Broadband Networks

Prof. A. Karandikar

Electrical Engineering Department

Indian Institute Of Technology, Bombay

Lecture - 15

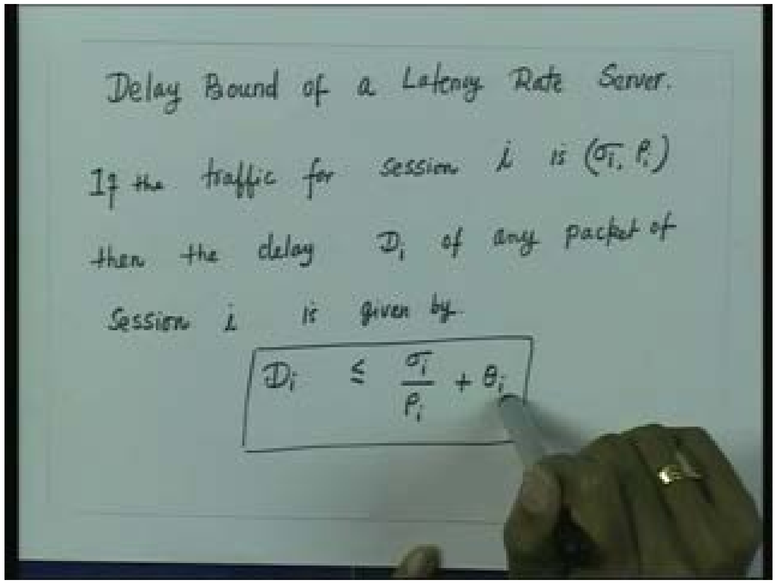
Latency Rate Serveres – II and Delay Bounds

So, we will now discuss about the delay bound of latency rate server, the delay bound that can be obtained in a latency rate server. So, as we have already defined latency rate server guarantees that the amount of traffic that can be served, that a amount of traffic that can be served during a sub interval of the session's busy period and out of all the traffic that arrived during the busy period, the session's busy period will always be greater than or equal to maximum of either 0 or rho times the duration of the interval minus the latency of the latency rate servers. So, that is so that is how a latency rate server is defined.

Intuitively, it means that sessions will have to wait for an amount of time equal to the latency of the latency of the server for its traffic to get started by the scheduler to get served by the scheduler. That is how the intuitive definition of the latency rate server is. Most of the packet scheduling algorithm that we have discussed till now that is the generalized processor sharing which is actually a fluid algorithm or the waited fair queuing or the self clocked fair queuing virtual clock or the rate proportional servers; most of these packet scheduling algorithms, they fall under the categories of latency rate servers. So, it can be actually shown that all these packet scheduling algorithms are actually latency rate scheduling algorithms with different latencies.

So, now what we will try to prove is that if the traffic if the traffic to these packet scheduling algorithms is rho sigma regulated, then what is the delay bound that can be achieved by latency rate server. So, we have this result which states that if the traffic for the session i is (sigma $_i$ rho $_i$) regulated, then the delay D $_i$ of any packet of session i is given by D $_i$ will be less than or equal to sigma $_i$ upon rho $_i$ plus theta $_i$.

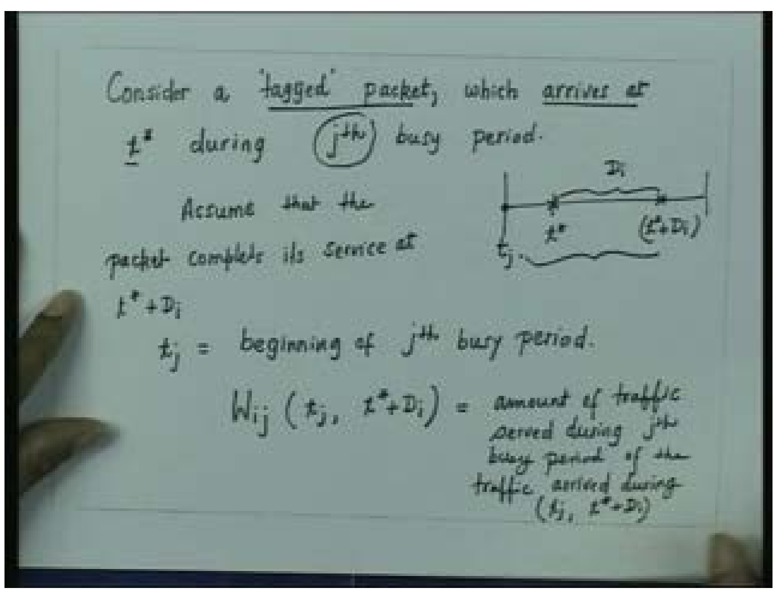
(Refer Slide Time: 4:09)



So, one assumption that we have made here in this latency rate scheduler is the allocated rate by the packet scheduler is sigma $_i$ which is also the long term average rate or the token rate of the rho sigma regulated. So, the same rate has been allocated by the packet scheduler also and what we would like to show that a packet scheduler which is a latency rate server can guarantee a delay bound if the traffic to this scheduler is a rho sigma regulated.

So, we will try to prove this result. Let us say that we consider for example some a tagged packet.

(Refer Slide Time: 5:28)

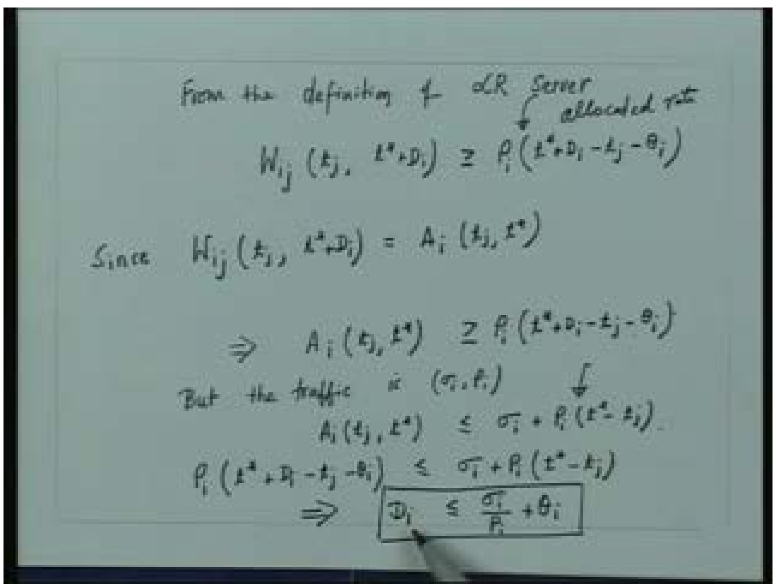


So, let us consider a tagged packet. By tagged packet I mean that a packet, a particular packet has arrived and that packet has been tagged. Then let us say that this packet arrives at some time say t star. So, let us say that this is a tagged packet which arrives at which arrives at point t star during j'th busy period. Now, assume that this particular packet, the particular packet, this packet arrives at time t star. So, something like this that this is the j'th busy period, the packet actually arrives here at t star and let us assume that the packet completes its service at t star plus D_i.

So, the packet here t star plus D_i, the packet complete it service. So, this D_i denotes the delay of the packet and let us say tau here, it denotes or let us say t_j denotes the beginning of the j'th period. So, let us say that t_j denotes the beginning of the j'th busy period. Now that means W_{ij} (t_j, t star plus D_i) what does it denote?

It denotes that the amount of traffic amount of traffic served during j'th busy period of all the traffic amount of traffic served during j'th busy period out of all traffic that arrived during this interval that is tj to t star plus D_i of the traffic arrived during this interval.

(Refer Slide Time: 9:18)



Now, from the definition of the latency rate servers, from the definition of latency rate server, we know that W_{ij} (t_j, t star plus D_i) that is the amount of traffic served during this period of the j'th busy period is greater than or equal to rho $_i$ into (t star plus D_i minus t_j minus theta $_i$) where theta i is the latency.

We also know that W_{ij} (t_j, t star plus D_i) is also equal to A_i into (t_j t star). This is equal to the amount of traffic that arrived during t_j to t star because all that traffic has been served by t star plus D_i. At t star, the packet which arrived at t star has been served by t star plus D_i. So, this means that so since this is equal to this, A_i (t_j into t star) is greater than - if you substitute this A_i here - is greater than rho $_i$ (t star plus D_i minus t_j minus theta $_i$).

So, what does it say is that the amount of traffic that arrived from the beginning of the j'th period the amount of traffic that arrived from the beginning of the j'th period to the time when this tacked packet has arrived; so, all the traffic that has arrived during this period can be shown to be greater than or equal to rho $_i$ into this duration (t star plus D $_i$ minus t $_j$ minus theta $_i$). But our traffic is rho sigma regulated. Since the traffic is a rho sigma traffic, sigma $_i$ rho $_i$ traffic; we know that A $_i$ t $_j$ into t star this itself will be less than or equal to sigma $_i$ plus rho $_i$ into t star minus t $_j$.

So therefore, from these 2 equations we can see that rho $_i$ into t star plus D $_i$ minus t $_j$ minus theta i will be less than or equal to sigma $_i$ plus rho $_i$ into t star minus t $_j$. One point to note here is when I write from the definition of latency rate server, the amount of traffic that has been served during the j'th busy period, during this interval, when i say that this is greater than or equal to rho $_i$ into this time interval, here the rho $_i$ is the allocated rate by this latency rate server. So, this denotes the allocated rate.

Now, when I am writing here that a traffic is sigma $_i$ rho $_i$ regulated, I am also writing the same rho $_i$. So, I am assuming that the long term average rate of the traffic which is indicated by rho $_i$ is equal to the allocated rate and that is why I am using the same rho $_i$. So, by looking at these 2 equations, I can see that rho $_i$ into (t star plus D $_i$ minus t $_j$ minus theta $_i$) is less than or equal to sigma $_i$ plus rho $_i$ into (t star minus t $_j$) and from this I can see that the D $_i$, the delay of any packet is less than or equal to sigma $_i$ plus rho $_i$ plus theta $_i$, just from this algebraic manipulations.

So, this shows that the delay of a packet is bounded by sigma $_i$ upon the rho $_i$ plus the latency of a particular packet scheduling algorithm. Now, this is a very important aspect of the packet scheduling algorithm that we have proved. If you recall in our previous lectures, when we started by the design of a packet scheduling algorithm; we had argued that we wanted to schedule the packets of different flows in a router such that every flow gets a fair share of the output link bandwidth and we argued that first come first serve scheduling algorithm cannot differentiate between different flows and therefore it cannot give fairness to the different flows.

So therefore, it is necessary that we must have some kind of non first come first serve scheduling algorithms which can guarantee fairness to the different flows. Now, this obviously raises the question that what is meant by fairness. Now, we said that our definition of fairness will be max min fairness. In max min fairness, no flow gets a share which is larger than its requirement and all the flows whose requirements cannot be satisfied, all those flows are given equal shares or shares in proportion to their weights.

Now, in order to achieve this maxmin fairness, we saw that if you assume the traffic to be fluid in nature, then if we implement a scheduler which serves an infinitesimal amount of fluid from each of the non empty queues, then this scheduler effectively achieves max min fairness. However, the trouble is that since the traffic is fluid in nature, the traffic is assumed to be fluid in nature and in practice the traffic will not be fluid in nature but it will be packetized in nature; so, it is necessary that we must have packetized versions of the scheduling algorithms.

We had discussed various packetized versions of the scheduling algorithms like weighted fair queuing and self clock fair queuing. All these scheduling algorithms were giving some kind of fairness. In fact, in the self clock fair queuing we saw that we could guarantee that the normalized service that is the services received by 2 sessions and normalized with respect to their weights or with respect to their allocated rates is always bounded. So, that is how the self clock fair queuing algorithm.

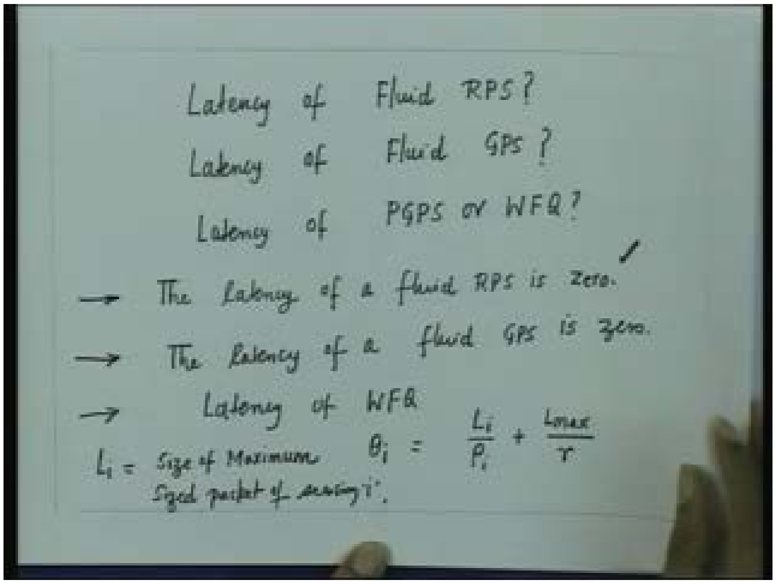
Now, this is about all giving fairness. Fairness is an important notion when we are serving best effort flows or best effort traffics. Now, we wanted to ask this question that suppose we want to give quality of service guarantees, if you want to give QOS to the different traffic flows; then whether we can achieve that different quality of service guarantees in our packet scheduling algorithm set up or not?

Now, just now we have proved that if the traffic is rho sigma regulated if the traffic is rho sigma regulated, then we can give delay guarantees we can give delay guarantees. In fact, we have shown that the worst case delays of a packet in a packet scheduling algorithm will be bounded by sigma $_i$ upon rho $_i$ plus the latency of the scheduler if the traffic, this input traffic to the session happens to be sigma rho regulated.

So, what does it mean is that if you want to give delay guarantees to various flows in a packet scheduling algorithm, then we must make sure that the traffic is rho sigma regulated. If we can have the traffic to be rho sigma regulated and allocate the same rho by the packet scheduling algorithm as its allocated rate, then we can also achieve delay guarantees.

Now, in order to find how much delay guarantees we can achieve, it all turns out that we now need to find out what are the latencies of the various packet scheduling algorithm. So, we need to find out what are the latencies of the different packet scheduling algorithms. Now, towards this end we would like to find out for example; what is the latency of a fluid rate proportional server a fluid RPS.

(Refer Slide Time: 20:12)

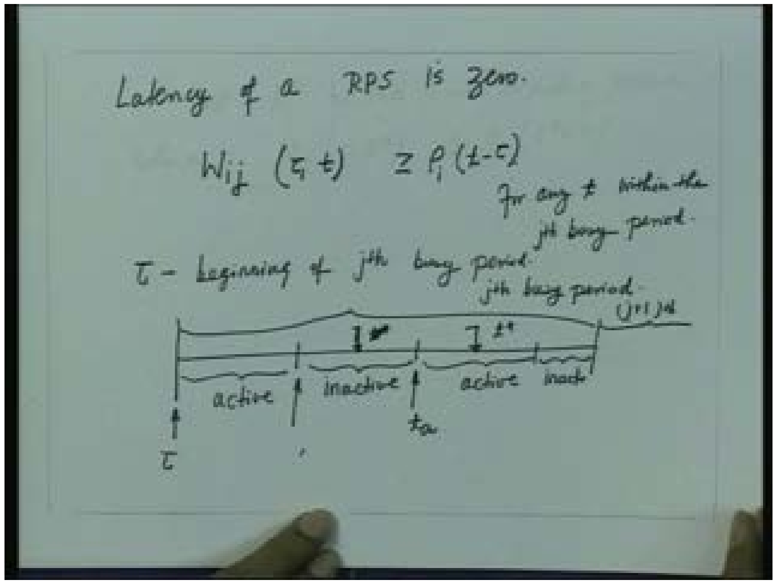


We would also like to find out what is the latency of a fluid GPS like fluid GPS server, we would also like to find out what is the latency of the packetized GPS that is PGPS or weighted fair queuing that is WFQ.

So, first result we would like to prove is that the latency of a fluid RPS is actually 0. Second result is that the latency of a fluid GPS scheduler is also 0. So, they are actually 0 latency rate servers. The latency of a WFQ scheduling algorithm is given by L $_i$ upon rho $_i$ L $_i$ upon rho $_i$ plus L $_{max}$ upon r, where L $_i$ happens to be the size of maximum sized packet, maximum sized packet of session i and L $_{max}$ happens to be the maximum sized packet in the particular scheduler.

First, let us try to prove that the latency of a fluid RPS is 0. So, we will try to prove first - this result.

(Refer Slide Time: 22:58)

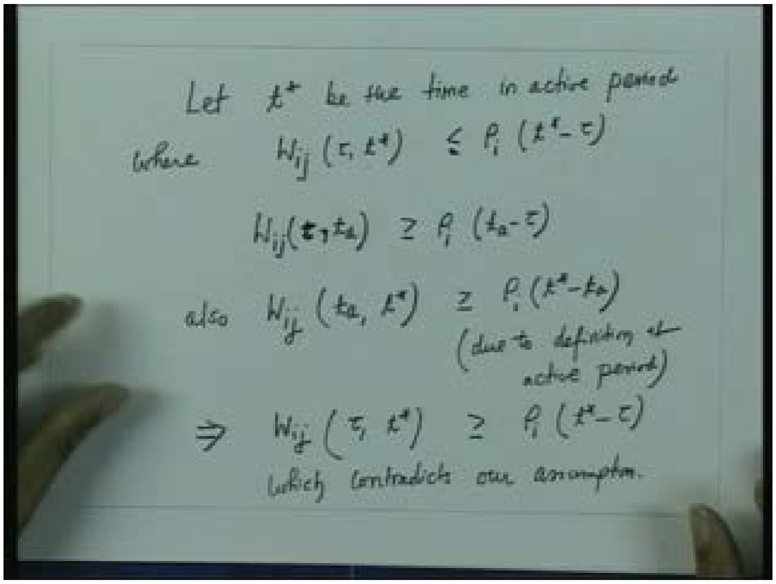


So, let us prove that the latency of a RPS, rate proportional server is 0. So, when we say that the latency of an RPS is 0, then we need to prove that during the j'th busy period of the session i; the amount of traffic which is served of all the traffic that arrived during this j'th busy period should be greater than or equal to rho $_i$ into t minus tau for any t within the j'th busy period. This is what we need to prove if the latency of RPS is 0.

If the latency was theta, then this result would have meant that the amount of traffic which is served during the j'th busy period is greater than or equal to rho $_i$ into t minus tau minus theta, whatever you know is the latency.

Now, we have assumed here that tau is the beginning of j'th busy period. So, something like this - tau is the so this is the j'th let us say that this is the j'th busy period, tau is the beginning of j'th busy period which will have this active and inactive sessions and here we have the j plus 1'th session begins. So, from here onwards j plus 1'th busy period determines. Now, let us say we will try to prove this result by contradiction.

(Refer Slide Time: 25:51)



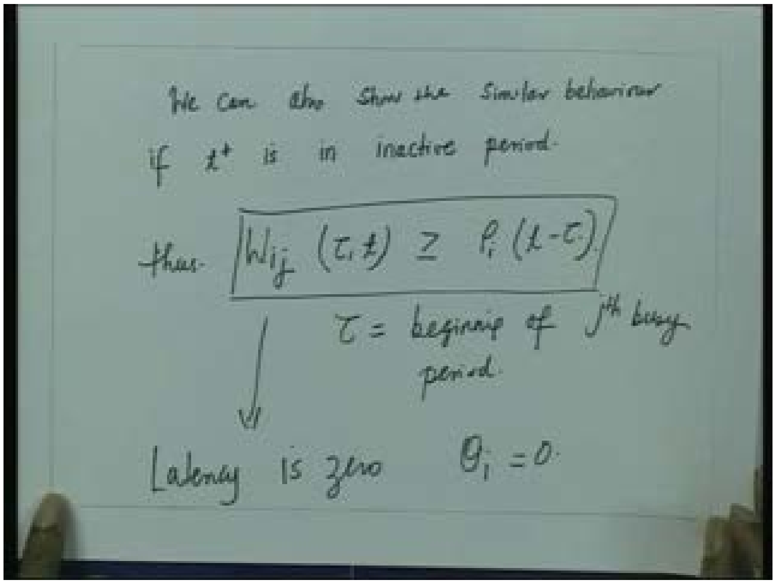
By contradiction means we will assume that this result fails for some d. That means we will assume that let t star be the time in the active period and let us say that this t star lies in the active period where this is happens for the first time that $W_{i j}$ (tau to t star) is less than rho $_i$ into (tau star minus tau). So, what does it happen is that that the t star is the time at which it happens that the amount of traffic that is served of all the traffic that arrived during this interval becomes less than rho $_i$ into (tau star minus tau) and let us say that sorry we have to assume this t star into the active period. So, i will assume this to be t star to be in the active period.

So, let us say that t_a is the beginning of that active period. Now, since tau is greater than or equal to tau, we also know that W_i into t minus t_a sorry W_i into W_i of tau into t_a, the amount of traffic that is served from tau to t_a that is greater than or equal to rho $_i$ into t_a minus tau. This is by assumptions. This becomes less than rho $_i$ into (tau star minus t) for the first time at t star. However, between tau and t_a; these results holds good.

Also, if you see then $W_{i j}$ (t_a into t star) - this has to be greater than or equal to rho $_i$ into (t star minus t_a) Why this result is applicable? If you see, this result is applicable because t_a into t star; they lie in the active period, the t star is assumed to be in the active period. So, this is true due to the definition of active period. This result\s holds true for the active period. So, which means if you combine these 2, then $W_{i j}$ (tau to t star) has to be greater than or equal to rho $_i$ into (t star minus tau). But which contradicts our assumption. So therefore, no, this cannot be true.

Similarly, we can also prove the result by assuming this t star to be in the active period also and by again considering that case where t star in the active period, we can prove the result.

(Refer Slide Time: 29:48)

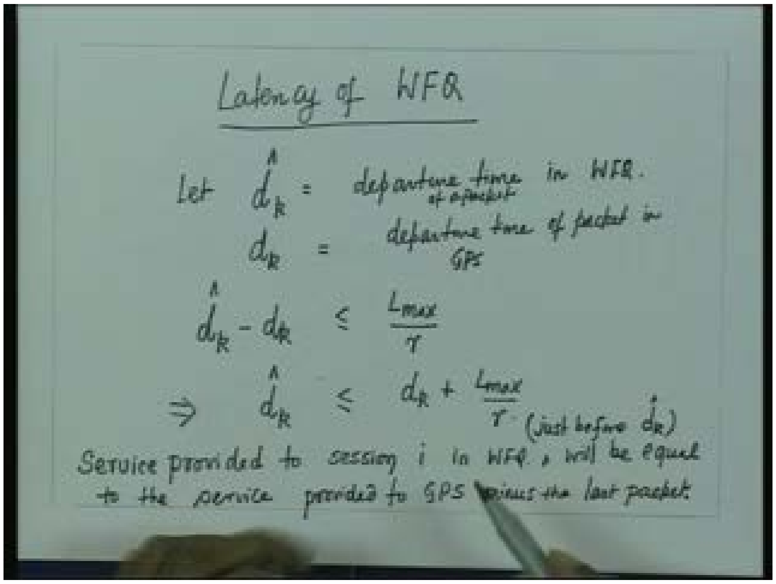


We can also similarly show we can also show the similar behavior if t star is in inactive period. So, we can show that and thus we can prove that the W $_{i\,j}$ (tau to t); so thus we have proved that this is greater than or equal to rho $_i$ into (t minus tau) where a tau is the beginning of some j'th busy period and t lies within this busy period. And, if this is so, then we also prove that the latency is 0. Essentially, theta $_i$ is 0.

So, what we have proved is that the latency of a fluid rate proportional server of fluid RPS is 0. Along the similar lines, we can prove that the latency of a fluid GPS will always be 0. This is because this result - the amount of traffic served during a subinterval tau to t is greater than or equal to rho $_i$ - this result holds true in the case of GPS for every backlogged period. So, therefore, this result also holds true for GPS and we can therefore argue that the latency of a GPS is also 0.

Now, let us try to prove the latency of a weighted fair queuing or WFQ.

(Refer Slide Time: 31:57)



So, we will now try to find out what is the latency of a weighted fair queuing algorithm. So, we now, we proved this result that let d hat k; d - the departure time in WFQ of a packet and d_k is departure time of the same packet. So, this is the departure time of a packet in WFQ and this d_k is the departure time of a packet in the GPS or the fluid flow fair queuing. Then, we had seen that d hat k minus d_k was less than or equal to L_{max} by r or in other words that d hat k was less than or equal to d_k plus L_{max} by r.

Now, one thing we have then see that suppose we are serving a particular packet in a weighted fair queuing and same packet is being served in the GPS; so, the service now in the weighted fair queuing we are assuming that a packet is served only when the last bit of the packet has been served and that is the time which is denoted by d hat k.

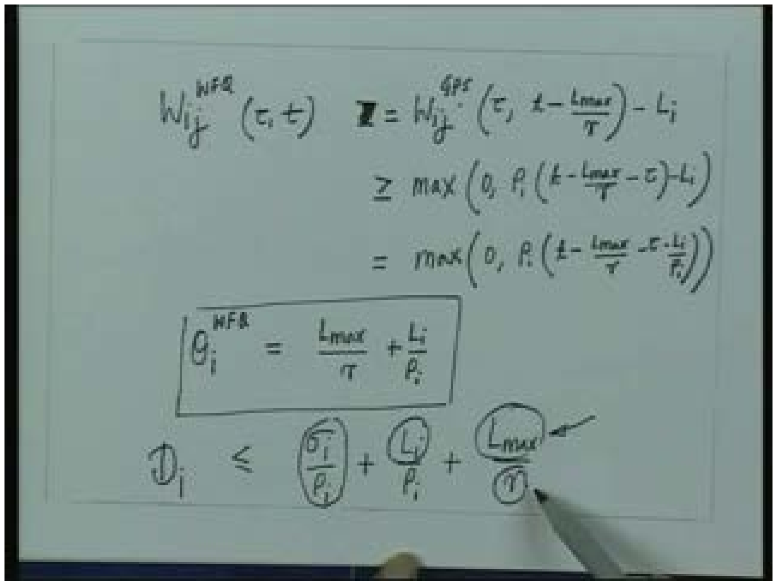
At this time, the packet would be considered as served that means the last bit of the packet has been served and then only the quantum of the service offered to the particular session will increase, that is W_i. Our W_i is also a stair case function, remember and our arrival A_i is also a stair case function. So, this function is increasing as the traffic is being served in the fluid scheduling algorithms, in the fluid GPS. But this traffic increases in the stair case manner in the packetized versions because in the packetized versions we consider a session to be served only when the last bit of a particular packet has been served.

So therefore, the service, in the service provided to session i in WFQ; so therefore, we can argue that the service provided to a particular session i in WFQ just before the time this d hat k just before this time d hat k will be equal to the service provided to the GPS minus this last packet.

So, the service provided to session i in WFQ just till just before the time d hat k, just before that, it will be equal to the service provided to GPS minus this last packet. Again, I want to emphasize here that this service in the WFQ is increasing in a stair case fashion. We will increase this W_i only when the last bit of the packet has been served.

Now, let us say particular packet is being served and this packet will leave at time d hat k. So, when this time it leaves at the time d hat k, we update the service W $_i$ by the amount of the traffic that has been served and that is proportional to the length of the packet. So, just before this time, the service received by the GPS, the service received by the WFQ will be equal to the service received by GPS minus the last packet.

(Refer Slide Time: 37:21)



So therefore, we can see that W $_{ij}$ that is the amount of traffic served by WFQ during an interval of tau to t will be greater than or equal to W $_{ij}$ in the fluid flow fair queuing or GPS (tau into t minus L $_{max}$ by r minus L $_i$) sorry this should be equal which in turn for the GPS; remember, the GPS has a latency of 0, so therefore this should be greater than or equal to maximum of 0 into rho $_i$ (t minus tau minus L $_i$) which will be equal to maximum of 0 or rho $_i$ into t minus L $_{max}$ by r minus tau minus L $_i$ by rho $_i$ which proves that the latency of the weighted fair queuing is L $_{max}$ by r plus L $_i$ upon rho $_i$.

Again, let us look at this point. What we are trying to say is that particular time t in the j'th busy period, the service provided to the weighted fair queuing at the particular time t will be equal to the service provided to the GPS at that time t minus amount of bits in that particular packet and we will consider it to be a maximum sized packet.

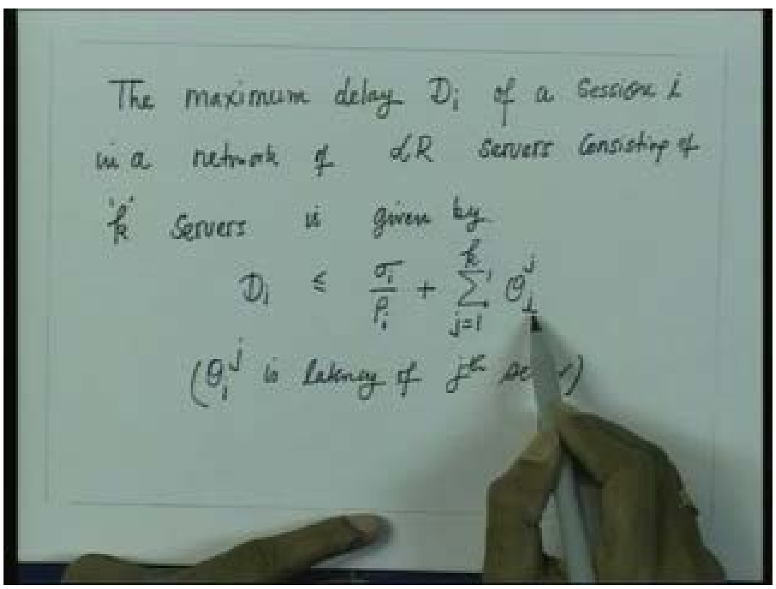
Now, since this GPS has a 0 latency, the amount of traffic served to session i during this j'th busy period and during an interval of tau to t minus L $_{max}$ by r will be greater than or equal to this quantity and from the simple algebraic manipulations we can show that the latency of a weighted fair queuing algorithm is given by L $_{max}$ by r plus L $_i$ by rho $_i$ which means that the delay bound in the case of a weighted fair queuing is given by sigma $_i$ upon the rho $_i$ plus L $_i$ upon the rho $_i$ plus L $_{max}$ upon r.

Now, note that this delay bound depends upon the parameters of the traffic which is sigma $_i$ upon rho $_i$ which indicates the burstenness in the long term average rate and L $_i$ indicates the maximum size packet, while the system diameters which are related to r which is the output link rate and L_{max} which denotes the maximum size packet among all sessions. So, in determining the delay, worst case delay, this is like you can say that effect of the other sessions in the scheduler and the effect of the traffic characteristics itself.

So obviously, the delay of a packet in a weighted fair queuing algorithm will be determined by the traffic characteristics as well as the characteristics of the other sessions in the packet schedulers. So, these 2 will determine the maximum delay. But all we can, we have actually proved that in weighted fair queuing we can give delay guarantees, provided the traffic is rho sigma regulated.

Now, this result we have proved it in the case of a single load case. That means that we are considering a particular router where this traffic flows are there and the scheduler is scheduling various flows in a WFQ fashion. Now, in a typical network, we will have a network of such nodes, multiple nodes. How does this result apply? That means what can we say about the end to end delay guarantees when we are having a multiple node case?

(Refer Slide Time: 43:32)



So, let us see what result holds true in the case of a multiple node case. Now, interestingly it turns out that in the case of a network of nodes the maximum delay maximum delay D_i of a session i in a network of latency rate servers consisting of k servers is given by D_i should be less than or equal to sigma $_i$ upon the rho $_i$ plus the summation of… where theta j is the latency of the j'th server. So, should be… So, what does it so this is i, so what does it means is that theta $_{i\,j}$ is latency of j'th server. For that particular session, of course this is independent of the session i. The latency is a property of the packet scheduler.

So, what interestingly turns out that if you have to provide the end to end delay guarantees in a network of these latency rate servers, then the delay is simply bounded by sigma $_i$ upon the rho $_i$

plus the sum of the latencies of all the packet schedulers. So, the delay guarantees actually is given by the sum of the latencies of all the packet servers.

So, this is an important result and what really we have proved is as again, I want to emphasize that we were discussing the packet scheduling algorithms from the point of view of providing fairness. But what we have really proved is that it is possible to give the delay guarantee that is the quality of service guarantees, provided the traffic is rho sigma regulated.

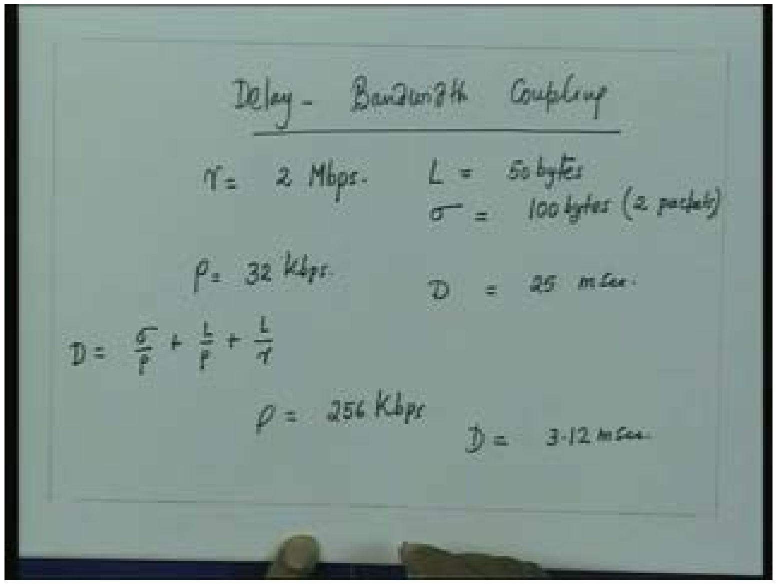
Now however, one thing remains here is that in the case of these packet scheduling algorithms, there is a coupling in terms of delay bandwidth. So, the delay bandwidths are somewhat coupled. How they are coupled? Let us see here that the particular traffic source wants a minimum bandwidth guarantees of a rho $_i$ which is equal to the long term average rate, a minimum bandwidth guarantees of rho $_i$.

Now, in order to give the traffic a delay D $_i$, it will be equal to sigma $_i$ upon the rho $_i$ plus L $_i$ upon the rho $_i$ plus L $_{max}$ upon r. That means if you fix the minimum rate rho $_i$, the delay bound automatically gets fixed depending upon the traffic characteristics. The 2 quantities that is the delay and the bandwidth, they are not decoupled. As you can see, from here that if you want to give a lower delay to a particular traffic flow; then it must have the higher allocated rate that is the rho $_i$, it must have the higher rate. It can only be done at the expense of giving the higher rate to a particular traffic flow.

Now, there may be situations in practical applications, for example like applications like telnet which wants not only the lower delays but its bandwidth requirements are also very low, It is also not it is not very high. Now, if you want to give this session a lower delay, then as you have seen in the weighted fair queuing, you can only give a lower delay by giving him by giving a telnet session a higher bandwidth.

Now, if you give a higher bandwidth, then obviously your multiplexing gain gets reduced because then you may not be able to multiplex other flows because you have already allocated a rate to this particular traffic flow which is more than its required rate.

(Refer Slide Time: 48.56)



Just to give you an example of how this bandwidth delays guarantees are coupled, let me just give you an example of this delay bandwidth coupling. So, let us take a specific example where we assume that r is equal to let us say 2 mega bits per second and let us assume these are all same size packets. So, let us say L is actually here equal to 50 bytes and let us say sigma $_i$ is actually 100 bytes that is equal to 2 packets.

Now, if you consider rho to be equal to 32 kilo bits per second, it turns out that the delay D is something like 25 milliseconds. This we just compute from sigma by rho plus L by rho plus L by r. So, this is 100 bytes divide by 32 kilo bits per second plus L is 50 bytes divide by 32 kilo bits per second and this is L - 50 bytes divide by 2 mega bits per second, actually we get 25 milliseconds. If we increase rho to something like 256 kilo bits per second, our delay reduces to 3.12 milliseconds. Our problem is that if you want to give a flow 3 milliseconds delay and at the same time the flow does not want more than 32 kilo bits per second; it is not possible to achieve this. In the case of a weighted fair queuing algorithm that is not possible to achieve.

So, in other words, what I was trying to point out is that the delay bandwidths are coupled in the case of a weighted fair queuing. To some extent, this decoupling can be achieved. We will see that later but there are some other problems also associated with the weighted fair queuing algorithms that I just wanted to illustrate and that problem was that even though the weighted fair queuing algorithms tries to approximate the fluid flow versions of the fair queuing algorithm by emulating the GPS on one side; it does not give the worst case fairness guarantees and I will explain that what does it mean by the worst case fairness guarantees.

Also on the other hand, the computation complexity of the weighted fair queuing is very high. But despite of all these deficiencies that we have seen that it does not keep the worst case fairness guarantees, its computational complexity is high and the delay bandwidth guarantees are somewhat decoupled. It still remains an important packet scheduling algorithms that has been deployed in practical routers even these days to provide the fairness guarantees or QOS guarantees to the different traffic flows.

(Refer Slide Time: 53:10)