

**Real Time Systems**  
**Professor Durga Prasad Mohapatra**  
**Department of Computer Science and Engineering**  
**National Institute of Technology Rourkela**  
**Lecture 56**  
**QoS Model and Soft Real-Time Communication in a LAN**

Good morning to all of you, today we will discuss the different quality of service models and then we will discuss about how soft real time communication occurs in a LAN.

(Refer Slide Time: 0:34)

**CONCEPTS COVERED**

- Important Non-Work Conserving Service Disciplines
- QoS Models
- Integrated Services
- Differentiated Services
- Soft Real-Time Communication in a LAN

The slide features a video inset of Professor Durga Prasad Mohapatra in the bottom right corner. At the bottom of the slide, there are logos for NITRR (National Institute of Technology Rourkela) and NPTEL (National Programme on Technology Enhanced Learning).

But before going to that something was left from last class, we have discussed about the different service disciplines, so we have discussed only the work conserving service discipline in the last class, the other category is non-work conserving service disciplines, that first let us finish up some important non-work conserving services principles we will see first, then we will see the different quality of service models. Most popular quality of service models we will see integrated services and differentiated services and then finally we will look at the soft real time communication in a LAN.

(Refer Slide Time: 1:06)

**KEYWORDS**

- Weighted Fair Queuing
- Internet Engineering Task Force
- RSVP
- Differentiated Service
- Traffic Smoothing

These are the keywords we will use here.

(Refer Slide Time: 1:11)

**NON-WORK CONSERVING DISCIPLINE**

- Each packet is assigned an eligibility time after which a packet is eligible to be serviced.
- NWCD can tightly bound delay jitter, whereas, WCD cannot bound delay jitter tightly.
- Ex: Jitter-EDD, Stop-and-Go, HRR, RCSP.

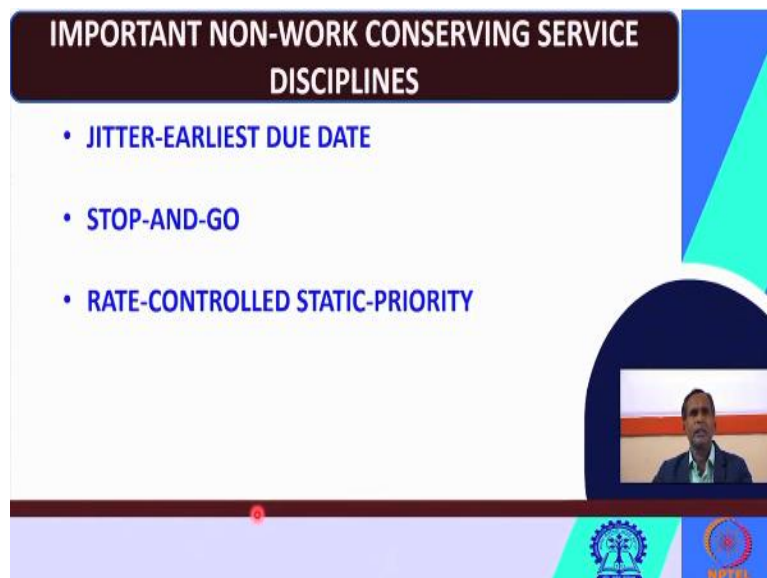
Now we will start with the non-work conserving discipline. I have already told you what is work conserving service discipline, what is non-work conserving service discipline, last class we have discussed, the differences I have already told you, still I am quickly revising it what we have discussed in last class about non-work conserving discipline.

I have told that in non-work conserving discipline each packet is assigned an eligibility time, in a non-work conserving service so each packet is assigned an eligibility time, what do you mean by eligibility time, it is the time after which a packet becomes eligible to be serviced before that time the packet cannot be taken up for schedule.

So only when that eligibility time occurs then only it can be serviced, so in non-work conserving discipline every packet is assigned an eligibility time after which a packet is eligible to be service, we have already discussed in last class. One major difference I have told in case or between WCD and NWCD you have already known that in NWCD that means non-work conserving service discipline it can tightly bound delay jitters.

So the delay jitter can be tightly bound in case of NWCD, but WCD that means work conserving discipline it cannot bound the delay jitter very tightly, this difference I have already told you in the last class. So here you will see how this NWCD can tightly bound the delay jitter. So examples of non-work conserving discipline are jitter EDD, stop and go, HRR and RCSP, we will take quickly two three important the disciplines now.

(Refer Slide Time: 2:59)



**IMPORTANT NON-WORK CONSERVING SERVICE DISCIPLINES**

- JITTER-EARLIEST DUE DATE
- STOP-AND-GO
- RATE-CONTROLLED STATIC-PRIORITY

We will start with so these are the important non-work conserving service disciplines Jitter-Earlier Due Date or Jitter EDD, Stop and Go, Rate Controlled Static Priority or RCSP.

(Refer Slide Time: 3:08)

**JITTER-EARLIEST DUE DATE**

- After a packet has been served at a server, a field in its header is stamped with the difference between its deadline and the actual finishing time.
- A regulator at the entrance of the next server holds the packet for this period before it is made eligible to be scheduled.
- It provides delay jitter bounds.

**IMPORTANT NON-WORK CONSERVING SERVICE DISCIPLINES**

- JITTER-EARLIEST DUE DATE
- STOP-AND-GO
- RATE-CONTROLLED STATIC-PRIORITY

We will start with Jitter earliest due date, so how this came in to works, so in this scheme after a packet has been served at a server, then what happens, a field in its header is created, in the header of that field, in the header of the packet a field is created and that field what does it contains?

It contains this timestamp; it is marked with the time and what is the time? It is the time difference between its deadline and it the actual finishing time, what is its schedule deadline and at the actual time when it is finished, what is the difference, that time is maintained, where? In the field in the header of the packet.

In the jitter earliest due date after every packet has been solved at a server, then in the header of that packet a field is stamped, a field maintained with what, with a time value which

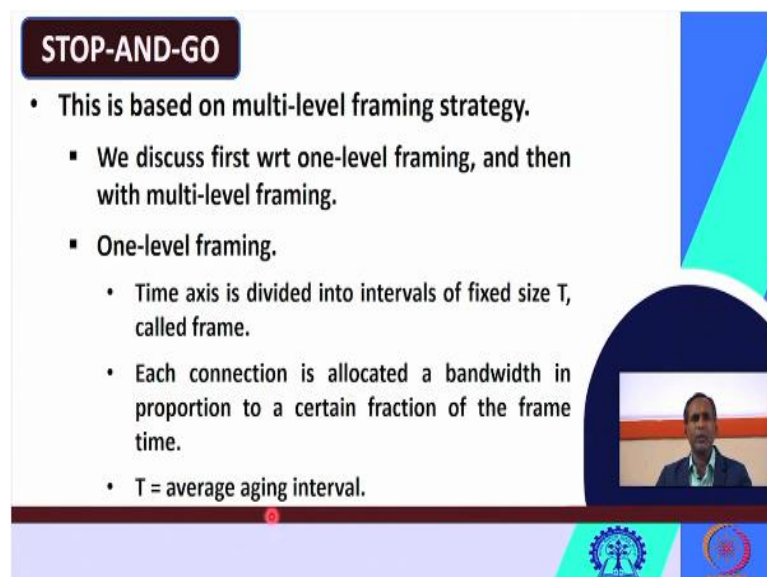
represents the difference between its deadline and what is the actual finishing time, then what happens?

A regulator at the entrance of the next server, so the servers are arranged in the different switches maybe hierarchically, a regulator at the entrance of the next server it holds the packet, it just keeps holds the packet for how much time? For this period, which period? The period which is equal to the difference between its deadline and the actual finishing time.

So a regulator or the entrance of the next server maybe present in the next layer of the switch, it holds the packet for this differential period of time before it is made eligible to be schedule, so before that what packet becomes eligible to be schedule or served a regulator at the entrance of the next server maybe present in the next layer of the switch, it holds the packet for this differential amount of time or for period after that the packet will be eligible to be schedule or it will be eligible to be served.

I have already told you that these kind of, this coming under what, non-work conserving service discipline and I have already told you that non-work conserving disciplines they can tightly bound the delay jitter. So this one since jitter earliest due date it is coming under NWCD, it also provides delay jitter bounds, it bounds the delay jitter very tightly. Now, this is all about the fundamentals of the jitter earliest due date.

(Refer Slide Time: 6:50)



**STOP-AND-GO**

- This is based on multi-level framing strategy.
  - We discuss first wrt one-level framing, and then with multi-level framing.
  - One-level framing.
    - Time axis is divided into intervals of fixed size  $T$ , called frame.
    - Each connection is allocated a bandwidth in proportion to a certain fraction of the frame time.
    - $T$  = average aging interval.

We will quickly see about the next scheme, that is stop and go, so this scheme is based on multi-level framing strategy, so here the concept of multi-level framing strategy is used. Let us

see, so since it addresses multi-level framing strategy let us first start with this scheme with respect to the one level framing and then we will discuss with multi-level framing.

In one level framing in stop and go scheme what is happening, that the time axis is divided into some intervals of fixed size and maybe of duration of  $T$ , in one level framing the time axis is divided into some intervals of fixed size maybe  $T$  duration which is called as frame.

Then each connection is allocated a bandwidth, so in stop and go scheme each connection is allocated a bandwidth in proportion to a certain fraction of the frame time, so then every connection is allocated a bandwidth, how? The connection is allocated bandwidth in proportion to a certain fraction, certain portion of the frame time. So how  $T$  will you calculate, is  $T$  is equal to the average aging interval.

(Refer Slide Time: 6:57)

**ONE LEVEL FRAMING IN STOP-AND-GO**

- At each switch, the arriving frame of each incoming link is mapped to the departing frame of the output link by introducing a constant delay  $\theta$ , where  $0 \leq \theta < T$ .
- The transmission of a packet that has arrived on any link  $l$  during a frame  $f$  should always be postponed until the beginning of the next frame.
  - The server remains idle even if there are packets queued for transmission.

**STOP-AND-GO**

- This is based on multi-level framing strategy.
  - We discuss first wrt one-level framing, and then with multi-level framing.
  - One-level framing.
    - Time axis is divided into intervals of fixed size  $T$ , called frame.
    - Each connection is allocated a bandwidth in proportion to a certain fraction of the frame time.
    - $T$  = average aging interval.

Let us first little bit see about one level framing, then we will see about multi-level framing. So in one level framing what happens at each switch there are two kinds of frame, the arriving frame and the departing frame, arriving frame means the frame which is just arriving and departing frame means the frame which is just departed.

So at each switch the arriving frame of each incoming link it is mapped to what? It is mapped to the corresponding departing frame of the output link, how? By introducing a constant delay, maybe that constant delay is  $\theta$ , where  $\theta$  lies between 0 to  $T$  and  $T$  I have already told you to the average aging interval.

So, here in one level framing at every switch the arriving frame of each incoming link, it is mapped to the corresponding departing frame of the output link by what, introducing a constant delay, which is represent as  $\theta$  and  $\theta$  lies between 0 and,  $T$ ,  $T$ , the average aging interval.

Now, the transmission of a packet, which has just arrived on any link  $I$ , suppose now a packet has arrived on your link  $I$  during the frame  $f$ , then what should be carried out? It should be postponed, until what time, until the beginning of the next frame.

That means what, the packet can be served, can be scheduled only of the beginning of the next frame so if it arrives, exactly at the starting of the frame, then it will be immediately processed, if just a frame had started, and it has come after that, then that packet has to wait till the beginning of the next frame because that can be processed at the beginning of the next frame.

In between, it cannot be processed. So ultimately what may happen, if a packet has arrived just a frame has been started, then the server has to remain idle, even if the packet, there are some packets, they are queued in the queue for transmission, but they have to wait the server has to remain idle because the packets can be processed only at the beginning of the frame. So here the server has to remain idle, even if there are so many packets are waiting for transmission.

I am quickly repeating again, the transmission of a packet that has already arrived on any link  $I$  during a frame, that means during the frame period, then it should always be postponed until what time, until the beginning of the next frame because the packets can be processed, they can be transmitted only at the beginning of a frame, hence the server may remain idle, even if there are so many packets, they are queued from transmission.

So now here I have told you, here we are using a framing strategy, is not it? So, I have already told this stop on go scheme is based on some framing strategy, we are dividing the time axis into a number of frames. So now there is a little bit problem.



(Refer Slide Time: 10:09)

**STOP-AND-GO**

- Framing strategy introduces the problem of coupling between delay bound and bandwidth allocation granularity.
  - The delay of any packet at a single switch is bounded by **two frame times**.
  - To reduce the delay, a smaller  $T$  is desired.
- Assuming a fixed packet size  $P$ , minimum granularity of bandwidth allocation is  $P/T$ .

**STOP-AND-GO**

- This is based on multi-level framing strategy.
  - We discuss first wrt one-level framing, and then with multi-level framing.
  - One-level framing.
    - Time axis is divided into intervals of fixed size  $T$ , called frame.
    - Each connection is allocated a bandwidth in proportion to a certain fraction of the frame time.
    - $T$  = average aging interval.

The framing strategy introduces the problem of coupling between delay bound and bandwidth allocation granularity. So this framing strategy that we have told you that the time axis is divided into so many frames, then what is the consequence let us see. The framing strategy it introduces the problem of coupling between what, between the delay bound and the bandwidth allocation granularity.

Now if we see the delay of any packet at a single switch is bounded by two frame, two frame times, please see the delay of any packet at a single switch, it is bounded by two frame times, why, I have already told you, any packet can be served at the beginning of the frame. If they arrive after the frame has been started, then it has to wait till the next till the beginning of the next frame, so the delay of any packet at a single switch is bounded by two frame size.



To reduce this delay, what is desired, you require a smaller value of  $T$ , to reduce this delay a smaller value of  $T$  is desirable or a smaller value of  $T$  is desired. Let us assume that the packet size is fixed assuming or assuming a fixed packet  $p$ , how can compute the minimum granularity bandwidth allocation?

The minimum granularity can be computed as  $p$  by  $T$  because  $T$  is what,  $T$  is have already told you  $T$  is the average aging interval, this is your fixed size. So assuming that fixed packet size, packet size is  $p$  now, so and the interval you know  $T$ , so the minimum granularity bandwidth allocation is computed as  $P/T$ .

(Refer Slide Time: 11:51)

**STOP-AND-GO cont...**

- To have more flexibility in allocating bandwidth or smaller bandwidth allocation granularity, a larger  $T$  is preferred.
- **Therefore, low delay bound and fine granularity of bandwidth allocation cannot be achieved simultaneously.**

**STOP-AND-GO**

- Framing strategy introduces the problem of coupling between delay bound and bandwidth allocation granularity.
  - The delay of any packet at a single switch is bounded by **two frame times**.
  - To reduce the delay, a smaller  $T$  is desired.
- Assuming a fixed packet size  $P$ , minimum granularity of bandwidth allocation is  $P/T$ .

Now, I have already told you to reduce the delay you require smaller value of  $T$ , or a smaller value of  $T$  is desired. On the other hand, we should get what flexibility in allocating the

bandwidth or smaller bandwidth allocation granularity is required. So if we require a smaller bandwidth allocation, what you required,  $T$  should be larger, I have already told you that the minimum granularity of bandwidth is  $P/T$ . So, how, what this granularity or this granularity bandwidth can be minimum, because the minimum granularity of the bandwidth is equal to  $P/T$ .

So granularity of bandwidth can be minimum if  $T$  can be maximum, so we require a larger value of  $T$ , in order to get the flexibility in allocating the bandwidth, so in order to have more flexibility in allocating bandwidth or in order to get smaller bandwidth allocation granularity, a larger  $T$  is preferred,  $T$  value should be higher. So this I have already told you because this minimum granularity bandwidth is equal to  $P/T$ , if you require granularity minimum, obviously,  $T$  has to be maximum in order to have this smaller bandwidth allocation granularity we require larger values of  $T$ .



So now, one side we have seen, we should require lower delay, and to reduce the value of delay, we require a small  $T$  but in order to have smaller bandwidth, we require larger  $T$ , so what does it imply, this implies that load delay bounds and fine granularity of bandwidth cannot be achieved simultaneously because these are contradicting now.

For lower reduced delay we require a small value of  $T$ , but for we require smaller bandwidth, for smaller bandwidth, we require larger value of  $T$ . So what does it say, those two things at the same time how  $T$  can be smaller or as well it is larger? So, these are contradicting both the things they are not achievable simultaneously, so therefore both the parameters, such as the low delay bound and the finer granularity of bandwidth allocation, they cannot be achieved at the same time, they cannot be achieved simultaneously.

(Refer Slide Time: 14:09)

## MULTI-LEVEL FRAMING IN STOP-AND-GO

- The time axis is divided into a hierarchical framing structure.
- For a n-level framing, suppose the frame sizes are  $T_1, T_2, \dots, T_n$ , with  $T_{m+1} = K_m \cdot T_m$  for  $m=1, \dots, n-1$ .
  - Then, packets on a level p connection need to use frame size  $T_p$ .
    - Level p packets which arrived at an output link during a  $T_p$  frame will not become eligible for transmission until the start of next  $T_p$  frame.

Now let us quickly look at the multi-level framing in stop and go scheme. So here the time axis is divided into a hierarchical framing structure, that is why it is named multi-level, multiple levels are there, the time axis is divided into a hierarchical framing structure. Now let us assume that for n level framing, suppose the frame sizes are denoted as  $T_1, T_2, T_n$  etcetera.

For n level framing, suppose there are n level of frames, suppose the frame sizes are  $T_1, T_2$  up to  $T_n$ , where  $T_{n+1}$  is equal to  $K_m$  into  $T_m$  and m can take the values from 1 to n minus 1 and  $K_m$  is a constant you can say, then the packets on a level p connection need to use frame size  $T_p$ .

So a level p packets which arrived at output link during a time frame  $T_p$ , it will not be eligible for transmission until the start of the next  $T_p$  frame, I have told you, a packet it can, if it arrives after a frame has been started, it cannot be immediately taken into account, it cannot be immediately process, it has to wait till the beginning of the next frame.

So let us quickly revise, for a n level framing, suppose the frame sizes are  $T_1, T_2, T_n$  etcetera, where  $T_{n+1}$  is equal to how much  $K_m$  into  $T_m$  where m can take the values 1 to n - 1, and  $K_m$  with a constraint, then the packets on a particular level p connection need to use what frame size, they need to use trim size  $T_p$ , this indicates that level p packets which arrive at an output link during a  $T_p$  frame because already  $T_p$  frame has been started.

So level p packets which are arrived at an output link during  $T_p$  frame it will not become eligible immediately. When it will become eligible for transmission? It will become eligible for transmission immediately after the start of the next  $T_p$  frame, then that will be become eligible, this is how the multi-level framing in stop and go works.

(Refer Slide Time: 16:28)

**TWO-LEVEL FRAMING**

$T_2$  frame

$T_1$  frame

$T_2 = 3 * T_1$

**MULTI-LEVEL FRAMING IN STOP-AND-GO**

- The time axis is divided into a hierarchical framing structure.
- For a n-level framing, suppose the frame sizes are  $T_1, T_2, \dots, T_n$ , with  $T_{m+1} = K_m \cdot T_m$  for  $m=1, \dots, n-1$ .
  - Then, packets on a level p connection need to use frame size  $T_p$ .
    - Level p packets which arrived at an output link during a  $T_p$  frame will not become eligible for transmission until the start of next  $T_p$  frame.

A small example I have told you. So here is p value we can, this  $T_{m+1}$  here suppose I have taken m equal to 1, so that means if  $T_2$  is equal to  $K_1$  into  $T_1$ , is not it? If m is equal to 1, I will put here, then  $T_2$  is equal to how much  $K_1 * T_1$ . So I have plus 1 here, so  $T_2$  I have taken.

And here you can see, this is the larger frame, this we are discussing what, multi-level framing, multiple levels are they are, this is the, on the higher level  $T_2$  frame and then there are so many smaller frames, we call them  $T_1$  frame, you can see, with in the  $T_2$  frame how many  $T_1$  frames are there, one, two and three, so  $T_2$  can be expressed as 3 into  $T_1$ , this also satisfy this equation,  $T_{m+1}$  is equal to  $K_m$  into the  $T_m$ , where m value is equal to 1. So here you can see, this is

larger frame T2 frame, it is divided into multiple levels, there are three T1 frame, so T2 can be expressed as 3 into T1, this is an example of two-level framing.

(Refer Slide Time: 17:32)

**RATE-CONTROLLED STATIC PRIORITY (RSCP)**

- **Drawbacks of previous service disciplines:**
  - Sorted priority service disciplines such as WFQ are complex and difficult to implement.
  - Framing strategy (such as Stop and Go) suffers from the dependencies between queuing delay and granularity of bandwidth.
- **In RCSP, the function of rate control has been decoupled from delay control.**

The slide features a dark blue header with the title in white. The main content is in black text on a white background. A small video inset shows a man speaking. Logos for IIT Bombay and NPTEL are visible at the bottom right.

We will go to now another the scheme that is rate control static priority scheme or RSCP. So let us quickly analyze the drawbacks of the previous service discipline. So the drawbacks of the previous service discipline are as follows. So, if you will use the sorted priority service disciplines such as a WFQ we have seen earlier in the last class, these are very much complex and difficult to implement, if we will consider the framing strategy, where we use strategy just right now we have used stop and go scheme.

So in framing strategies such as stop and go scheme so it suffers from the dependencies between queuing delay and granularity of bandwidth, so it suffers from what, it suffers from the problem, it suffers from the dependencies between what, queuing delay and then granularity bandwidth. So, in order to overcome and these are the problems of the previous schemes.

So, in this RCSP that means rate controlled static priority, the function rate control has been decoupled from delay control. So in order to overcome these problems in this scheme, the function of the rate control we have already discussed what is rate control in earlier classes, here the function of the rate control, it has been decoupled, decouple from whom, decoupled from the delay control.

(Refer Slide Time: 18:51)

### RATE-CONTROLLED STATIC PRIORITY cont...

- An RCSP server has **two** components:
  - A **rate-controller** – It consists of a set of regulators for each connection.
    - Each regulator is responsible for shaping the input traffic into a desired pattern.
    - When a packet arrives, its eligibility time is calculated and is assigned to the packet.
    - The packet is held at the regulator until its eligibility time expires, after which it is handed to the scheduler for scheduling and transmission.



### RATE-CONTROLLED STATIC PRIORITY cont...

- A **static priority scheduler** – It is implemented as a multilevel FCFS queue which serves packets in order of their priorities.
- Regulators can be of **two** types:
  - **Rate-jitter (RJ)**
    - It controls **rate-jitter** by **partially** reconstructing the traffic.
  - **Delay-jitter (DJ)**
    - It controls **delay-jitter** by **fully** reconstructing the traffic.



So how does it work? An RCSP server has two important components, a rate controller, and a static priority controller and this rate controller it consists of a set of regulators for each connection. So for every connection there is a set up regulators, it is available and what this regulator will do, each regulator is responsible for shaping the input traffic into digital pattern. Last class we have already discussed about traffic shaping, so policing and shaping we have discussed in the last class.

So here each regulator is responsible for shaping the input traffic into a desired pattern. So here what each regulator do, each regulator will shape the input traffic into a desired in to specific pattern. When a packet arrives then eligible time is calculated, whenever a packet arrives, its eligibility time is calculated and it is assigned to the packet, that means that much time it has to wait. So when a packet arrives, it is eligibility the time is calculated and it is assigned to the packet.



Now, the packet is held at the regulator, why, because the eligible time, it has not met, till the eligible time is met, it has to be hold somewhere else, where it will be hold, it will be it will be held at the regulator. So the packet is held at the regulator until it eligibility time expires, and only after which it is handed to the scheduler for scheduling and transmission.

So, only when the eligibility time will be expired, it will be handed over to the scheduler for scheduling and then that packet will be transmitted. So we have seen and that in a RCSP server there are two components, a rate controller and this static priority scheduler and what is the job, and rate controller consists of so many regulators, what is the function of regulator we have just discussed.

Now let us see what is the job of static priority scheduler. The static priority scheduler I hope you have already studied static priority scheduler etcetera in task scheduling, so a static priority scheduler it is implemented as a multi-level FCFS queue, the static priority scheduler is normally it implemented as a multi-level first come first serve queue, which serves the packet in the order of their priorities.

So every packet is having some priority so this scheduler it serves the packets in order of their priorities, these regulators I have already told you that a rate controller consists of some regulators, so the regulators are up to two types, one is the rate jitter and the other is delay jitter.

So what does the rate jitter do, it controls the rate jitter as its name suggests rate jitter regulator, it controls the rate jitter by partially reconstructing the traffic. So it partially reconstructs the traffic in order to control the rate jitter or it control the rate jitter by partially reconstructing the traffic.

And what does this delay jitter do, it controls the delay jitter, so the delay jitter regulator as its name suggests, it controls the delay jitter by fully reconstructing the traffic, please see the difference, here it controls the rate jitter by partially reconstructing the traffic, whereas the delay jitter it controls the delay jitter by totally, by completely, by fully reconstructing the traffic.

(Refer Slide Time: 22:17)

**RATE-CONTROLLED STATIC PRIORITY cont...**

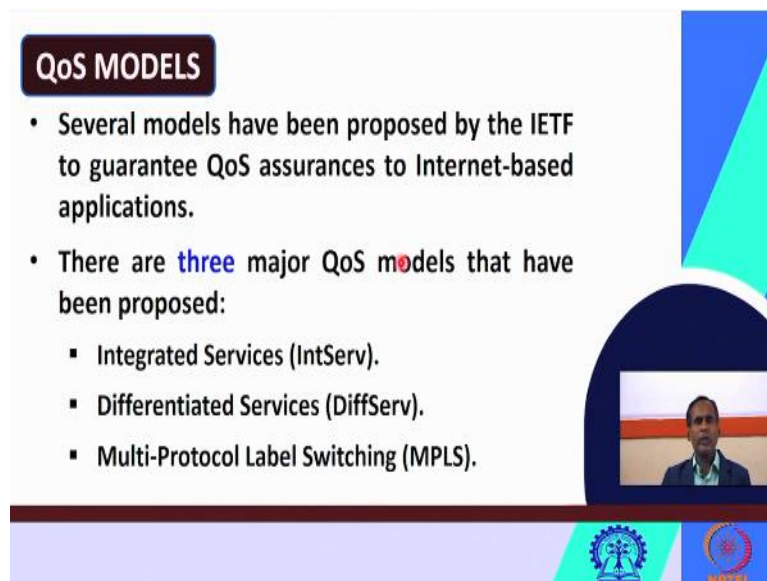
The diagram illustrates the architecture of a rate-controlled static priority scheduler. It consists of three main components: a Rate Controller, a Scheduler, and an Output stage. The Rate Controller, enclosed in a dashed box, contains multiple regulators (Regulator 1, Regulator 2, ..., Regulator H) that process the Input traffic. The output of these regulators is Regulated Traffic, which is then fed into the Scheduler. The Scheduler is a multilevel FCFS queue with Priority Levels (1, ..., N). The Scheduler outputs the traffic to the Output stage.

**RATE-CONTROLLED STATIC PRIORITY cont...**

- A **static priority scheduler** – It is implemented as a multilevel FCFS queue which serves packets in order of their priorities.
- Regulators can be of **two** types:
  - **Rate-jitter (RJ)**
    - It controls **rate-jitter** by **partially** reconstructing the traffic.
  - **Delay-jitter (DJ)**
    - It controls **delay-jitter** by **fully** reconstructing the traffic.

So now let us see how the rate controller and static priority scheduler they are arranged, they are arranged like this, this is a schematic diagram showing the different components in rate controlled static priority, you can see there are two important components, rate controller and scheduler, in rate controller, there are so many a set of regulators are there, a set up regulators is there, in scheduler I have already told you this is based on what, multi-level FCFS queue, so these are the priority levels, this is the scheduler and here regulated traffic is used, in this way, the rate control traffic priority scheme it looks like.

(Refer Slide Time: 22:52)



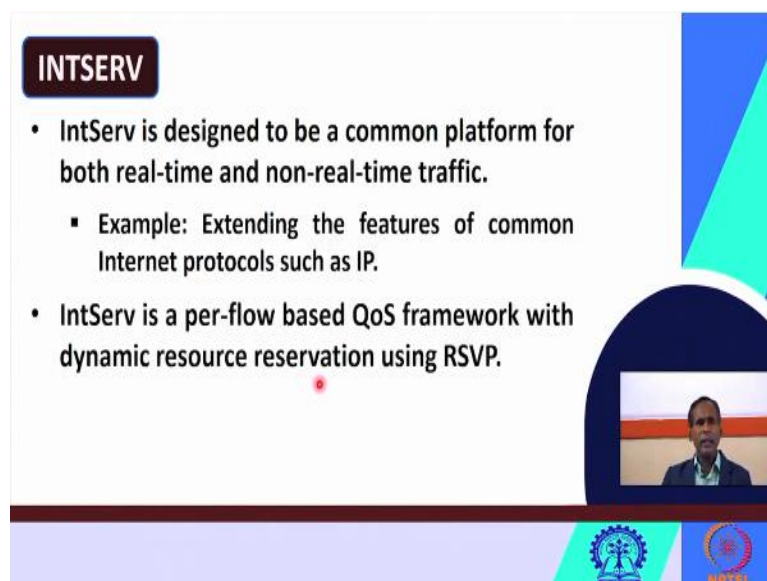
**QoS MODELS**

- Several models have been proposed by the IETF to guarantee QoS assurances to Internet-based applications.
- There are **three** major QoS models that have been proposed:
  - Integrated Services (IntServ).
  - Differentiated Services (DiffServ).
  - Multi-Protocol Label Switching (MPLS).

The slide features a video inset of a man in a suit speaking, and logos for IIT Bombay and NPTEL at the bottom.

Now we will go to the second aspect that is the different Quality of Service models available, very much important. Several modules have been proposed by IETF to guarantee the quality service, quality of service assurances to the internet based applications. There are three major quality of service models, which have been proposed by different researchers, one is the integrated services or in short, we call it as IntServ, another is differentiated services that we call DiffServ and one is multi-protocol label switching or MPLS. So today we will discuss these two most popular used models that is the IntServ and the DiffServ.

(Refer Slide Time: 23:30)



**INTSERV**

- IntServ is designed to be a common platform for both real-time and non-real-time traffic.
  - Example: Extending the features of common Internet protocols such as IP.
- IntServ is a per-flow based QoS framework with dynamic resource reservation using RSVP.

The slide features a video inset of a man in a suit speaking, and logos for IIT Bombay and NPTEL at the bottom.

Let us start with IntServ, IntServ is design to be a common platform for both the real time and non-real time traffic, both can be handled, real time and non-real time traffic can be handled

by IntServ. Example, is extending the features of the common Internet protocol such as IP. So if you are, if you are extending the features of the common Internet protocol such as IP version 4 or IP version 6, this is an example of IntServ.

This IntServ is per flow based quality of service framework, IntServ per flow based quality of service framework with a dynamic resources reservation using RSVP, we have already seen resources reservation protocol RSVP, so this IntServ is per flow based quality of service framework with dynamic resource reservation, here the resource can be reserved dynamically using what, using RSVP protocol, RSVP protocol we have already discussed in the some of the previous classes.

(Refer Slide Time: 24:28)

**INTSERV cont...**

- It includes **two** types of services targeted towards real-time traffic:
  - **Guaranteed**
    - These services provide an **upper bound on end-to-end queuing delay**. This model aims to support hard real-time requirements.
  - **Predictive**
    - These services provide **statistical guarantees only**.

This IntServ includes two types of services, targeted towards the real time traffic, what are the two types of services, guaranteed service, another is predictive service. In guaranteed service what is happening, these services provide an upper bound on end to end queuing delay, so these guaranteed services they provide an upper bound on what, on the end to end queueing delay. This model aims to support hard real time requirements.

So if you want to meet hard real time requirements, you should go for the guaranteed services. So guaranteed services or guaranteed service model aim at supporting hard real time requirements. What is predictive service, what this model does? These services provide statistical guarantees only. So predictive services they provide the statistical guarantees only.

(Refer Slide Time: 25:20)

**LIMITATIONS OF INTSERV**

- The amount of state information increases proportionally with the number of flows placing a huge overhead on storage and processing on the routers.
  - Routers also have to implement RSVP, admission control, packet classification, etc.
- Ubiquitous deployment of IntServ is required for guaranteed service.
- Incremental deployment of IntServ is difficult.

**INTSERV**

- IntServ is designed to be a common platform for both real-time and non-real-time traffic.
  - Example: Extending the features of common Internet protocols such as IP.
- IntServ is a per-flow based QoS framework with dynamic resource reservation using RSVP.

There are some of the limitations of IntServ we can quickly look at. The amount of the state information it increases proportionally to the number of flows placing a huge overhead on storage and processing on the routers, I have already told you this IntServ is a per flow based quality of service framework.

So now since it is a per flow based quality of service framework, so the amount of the state information it increases proportionally with a number of flows. So if the number of flows increases, then the amount of state information also increases proportionally, the amount of state information increases proportionally with the number of flows placing a huge, huge overhead and storage and processing on the router.

So, this will place a large amount of overhead, a huge overhead on what, on the storage capacity, as well as on the processing on the routers. Another difficulty is that the routers they also have to implement RSVP admission control policy packet classification, etcetera. So, obviously it required huge overhead on the storage and processing on the routers.

So ubiquitous deployment of IntServ is required for guaranteed service, so in order to achieve guaranteed service ubiquitous deployment of IntServ is required. Incremental deployment of IntServ is difficult, especially in IntServ the incremental deployment of IntServ is very much difficult.

(Refer Slide Time: 26:52)

**DIFFSERV**

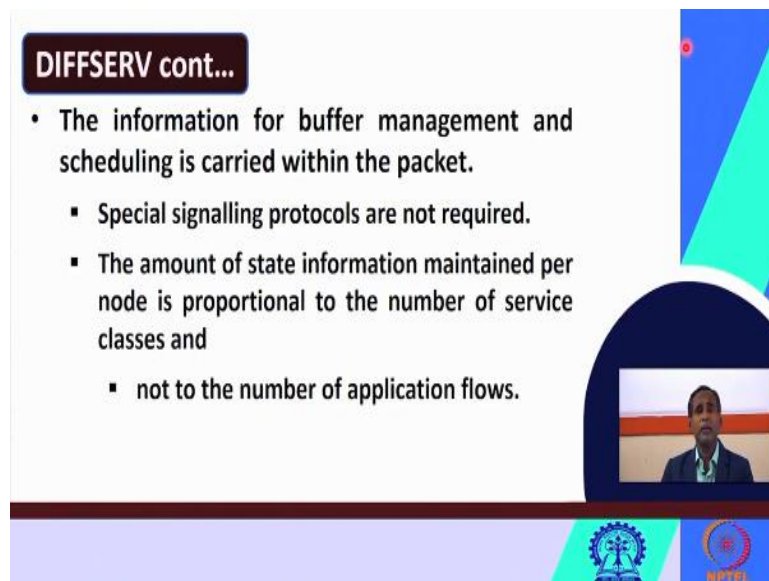
- DiffServ overcomes the limitations of IntServ.
  - It is simpler and is more scalable.
- It redefines the Type of Service (TOS) field in the header of IPv4 or IPv6 traffic class byte as a Differentiated Service (DS) field.
  - The first six bits are called DS Code Point (DSCP).

We will see the other model that is this DiffServ, so DiffServ overcomes some of the limitations of IntServ, we have already seen some of the limitations of IntServ, DiffServ overcome some of the limitations of IntServ. So it is simpler and is more scalable, so DiffServ is simpler than IntServ and it is more scalable, it can be easily scalable to a large size, so it redefines the type of the service field in the header of IP version 4 and IP version 6, I hope you have already known IP version 4 and IP version 6 traffic class.

So, in this IP version 4 or IP version 6 in the header, there is a field called as a TOS field, type of service. So these DiffServ redefines these TOS field, that means this type of service filed in the header of the IP version 4, IP version 6 traffic class byte as a differentiated service field. So this DiffServ what is does it redefine this TOS field as a differentiated service field or as DS field. The first six bit are called DS code point, so in this field the first the six bits are called DS code point or DSCP.



(Refer Slide Time: 28:02)



**DIFFSERV cont...**

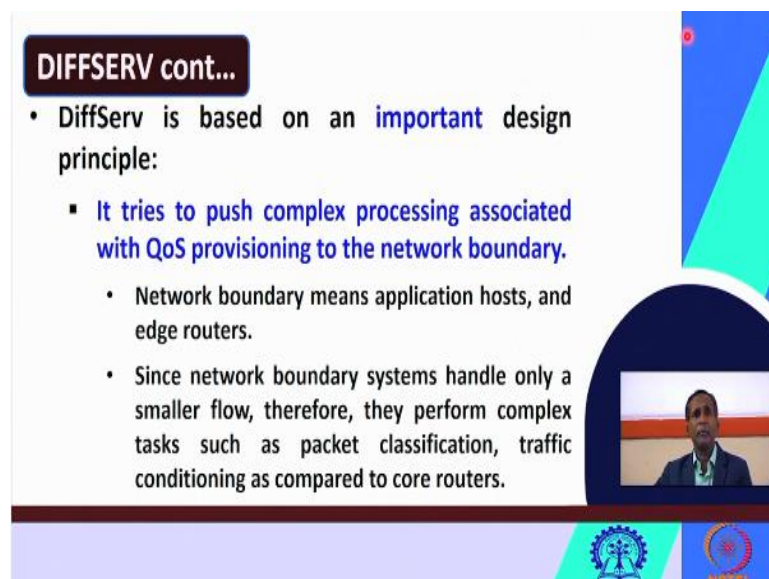
- The information for buffer management and scheduling is carried within the packet.
  - Special signalling protocols are not required.
  - The amount of state information maintained per node is proportional to the number of service classes and
    - not to the number of application flows.

The slide features a video inset of a man in a suit speaking, and logos for IIT Bombay and NPTEL at the bottom.

The information for management and scheduling is carried within the packet, that means in DiffServ the packet contains the information for buffer management and scheduling, no special signaling protocols are required, here the advantage is that the amount of state information maintained per node is proportional to the number of the service classes but see in IntServ what is happening, this was proportional to the number of application flows, hence the overhead was more.

But here, the amount of state information maintained per node is proportional to the number of service classes, not to the number of application flows as it was occurring in the IntServer cases. Hence, so that is why the overhead is less, very less compared to the IntServer model.

(Refer Slide Time: 28:49)



**DIFFSERV cont...**

- DiffServ is based on an **important** design principle:
  - It tries to push complex processing associated with QoS provisioning to the network boundary.
    - Network boundary means application hosts, and edge routers.
    - Since network boundary systems handle only a smaller flow, therefore, they perform complex tasks such as packet classification, traffic conditioning as compared to core routers.

The slide features a video inset of a man in a suit speaking, and logos for IIT Bombay and NPTEL at the bottom.

The DiffServer is based on an important design principle, so this principle says that it tries to push the complex processing associated with QoS provisioning into the network boundary. So in DiffServer model, or this model, it tries to push the complex processing logic associated with QoS provisioning, where, to the network boundary. And since you know that network boundary means what, network boundary means the application host and the age, routers etcetera.

Since the network boundary system it handles only a smaller flow hence they perform the complex tasks, I have already told you, this DiffServer model it pushes the complex processing over to the network boundary, since the network boundary systems, they handle only smaller flow, therefore they perform the complex task, they can perform the complex task such as packet classification, traffic conditioning as compared to the core routers. This is what DiffServer is based on this design principle.

(Refer Slide Time: 29:55)

**DIFFSERV cont...**

- DiffServ provides **two** service models:
  - **Premium service**
    - It is a guaranteed peak rate service which is optimized for regular traffic patterns and incurs almost no queuing delays.
    - Example: virtual leased line.
  - **Assured service**
    - It provides statistical guarantees to applications.

DiffServer provides two service models, premium service and assured service. So, premium service is a guaranteed peak rate service, which is optimized for regular traffic patterns and incurs almost no queuing delay. In premium service, you will not get any queuing delays, example is the virtual leased line you are using, that is an example of premium service. And in assured service, what does it do? It provides a statistical guarantee to applications, so assured service it provides statistical guarantees to applications.

(Refer Slide Time: 30:27)

**SOFT REAL-TIME COMMUNICATION IN A LAN**

- Characteristics of Soft Real-Time Communication Networks:
  - No absolute QoS guarantees are provided to the applications.
  - Only ensures prioritized treatment for real-time messages.
    - Helps to keep the message-deadline miss ratio to a minimum.
    - Provides statistical guarantees on delay bounds.

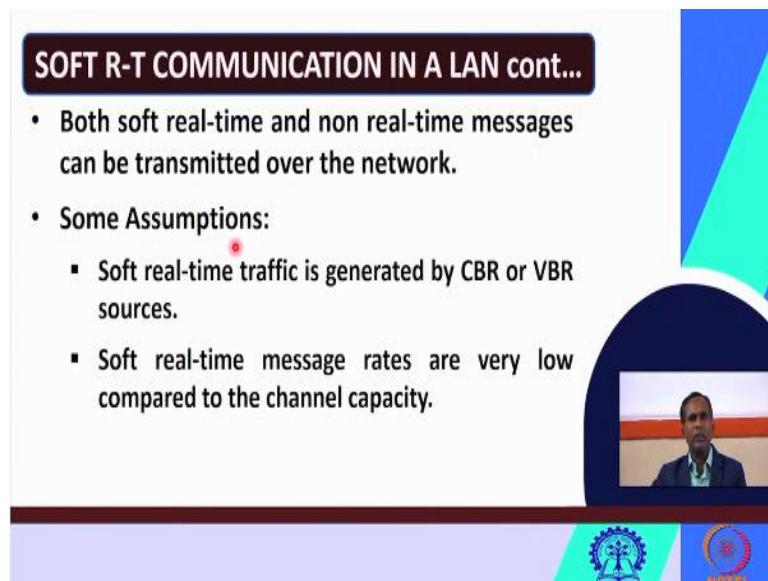
The slide features a dark blue header with the title in white. The main content is on a white background with blue and red text. A small video inset in the bottom right shows a man speaking. Logos for institutions are visible at the bottom.

So, in this we have seen a successful QoS models, two important models you have discussed that is IntServ and DiffServ. Now, let us see about soft real time communication in LAN, in my earlier classes I have already discussed about hard real time communication in LAN, different protocols we have seen, so now let us quickly look into the soft real time communication in LAN, how soft real time communication occurs in a LAN.

So, the characteristics of soft real time communication networks are follows, so let us see first some of the important characteristics of soft real time communication networks. Here are no absolute QoS guarantees are provided to the applications, no absolute quality of service guarantees can be provided to the applications, number one characteristic.

Then number two, it ensures only the prioritized treatment for real time messages, it only what does, it prioritizes the real time messages, it only ensures the prioritize treatment for the different real time messages. This will help keep the message deadline miss ratio to a bare minimum, so since it prioritizes the real time messages, it helps keep the message deadline miss ratio to a bare minimum. It also provides statistical guarantees and delay bounds.

(Refer Slide Time: 31:42)



**SOFT R-T COMMUNICATION IN A LAN cont...**

- Both soft real-time and non real-time messages can be transmitted over the network.
- Some Assumptions:
  - Soft real-time traffic is generated by CBR or VBR sources.
  - Soft real-time message rates are very low compared to the channel capacity.

The slide features a dark blue header with the title in white. The main content is on a white background with a blue and green geometric design on the right. A small video inset shows a man in a suit speaking. At the bottom, there are logos for IIT Bombay and IIT Madras.

So both the soft real time as well as the non-real time messages can be transmitted over the network using soft real time communication in a LAN. So some assumptions are those are taken in soft real time communication in LAN are as follows. It assumes that soft real time traffic is generated by CBR or VBR sources, I have already told you CBR and VBR these traffics, constant bitrate and the variable bitrate, so soft real time traffic is generated by either CBR or VBR sources.

Similarly, another assumption it makes is that soft real time message rates are very low compared to the channel capacity, compared to the channel capacity, the soft real time message rates are very low, based on these two assumptions, soft real time communication it works.

(Refer Slide Time: 32:25)

**SOFT R-T COMMUNICATION IN A LAN cont...**

- Both soft and non real-time messages may arrive periodically in bursts.
  - In presence of bursts, it is more difficult to sustain the guarantees to the soft real-time traffic.
  - Therefore, soft real-time communication protocols involve techniques to “smoothen” the effect of bursts.

The slide features a video inset of a man speaking in the bottom right corner. At the bottom, there are logos for IITM (Indian Institute of Technology Madras) and IITB (Indian Institute of Technology Bombay).

I have already told you, both soft and non-real time messages can be transmitted over the network using this soft real time communication, both soft and non-real time messages may arrive periodically in bursts, it is possible both soft and non-real time messages they may arrive periodical in burst.

In presence of burst it will be very much difficult to sustain the guarantees to the soft real time traffic, so when bursty nature is there in the presence of burst, it will be very much difficult to sustain the guarantees to the soft real time traffic, therefore the soft real time communication protocols they involve techniques to smoothen the effect of the burst. It is at all required to smooth the effect of the burst. Therefore, the soft real time communication protocols, they involve, include some techniques to smoothen the effects of the burst.

(Refer Slide Time: 33:15)

**FIXED RATE TRAFFIC SMOOTHING ALGORITHM**

- It was developed by Kweon and Shin.
- It takes into account limits of the transmission capacity of the network, and then derives the input limit for each node in the system.
- The “traffic smoother” is placed between the MAC and the TCP/IP layer.
- It uses a leaky bucket algorithm called Credit Bucket Depth (CBD).

**SOFT R-T COMMUNICATION IN A LAN cont...**

- Both soft and non real-time messages may arrive periodically in bursts.
  - In presence of bursts, it is more difficult to sustain the guarantees to the soft real-time traffic.
  - Therefore, soft real-time communication protocols involve techniques to “smoothen” the effect of bursts.

So we will see two important algorithms for the soft real time communication in a LAN, first we will see fixed rate traffic smoothing algorithm, we have to smooth the algorithm because I have already told you, therefore the soft real time communication protocols involved techniques for what, to smoothen the effect of burst, how to do?

One algorithm we will see here, this is the fixed rate traffic smoothing algorithm, this algorithm was proposed by Kweon and Shin. This algorithm takes into account the limits of the transmission capacity of the network, so this algorithm it considers the limits of the transmission capacity of the network, and then what does it do, it derived the input limit for each node in the system, what will be the input limit for each node in the system it derives.



As its name is traffic smoothing, so there is traffic smoother component, the traffic smoother component is placed between the MAC layer and the TCP IP layer. So this algorithm fixed rate smoothing algorithm it uses a leaky bucket algorithm, you have already known leaky bucket algorithm in computer networks paper, also in my last class I was discussing little bit related to this algorithm. So, this fixed rate traffic smoothing algorithm, it uses a leaky bucket algorithm called CBD or credit bucket depth.

(Refer Slide Time: 34:33)

**FIXED RATE TRAFFIC SMOOTHING ALGO cont...**

- The algorithm has two important parameters: **CBD and Refresh Period (RP)**.
  - **CBD indicates**
    - The maximum number of credits that are added to the bucket at every refresh,
    - The maximum number of credits that a bucket can hold.
  - **RP stands for the refresh period with which the bucket is replenished with new credits.**

**FIXED RATE TRAFFIC SMOOTHING ALGORITHM**

- It was developed by Kweon and Shin.
- It takes into account limits of the transmission capacity of the network, and then derives the input limit for each node in the system.
- The “traffic smoother” is placed between the MAC and the TCP/IP layer.
- It uses a leaky bucket algorithm called **Credit Bucket Depth (CBD)**.

So, this fixed rate traffic smoothing algorithm, it has two important parameters, one is the CBD, that is this credit bucket depth and the RP, RP stands for the refresh period. So CBD indicates what, CBD indicates the maximum number of credits, which are added to the bucket at every

refresh, so after every refresh what is the maximum number of credits that can be added, that is indicated by CBD.

It also indicates the maximum number of credits that a bucket can hold. So CBD indicates two important things, it indicates the maximum number of credits that are added to the bucket at every refresh and it also indicates the maximum number of credits that a bucket can hold.

Now, this RP I have already told you RP stands for refresh period with which the bucket is replenished to the new credit, new credit, so what do you mean by RP, so RP is the refresh period with which the bucket it is replenished, it is replenished with the new credit, so this is known as what refresh period.

(Refer Slide Time: 35:40)

**FIXED RATE TRAFFIC SMOOTHING ALGO cont...**

- The ratio  $CBD/ RP$  stands for the average guaranteed throughput for non RT messages.
- Current Network Share (CNS) stands for credits present in the bucket at any time.
- A waiting message is transmitted if its size is less than CNS.
- Otherwise, if  $CNS > 0$  but is less than the size of the message, credits are allowed to be borrowed.
  - At every refresh,  $CNS = \min(CNS + CBD, CBD)$

Slide 1 includes a video inset of a man in a suit and logos for IIT Bombay and NPTEL.

**FIXED RATE TRAFFIC SMOOTHING ALGO cont...**

- The algorithm has two important parameters: **CBD and Refresh Period (RP)**.
  - CBD indicates
    - The maximum number of credits that are added to the bucket at every refresh,
    - The maximum number of credits that a bucket can hold.
  - RP stands for the refresh period with which the bucket is replenished with new credits.

Slide 2 includes a video inset of a man in a suit and logos for IIT Bombay and NPTEL.

So then, if I can compute the same, because I know what is CBD, I know what is the RP, if I can compute the ratio CBD by RP, what does it represent, what does it stand? The ratio CBD by RP stands for or it represents, the average guaranteed throughput for the non-real time messages.

So for the non-real time messages, but in the average guaranteed throughput, this can be represented by the ratio CBD by RP. The current network share, there is another parameter we use in fixed rate traffic smoothing that is CNS called as current networks share, which stands for credits present in the bucket at any time.

So in the bucket, how many credits are present at any particular point of time, that is represented by a parameter called as CNS which stands for current networks share. Now a waiting message is transmitted, if its size is less than CNS, so if a message is waiting in the queue, and it when it can be transmitted? It can be transmitted only when the size of the message is less than the current networks share, otherwise if size is not less, then it cannot be transmitted.

However, if the value of CNS is greater than 0, but less than the size of the message, then what is possible, the credits can be allowed to be borrowed, credits are allowed to be borrowed, so that the message can be transmitted. So there are two possibilities you see that if size of the message is less than CNS, then the waiting message can be transmitted no problem at all.

If it is less than, if the value of CNS is less than 0, the message cannot be transmitted at all. If the value of the CNS is greater than 0, but less than the size of the message, then what is possible, credits are allowed to be borrowed, so that the message can be transmitted.

Now let us compute the value of CNS, at every refresh the value of CNS, should be the minimum value of two parameters, what is, whichever is the minimum CNS plus CBD, and the only CBD, which value is minimum, that value is assigned to CNS, so in this way at every refresh the value of CNS can be computed.

(Refer Slide Time: 37:54)

**FIXED RATE TRAFFIC SMOOTHING ALGO cont...**

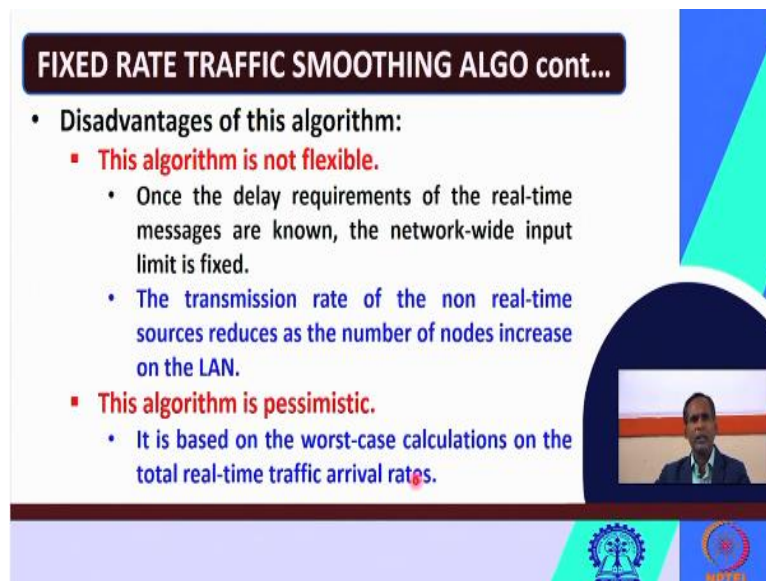
- For non real-time messages, the following steps are followed for smoothing:
  1. `if(CNS>0){`
  2. `CNS=CNS-message.no_of_bytes;//subtract size of the message`
  3. `send message for transmission;`
  4. `}`
  5. `else`
  6. `{`
  7. `hold message in buffer until CNS > 0;`
  8. `}`

Now, for non-real time messages which steps we will follow for smoothing? So for non-real time messages, the following steps can be followed for smoothing. First you check the value of CNS, if CNS is greater than 0, then you update CNS as CNS minus the message dot number of bytes. So what is the message dot number of bytes, message dot number of bytes it represents the size of the message.

So if the CNS is greater than 0, update CNS as CNS minus the size of the message, then you can transmit the message, or send the message for transmission, else if CNS is less than 0, if CNS is greater than 0, then update CNS and send the message for transmission, if CNS is less than 0, then you have to hold the message in buffer, until it is greater than 0 because if the value of CNS is negative, it cannot be, message cannot be transmitted, you have to keep it in hold until the value of CNS is greater than 0.

So if the value of CNS is less than 0, hold the messages in the buffer until the value of CNS it is greater than 0, this is how the non-real time messages can be treated for achieving smoothing, for non-real time messages, the following steps should be followed for smoothing.

(Refer Slide Time: 39:13)



**FIXED RATE TRAFFIC SMOOTHING ALGO cont...**

- Disadvantages of this algorithm:
  - **This algorithm is not flexible.**
    - Once the delay requirements of the real-time messages are known, the network-wide input limit is fixed.
    - The transmission rate of the non real-time sources reduces as the number of nodes increase on the LAN.
  - **This algorithm is pessimistic.**
    - It is based on the worst-case calculations on the total real-time traffic arrival rates.

The slide features a blue and green geometric design on the right side, a small video inset of a speaker, and logos for institutions at the bottom.

Two major disadvantages of this algorithm. This algorithm is not flexible at all and this algorithm is not optimistic, this algorithm is pessimistic. So how it is not flexible, once the delay requirements of the real time messages are known the network wide input limit is becoming fixed. The transmission rate of the non-real time source it reduces as the number of nodes increase on the LAN.

So when the number of nodes increase on the LAN, then the transmission rate of the non-real time sources is reduced. So this algorithm is not flexible. By, it is not optimistic, why it is pessimistic? Because it is based on the worst case calculations on the total real time traffic arrival rates, that is why, so since this algorithm based on the worst case calculation on the total real time traffic arrival rates. So this is not optimistic, this is pessimistic.



(Refer Slide Time: 40:03)

**ADAPTIVE TRAFFIC SCHEDULING**

- It was proposed by Kweon and Shin to address the shortcomings of the CBD algorithm.
- This technique essentially tries to achieve a better n/w utilization for non real-time traffic.
  - The non real-time transmission rate is adapted according to the VBR real-time traffic.
- Two issues that need to be handled are:
  - How to detect changes in network utilization?
  - How to adapt the transmission limits to a detected change in network utilization?

**FIXED RATE TRAFFIC SMOOTHING ALGO cont...**

- Disadvantages of this algorithm:
  - **This algorithm is not flexible.**
    - Once the delay requirements of the real-time messages are known, the network-wide input limit is fixed.
    - The transmission rate of the non real-time sources reduces as the number of nodes increase on the LAN.
  - **This algorithm is pessimistic.**
    - It is based on the worst-case calculations on the total real-time traffic arrival rates.

So, now we will quickly look at another algorithm for soft real time communication in LAN, that is known as adaptive traffic scheduling. This algorithm was proposed by Kweon and Shin to address some of the shortcomings of the CBD algorithm, I have already told you, some of the shortcomings of this what this previous algorithm or the CBD based algorithm. This algorithm was proposed to address some of the shortcomings of this CBD based, the previous CBD based algorithm.

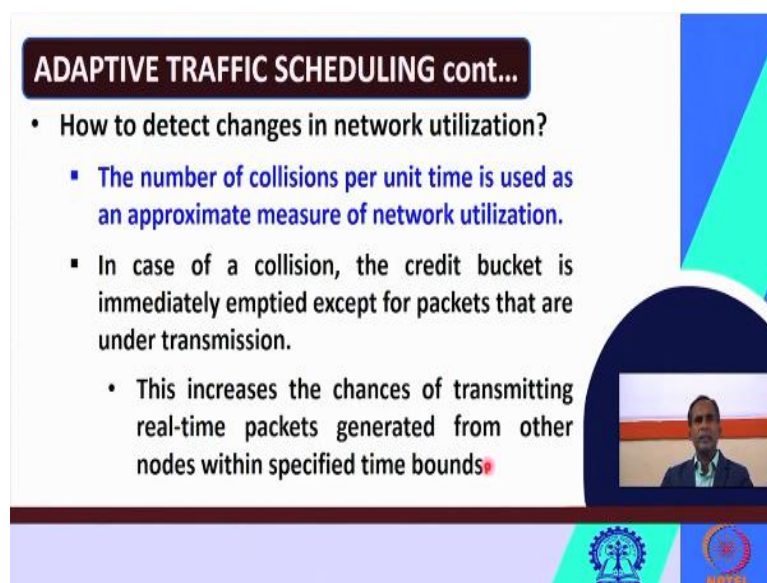
So this technique it essentially tries to achieve a better network utilization for non-real time traffic. So, this adaptive traffic scheduling, it tries to achieve a better improved network utilization value, so this technique essentially tries to achieve a better network utilization value for the non-real time traffic here, the non-real time traffic or here the non-real time transmission



rate is adapted according to the VBR real time traffic, we have already discussed what is VBR real time traffic variable bitrate real time traffic.

In this technique, the non-real time transmission rate is adapted according to what, it is adapted according to the (()) (41:14) ... adaptive traffic scheduling. Here two important issues that need to be handled, are as follows, how to detect the changes in network utilization and how you can detect the changes occurring in network utilization. Second issue is, how to adapt the transmission limits to a detected change in network utilization, how you can adapt the transmission limits to a detected change which has occurred in the network utilization.

(Refer Slide Time: 41:48)



**ADAPTIVE TRAFFIC SCHEDULING cont...**

- How to detect changes in network utilization?
  - The number of collisions per unit time is used as an approximate measure of network utilization.
  - In case of a collision, the credit bucket is immediately emptied except for packets that are under transmission.
    - This increases the chances of transmitting real-time packets generated from other nodes within specified time bounds.

The slide features a blue and green geometric design on the right side, a small video inset of a speaker, and logos for IIT Bombay and NPTEL at the bottom.

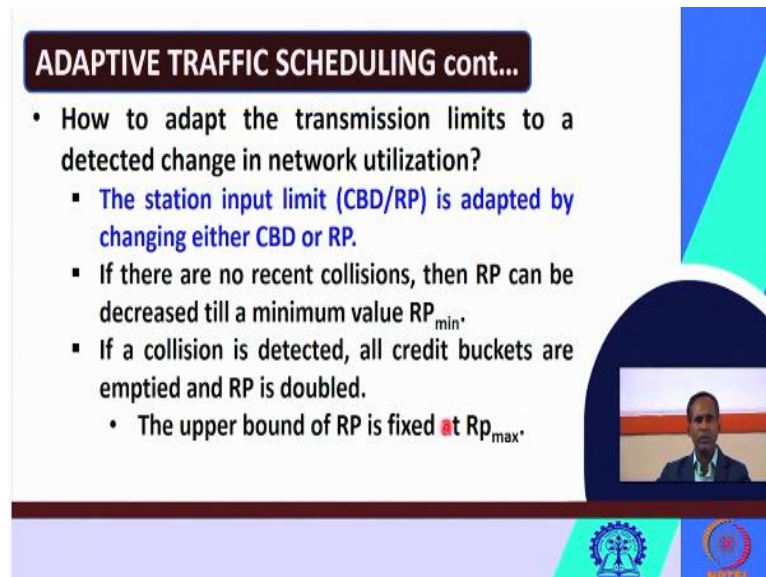
Let us see the first issue, how to detect changes in network utilization? So, you know, the number of collisions per unit time is used as an approximate measure of network utilization, so in this scheme, the number of collisions occurred per unit time maybe per second, or what is used as an approximate measure network utilization.

So when a collision occurs in case of a collision, the credit bucket is immediately emptied, except for packets that are under transmission, so we have already seen this leaky bucket kind of thing earlier. So when a collision occurs, the credit bucket is immediately emptied, it is made empty the credit bucket is immediately made empty except for the packets which are under transmission.

This will increase the chances for transmitting the real time packets generated from other nodes within specified time bounds. So since the credit bucket is immediately made empty, excepting

for the packets which are under transmission, this will increase the chances of transmitting the real time packets which are generated from other nodes within some specified time bounds.

(Refer Slide Time: 43:00)



**ADAPTIVE TRAFFIC SCHEDULING cont...**

- How to adapt the transmission limits to a detected change in network utilization?
  - The station input limit (CBD/RP) is adapted by changing either CBD or RP.
  - If there are no recent collisions, then RP can be decreased till a minimum value  $RP_{min}$ .
  - If a collision is detected, all credit buckets are emptied and RP is doubled.
    - The upper bound of RP is fixed at  $RP_{max}$ .

The slide features a video inset of a man in a suit on the right side. At the bottom, there are logos for IIT Bombay and NPTEL.

Then let us see quickly the second one, how to adapt the second issue, how to adapt the transmission limits to a detected change in the network utilization. The station input limit that is CBD by RP it is adapted by changing either CBD or RP, how this station input limit is adapted? The station input limit which is given by CBD by RP is adapted by changing either the value of CBD or by changing the value of RP.

If there are no recent collisions, then the value of RP can be decreased to a minimum value, maybe RP minimum, so if there are no recent collisions now, the value of RP, it can be decreased, it can be reduced to till getting the minimum value, which may be represent as RP min.

On the other hand, if a collision is detected, then all credit buckets, they are emptied, all the credit buckets are made empty, and this value of RP is doubled. The upper bound of this RP value is fixed, and we may call as RP max, the upper bound of this value RP is fixed and it is fixed at what value, it is fixed at the value RP max, we can represent this value as RP max, okay so this is how this adaptive traffic scheduling works.

(Refer Slide Time: 44:25)

## CONCLUSION

- Discussed some non-work conserving disciplines.
- Presented basics of QoS Models.
- Discussed about Integrated and Differentiated Services.
- Explained soft real-time communication in a LAN.



## FIXED RATE TRAFFIC SMOOTHING ALGO cont...

- Disadvantages of this algorithm:
  - **This algorithm is not flexible.**
    - Once the delay requirements of the real-time messages are known, the network-wide input limit is fixed.
    - The transmission rate of the non real-time sources reduces as the number of nodes increase on the LAN.
  - **This algorithm is pessimistic.**
    - It is based on the worst-case calculations on the total real-time traffic arrival rates.



## ADAPTIVE TRAFFIC SCHEDULING cont...

- How to adapt the transmission limits to a detected change in network utilization?
  - The station input limit (CBD/RP) is adapted by changing either CBD or RP.
  - If there are no recent collisions, then RP can be decreased till a minimum value  $RP_{min}$ .
  - If a collision is detected, all credit buckets are emptied and RP is doubled.
    - The upper bound of RP is fixed at  $RP_{max}$ .



So today I have discussed some non-work conserving disciplines, we have discussed two important quality of service models like integrated services and differentiated services. We have also explained how soft real time communications occurs in LAN. Particularly, we have seen two algorithms here. One is the fixed rate traffic smoothing algorithm, another is the adaptive traffic scheduling algorithms, these two algorithms are very much popular, which are used for achieving soft real time communication in LAN.

(Refer Slide Time: 45:02)



**REFERENCES**

1. Rajib Mall, Real-Time Systems: Theory and Practice, 1st Edition, 2007, Pearson Education
2. C. M. Krishna & K. G. Shin, Real-Time Systems, 2017, Tata McGraw Hill Education

The slide includes a video inset of a man speaking in the bottom right corner and logos of institutions at the bottom.

We have taken on the content from these books. Thank you very much.