutive visits of a node. What does it indicate, it indicates how long or how much message might have to wait and just see here that it is the expected time TTRT is the expected time. What is the expected value for which a node would have to wait before it can start transmitting or for how long the expected value of inversion that a high priority or let us say a message that might need to get transmitted even if the message is there its high priority it cannot be transmitted until it gets the token and therefore, TTRT would be a expected value of that, but we know that for real-time situations expected value is not really a very useful thing.

Now let us see how we use this to get the worst case value. So, the TTRT is an important design parameter. It is initialized during the network setup and the value of the TTRT it selected based on the messages that are there with the different nodes the amount of data that needs to be transmitted. We will solve some problems based on this depending on what is the size of the messages at different nodes and here the Real-time messages are considered to be periodic and they are called as the synchronous messages and the other types of messages the non real-time are called as asynchronous messages and the

asynchronous messages get transmitted only when there are no synchronous messages is that.

(Refer Slide Time: 49:36)



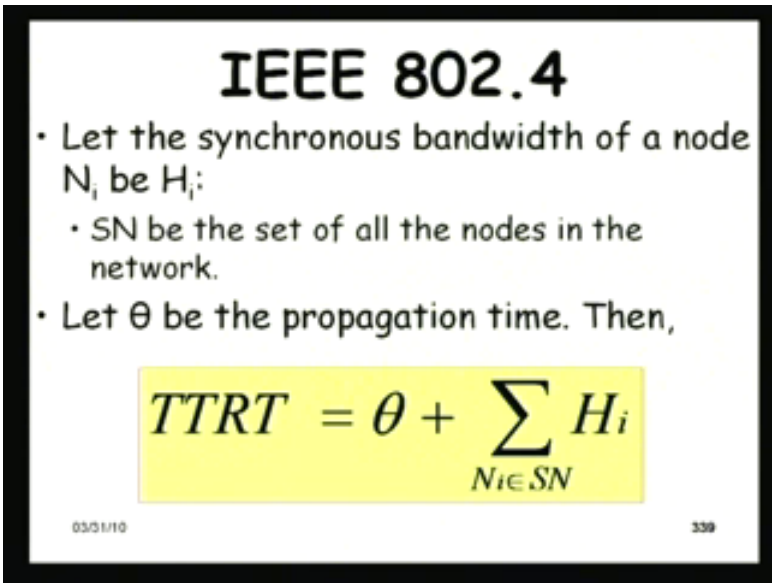
The real-time messages are called as synchronous messages. Only one there are no synchronous messages the asynchronous messages will be transmitted. So, once a node gets hold of the token it will first try to complete the synchronous messages and then start transmitting the asynchronous messages, until its time bound is over and of course, if it does not have any real-time messages at that instant it would transmit asynchronous messages. When a node receives a token transmits real-time messages for if it finds that there are free time after transmitting the synchronous bandwidth. So, it starts transmitting the synchronous messages and if it finds that there are some time left out after transmitting the synchronous traffic then it starts transmitting the asynchronous messages.

Of course, you might ask here that see if it is a CBR kind of traffic very predictable traffic. Then how will the asynchronous I mean why should it take less time , because each time it should take the same amount of time to transmit ,because similar traffic will be there with this. Then where is the time to transmit the asynchronous messages .I mean how will it transmit complete its transmission earlier. Would anybody like to answer, how this can arise , anybody would like to answer anything, the answer is that even the tasks the messages which are a periodic .We need to consider them as synchronous we need to reserve bandwidth them and consider synchronous messages if we need to transmit that with guarantee. So, that those messages which are a periodic they might

occur or might not occur and we will have a free slot. I mean it will complete transmission earlier it does not complete transmission early depending on whether that specific message has arisen or does not has not raised in that period. A periodic time means the allowing for even asynchronous. I mean the ones that are imported the critical messages for example, an alarm.

For that we have to have a some we have to consider that as a synchronous traffic otherwise that node will not have any time left for it . So, once after completing transmission of the synchronous traffic the node starts to transmit the asynchronous messages and it can get the token early see even if there are only the synchronous traffic with it. There are synchronous only CBR traffic you and no a periodic traffic which are of high priority. Then how will it transmit the asynchronous messages. The idea is that sometimes it will receive the token early. It might so happen that its previous node neither had asynchronous nor synchronous traffic, because it had made reservation for some a periodic traffic which does not did not arise it neither had any asynchronous traffic to transmit it just release the token earlier. So, in that case the token can arise early at a node.

(Refer Slide Time: 54:04)



So, we are running out of time now we need some time to discuss this about some results here that for how long. I mean how do we distribute the time the TTRT is the design parameter we had said is the expected time for a token to arrive at a node and theta is the propagation time. So, how do we distribute this time of the token rotation among the different nodes in the network and based on that we will do few problems on given a set

up where we know that what are the nodes and what kind of traffic they have what kind of messages they need to transmit how do you assign slots to them. So, this we will discuss in the next lecture. So, I will stop here today.