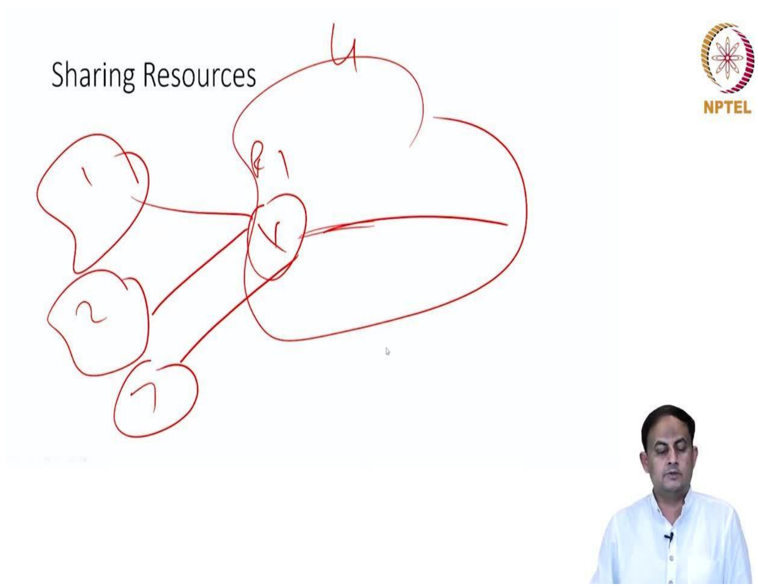
Advanced Computer Networks

Professor Neminath Hubballi
Department of Computer Science and Engineering
Indian Institute of Technology Indore

Lecture 20
Traffic Management – Part 7

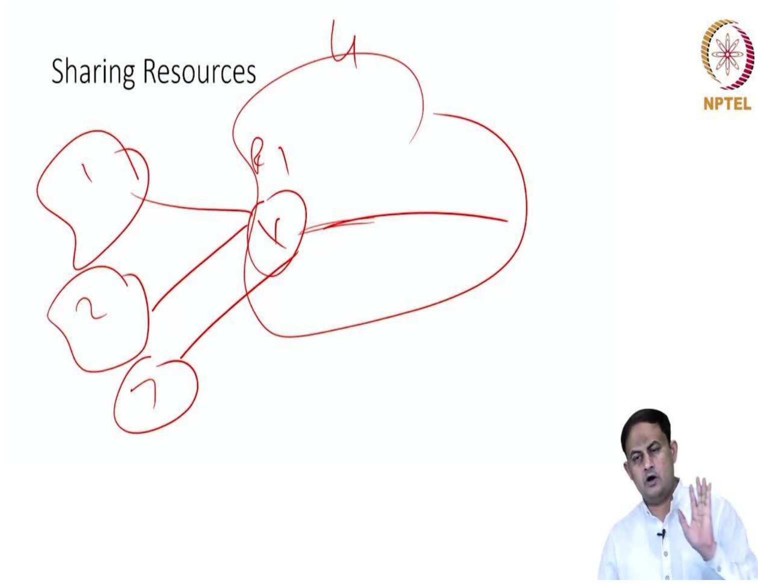
(Refer Slide Time: 0:19)

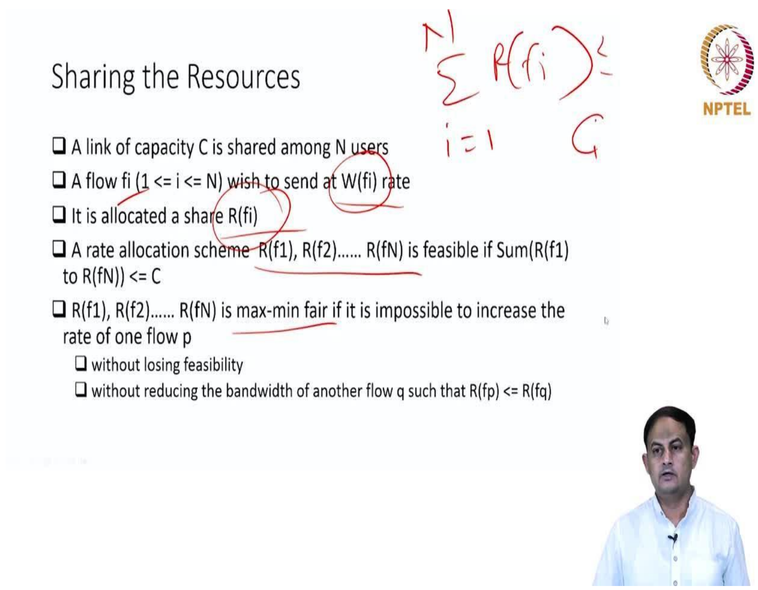


Now, what I said is this scheduler actually helps us to achieve sharing of resources. So remember the discussion, you have a router, you got capacity and then not necessarily one user, you might be receiving packets from so many networks, network 1, 2, 3, and this is the network, this is the router which is inside network number 4. The exit link has got a finite capacity, and you want to receive packets from all these networks and then transmit them on this link to the next hop.

So that is actually sharing of the resources you want to optimize. You want to transmit all of the packets, receive them from all of these networks, and send them to the respective next hop.

(Refer Slide Time: 01:17)





Now how does the scheduler actually work? So let us formalize the discussion of the previous diagram that we have got; these might be end users or other networks. So even if it is other networks, also we call it the users. So I want to assign a kind of allocation rate to each of these users, and the scheduler's job is to give that fair share to all of these users, and the network.

So the outgoing link has a capacity C, and there are N users to serve. The traffic from each user is identified by a flow, f_i is the ith user's flow, or ith network's flow coming to a router in your network, and it is transmitting at a certain rate called $W(f_i)$.

The ith user or ith network is transmitting at the rate of $W(f_i)$ right now, and I want to assign a priority to this transmission, I want to give a capacity of transmission. Assigning the priority means, now what we said is every two packets are picked up from Q1, six packets are selected from Q2, so if I want to do that, that is actually you are assigning the rate at which the user 2's traffic needs to be transmitted. So that selection or assignment of the weight is called as the $R(f_i)$, it is called as your transmission rate. The arrival rate is W(fi,) and out of that, you are actually transmitting at a rate of $R(f_i)$, this is called an allocation.

So the user is transmitting at a certain rate, you are allocating something that is called an $R(f_i)$, and what we want is the scheduler inside the router needs to come up with some allocation scheme, which is feasible. So when this allocation scheme is feasible? when the rate of allocation or rate of transmission for all the users put together is less than or equal to the available capacity. What it means is, so summation of the $R(f_i)$, where i is varying from 1 to N, N users are there, and the capacity allocated to each of these users put together is less than or equal to the available capacity C ($\sum_{i=1}^{N} R(fi) \leq C$)

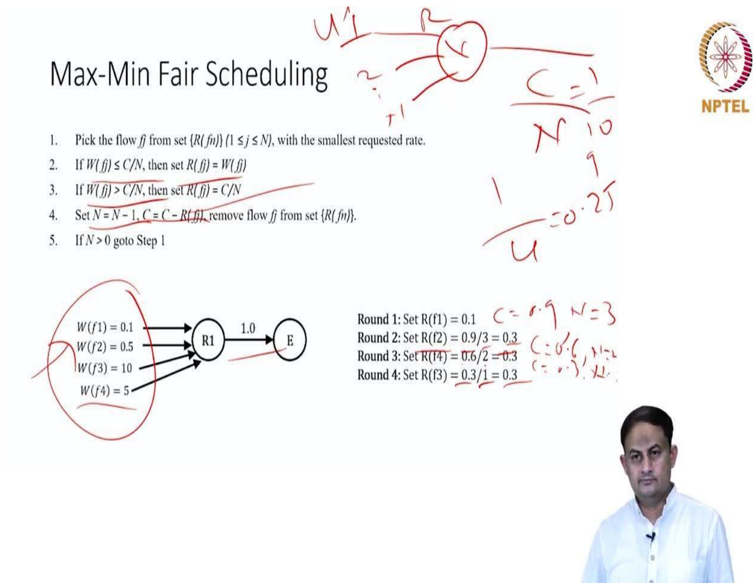
So let us say the outgoing link is 1 Mbps and the scheduler has come up with such a mechanism where each user's transmission rate is not or all the transmissions is not violating the constraint C, so they are put together less than the capacity of C, then only the allocation is said to be feasible.

So it is obvious that no single user or no single allocation is actually exceeding the capacity C. If the sum of the allocations needs to be bounded by this capacity C, then the individual user transmission is also bounded by the capacity C. So now the scheduler has to come up with this kind of vision.

So now what we do is the schedule is feasible when this constraint is met. I want to come up with such an allocation scheme, where without reducing the bandwidth of the particular user, I want to maximize the available utilization.

So I do not want to suppress another flow from another user, my peer user, but I want to maximize my own transmission, which means the router does not want to hurt me at the same time it wants to also utilize the available capacity and you want to give a maximum amount of the bandwidth that is available to me at this point of time. So that is where an algorithm called as the Max-Min fair algorithm comes in, and if this condition is violated, then the Max-Min fair algorithm is actually said to be not feasible

(Refer Slide Time: 06:02)



So let us see how exactly this Max-Min Fair Scheduling algorithm exactly works. So the way the Max-Min Fair Scheduling algorithm works is it picks up that flow every user is transmitting at a certain rate, and the jth flow, which is having minimum transmission rate among the N users, you pick up that user first and then see what it is asking for.

Look at this diagram, there is a router that has the capacity of this, and there are N users 1, 2, up to N, and C/N is the fair share of the capacity available to one user. And if user number 1 is transmitting at a capacity less than C/N, if let us say C is 1 Mbps, and the users are 10, so 1 Mbps is divided among 10 users and one-tenth of the 1 Mbps, 0.1 Mbps, if the transmission rate of the user 1 is smaller than C/N, then whatever he is asking for, that you allocate to the user number 1, that is what it says. And on the other hand, if the transmission rate of the user is exceeding C/N, then you reduce it, and the ceiling of that allocation would be C/N, whatever is the fair share; if its transmission rate is less than that, then you give the full capacity of that user, if your transmission rate is exceeding the fair share, then you restrict to the fair share that is what is done here. If $W(f_j) > C/N$, then $R(f_j)$ allocation is restricted to C/N.

And then, once you do the allocation, you update the number of users that are transmitting and the capacity that is available. Once you do one allocation, the number of users will be reduced to 9 and the capacity will be appropriately reduced by whatever allocation you have done.

So that is how you do it, you iterate it, you take one user, and ask what is he asking for? Is he asking for something more than the fair share? if it is, restrict it to fair share. If its transmission rate is less than or equal to its fair share, then you give him whatever he is asking for. That is what maximizes the transmission by serving the people who are asking for the minimum amount of the share that is what the Max-Min Fair scheduling is.

So here is a picture which is actually showing four users and the outgoing link's capacity is 1 unit, 1 unit, might be anything, might be Mbps or the GPS, let us assume 1 Mbps and these users are also asking for in Mbps. Flow number 1 is 0.1 Mbps, flow number 2 is 0.5 Mbps, flow number 3 is 10 Mbps, and flow number 4 is 5 Mbps, or you can call them as users. And those packets are arriving at router R1, but the outgoing link capacity is only 1 Mbps. Now how do I do the scheduling? How do I assign the weights to the different users' traffic is the question.

So in round R1, what you do is, pick up the user who is asking for the lowest quantum of the bandwidth, which turns out to be user 1, and he is asking for 0.1 Mbps. And what is the fair share now? There are four users, N is 4, and the capacity is 1, 1/4 = 0.25. So the fair share of user 1 is 0.25, the peak transmission rate is 0.1, which is less than 0.25, you allocate 0.1 Mbps to that user, whatever he is asking for, you just give them, and now you update.

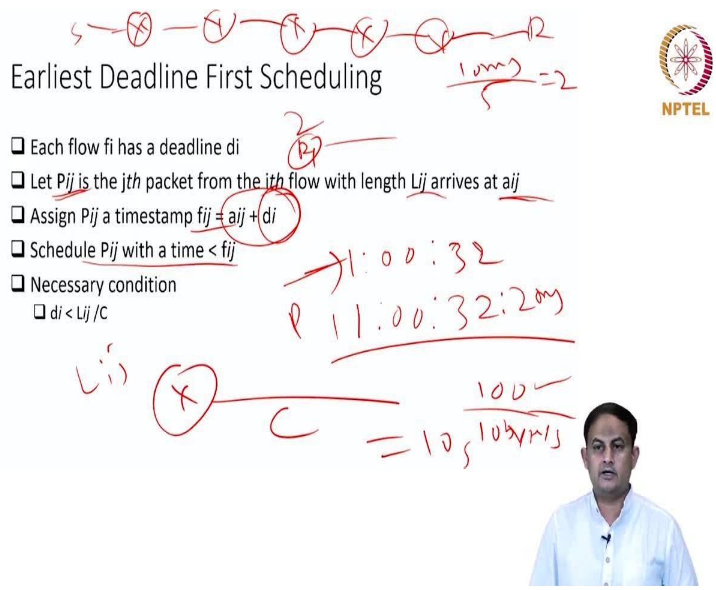
So what do you update? Now after you do this allocation, the capacity would reduce to 0.9, 0.9 Mbps is available, and the number of users which are yet to be served is reduced to three, that is what happens in round 1. Now in round 2, the available capacity is 0.9, and there are three users, the fair share here becomes 0.3; you look at the next user who is having the lowest demand. user number 2 is asking for 0.5 Mbps, but the fair share already turned out 0.3 Mbps. So you give them, $R(f_2)$ 0.3 Mbps, and you update the capacity. Now 0.6 is left, and the number of users who are actively served is 2. That is your second round.

And in round number 3, who is the user who is having the next lowest demand is user number 4 and the available capacity is 0.6. The user to be served are two, and now the C/2 is going to be 0.3 and he is asking for 5 Mbps, you give him 0.3 Mbps.

And the last one which is having the highest demand is user number 3, now again, you update after round 3. The available capacity is 0.3 and N is equal to 1 now in the fourth iteration. So 0.3 is the leftover capacity one user is left to be served. So 0.3 is the bandwidth, 0.3 Mbps is the bandwidth given to user number 3. So that is how you maximize the capacity allocation to the users who are having less demand and curtail the transmission rate of the other users, which are actually transmitting at a higher rate that is what the max-min fair scheduling is.

So this max-min fair scheduling is bringing the notion of fairness among the prioritized transmissions. So when I say user 1, 2, 3, 4 is transmitting now, that is an instantaneous transmission. If I do not have a transmission from user number 4, then only among the three, we adjust the transmission rate and then you do the transmission, that is how the max-min fair scheduling actually works.

(Refer Slide Time: 12:54)



Now, this is one kind of scheduler, and I want to end the discussion on traffic management with another scheduling algorithm called the Earliest deadline first scheduling algorithm. So this Earliest deadline first scheduling algorithm comes from the notion of end-to-end delay.

So particularly if you can recollect your earlier discussion that the audio traffic has to be delivered at the other end at a certain deadline. The source is here, and the receiver R is here, and the packets are going through a bunch of intermediate routers, and it is reaching the receiver R, and the end-to-end delay is let us say 10 milliseconds and the number of the routers in between is, let us say here, in this case, is 5. And at each router, the maximum amount of time that it can spend turns out to be 2 milliseconds, within 2 milliseconds this router has to forward that particular packet.

Now that is the constraint on the individual router given that the number of the routers in between the source and receiver is 5, and the 10 millisecond is the total end-to-end delay time. Now what I do is when the packet arrives at the input queue of any router, I am going to assign a deadline for that packet to exit that router, so within 2 milliseconds, this packet has to exit.

So, if a packet P_{ij} is the jth packet from the ith flow, ith flow means ith user or ith network whatever you want to call. From the ith user, the jth packet has come and it is carrying L_{ij} amount of the bytes inside that, and that packet itself has arrived at for timestamp a_{ij}, so a_{ij} is the timestamp.

Let us say the timestamp now is 01:00:32, and 2 milliseconds is the deadline. So I am going to set a deadline for this packet as 01:00:32:2 ms, so that is the time by which this packet P1 has to exit the router and go to the next hop.

So that deadline time stamp is called as the f_{ij}, f_{ij} is 1:00:32:2 ms. How did we arrive at it? So the timestamp at which the packet has arrived plus whatever the deadline d_i, d_i is the ith routers timestamp, whatever the delay you can afford at each of your routers that is d_{ij} and I add these two and then assign this number.

I need to schedule this packet such that this packet P_{ij} exits the router within the timeline; whatever the deadline that is set, f_{ij} is the deadline for the jth packet from the ith network or the ith user, so that condition is met there. So I am going to transmit that packet which has got the earliest deadline that is going to transmit it first.

And it turns out that because the packets have got a variable amount of data and C is the capacity or the rate at which the bytes are transmitted from this router and L_{ij} has got a variable amount of the number of bytes in it, let us say the rate at which the packet is transmitted is the maybe 10 bytes per second and the L_{ij} has got the 100 bytes, so the transmission time for this packet is going to take 10 seconds and a necessary condition for the earliest deadline for scheduling to work is you have to meet this deadline that can happen only when your d_{ij} is smaller than L_{ij} divided by the rate at which the packet is actually transmitted ($d_{ij} < L_{ij}/C$)

So what it means is, if packet P_{ij} arrives at the condition that I need to exit this router number R1 within 2 milliseconds and the amount of the data that is inside this packet is actually 100 bytes, and the rate at which transmission is done it is taking 10 milliseconds, the 2 milliseconds transmission deadline cannot be met.

So if that is the situation, then the scheduling itself will say that it is not feasible. A necessary condition for this kind of scheduling to work is $d_{i,}$ the deadline at an individual router for any packet needs to be smaller than the transmission time of that packet, how many bytes are available in that packet, and how much time it takes to transmit that packet, the deadline cannot be smaller than that. So if it is the deadline is bigger than that, then you can always do the transmission.

So a necessary condition is that the transmission time needs to be smaller than the deadline time, so then only the scheduling can work. So this turns out to be a necessary condition, but it is not a sufficient condition. Having said that, if every transmission takes less amount of time than the deadline that packet has got, then also the schedule earliest deadline for scheduling may not be feasible. So I will leave it as an exercise to figure it out why this is not a sufficient condition.