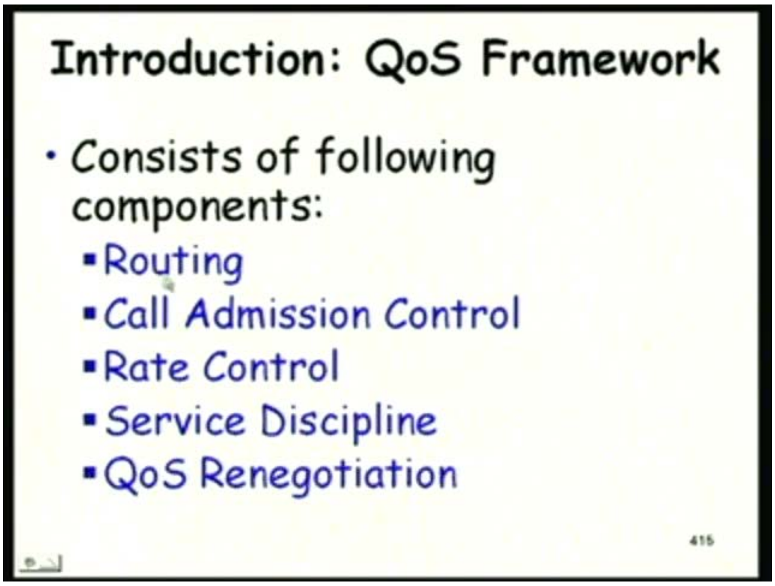
Real-Time Systems
Prof. Dr. Rajib Mall
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Module No. # 01
Lecture No. # 38
Real Time Communication over Packet Switched Networks (Contd.)

Good morning. So, let us get started. We will continue from what we were doing, last time we could not complete the topic, that was quite a big topic that we were discussing - that is, real time communication over packet switched networks. So, let us proceed.

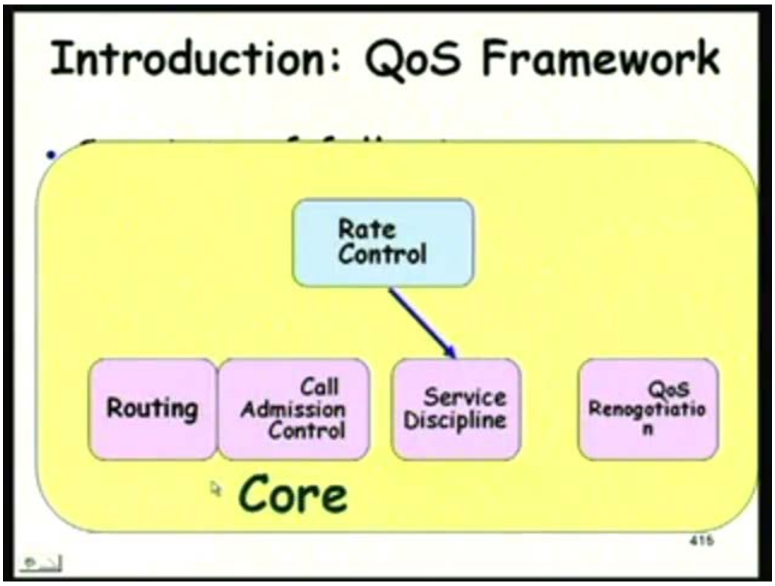
(Refer Slide Time: 00:36)



So, we had seen that packet switched networks are becoming commercially important for real time applications. For example, on the internet, you might use the internet to control certain real time devices or may be like to carry out real time transactions, for example, banking or other transactions on the internet. So, we are trying to see how real time communication can be supported in a packet switched network and we had seen just to recollect, what we were discussing? We were seen that the traffic characteristics of the source has to be indicated to the network and also the specific quality of the service requirements.

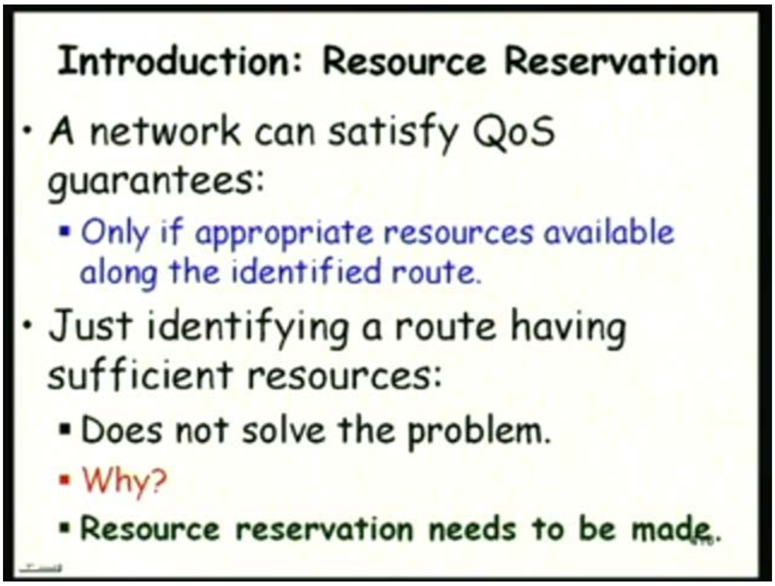
So, the network in turn checks whether the required quality of the service for the specified traffic can be supported or not, and then it once it agrees to support the traffic, it continues to provide the required quality of service. So, we are trying to discuss the quality of service framework that a communication network could use to support the specified quality of service requirement, and we had see that the different components in the quality of service framework, that implement the guarantee quality of service guarantee, include - the routing module, the call admission control module, the rate control module, the service discipline module and the quality of service renegotiation module - all of these play a part in guaranteeing the quality of service requirement of an application. And if we think of it, the responsibility actually boils down to different routers in the network and we had seen that the routers can be edge routers or core routers.

(Refer Slide Time: 02:45)



Let us look at any of the router, may be a core router. So, we will have different modules here: the routing call admission control, service discipline and quality of service renegotiation and the rate control. The rate control, it examines the incoming traffic, and then it either uses a policing or it uses resetting to get the traffic into the agreed form, and based on that, it gives to the service discipline and the service discipline in turn, based on the reservations made for the specific communication, it sends the packets out on the link. So, this we had seen yesterday, and we had seen to some extent the routing.

(Refer Slide Time: 04:37)

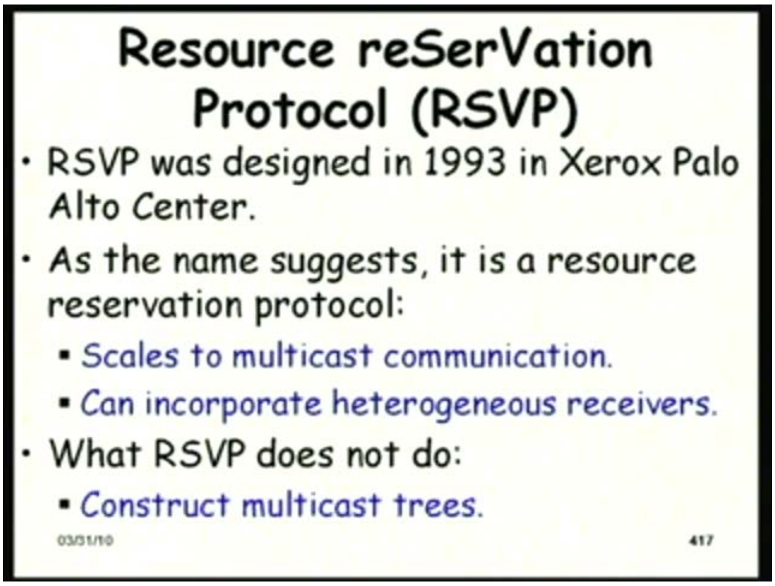


Now, once the request for a call comes, not only a path has to be established by the router, which can support the specified quality of service, but also the resources need to be reserved, because just checking whether a path exists with a specific quality of service will not guarantee that the quality of service will be met, unless it is reserved for that specific connection.

So, a network can satisfy a quality of service guarantee; the first requirement is that appropriate resources must be available along the identified root, for example, bandwidth, buffer storage and so on, but just checking that resources are available on a root, does not indicate that the quality of service guarantees can be met, unless we reserve the resources that is what we are just saying.

Just identifying that a path exist with sufficient resources, does not solve the problem; because if too many connections are undertaken on that. It will become congested and it will become difficult to guarantee; so, reservation has to be made. So, let us discuss, how resource reservation can be made.

(Refer Slide Time: 06:07)



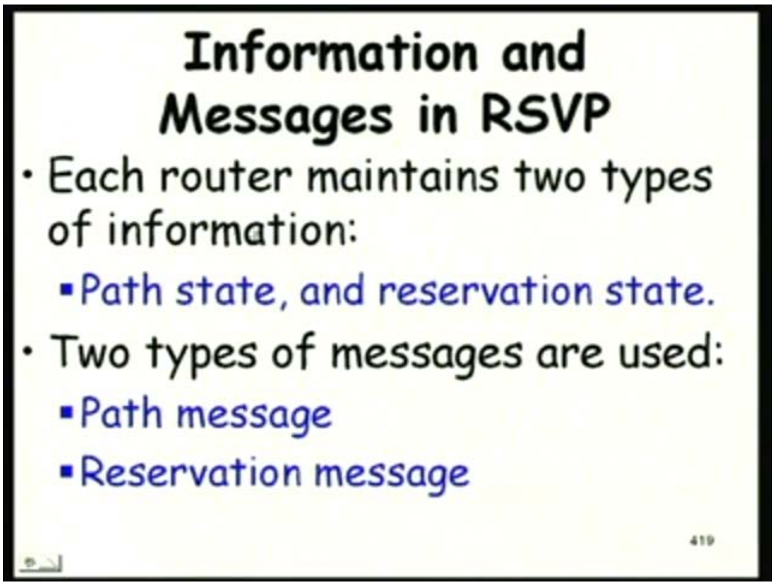
So, here the most popular protocol is the resource reservation protocol RSVP. If we understand RSVP, then we understand how resource reservation can be made in a packet switched network, for example, the internet; and of course, there may be small variations of this main theme of resource reservation. It was designed way back in 1993, at the Xerox Palo Alto research Center; and we should notice that it is actually, a resource reservation protocol, we should not associate other responsibility with this, it just does the resource reservation. For example, identifying a path with required resources etcetera is not the responsibility of resource reservation protocol, but this protocol can work on multi cast communication; it can scale to multi cast communication for a single communication path of course, it will work, but even it is general enough to work in a multi cast communication environment, and also it can incorporate heterogeneous receivers.

Heterogeneous receivers, by heterogeneous receiver, what we mean is that receivers requiring different quality of service from the network; for example, there may be a mobile phone or a hand held device, which requires data in a gross form right., so little data transfer, I mean smaller amount of data transfer may be necessary; and another one may be, another receiver may be a desktop; so, which can receive data at a faster rate? So, this is what we mean by heterogeneous receivers, but we must remember that RSVP is just a resource reservation protocol, it neither constructs multi cast trees that is routing is not its role; neither is its role to make admission c ontrol decisions. So, those are separate modules.

(Refer Slide Time: 08:15)

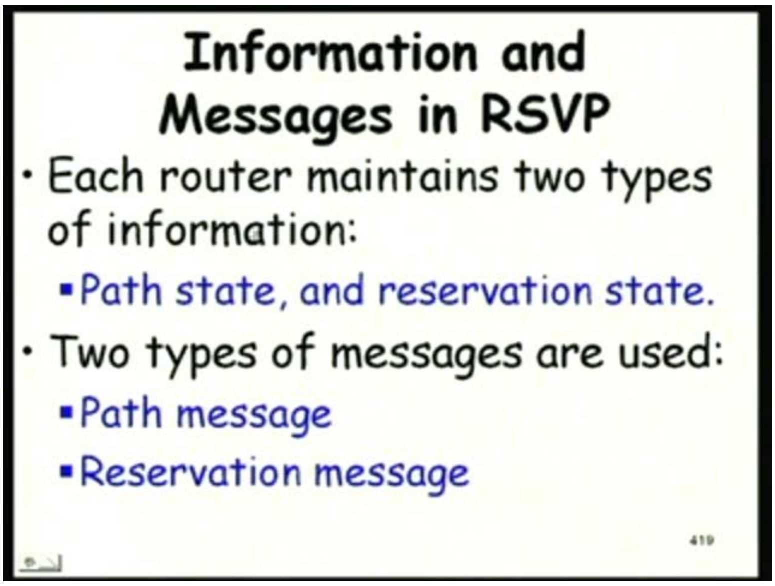
So, let us understand how RSVP will work. Now, as we were saying that, there can be multiple destinations for one or more sources, and for each destination, RSVP maintains a sink tree that is the terminology used. The sink tree is actually a spanning tree, which is routed at the destination; starts form the destination and it spans or connects all the sources in the multicast group; and the sink tree is used to send control messages from the destination, so each destination will have its own sink tree; starting from that destination and connecting all the sources in the multi cast group, and the purpose of the sink tree is that the destination keeps on sending control messages on this, the sink tree. Now, before we discuss further, let us just understand what are the information stored or information storage requirement for this protocol and what are the different types of messages that are exchanged in RSVP.

(Refer Slide Time: 09:40)



In this protocol, every router maintains two types of information: one is called the path state, and the other is called the reservation state. So, we will just elaborate this in a minute. So, let us just understand that right now, let us understand that two separate categories of information is maintained to implement the protocol; This information is stored at every router, which store the path state in the reservations state; and there are two types of messages that are used in this reservation protocol: one is called as a path message, and the other is called as a reservation message. So, let us try to understand, what are these two types of information? And what are these two types of messages? And then once we look at this, we will be able to understand how the protocol operates.

(Refer Slide Time: 10:55)

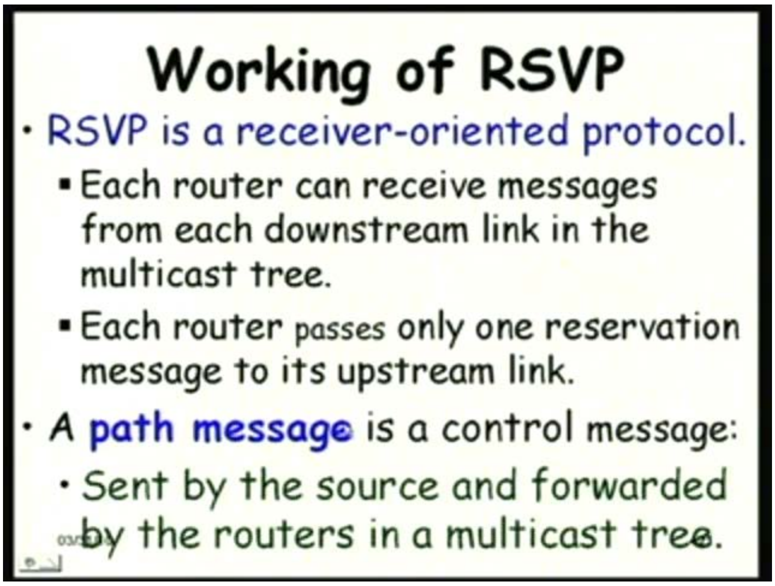


First, let us look at the information storage; one we were saying is the path state; the path state is stored at each router, and it is the path used by the multi cast group from that router. So, it just maintains that, from that router what are the paths that need to be taken; that is why it is stored at each router. So, basically the router would know that how is the path established or once there is a message, where it should send. So, the path state is used to maintain the sink tree of the destination.

(Audio Not Clear. Refer Time: 11:45 to 11:51)

The sink tree is for every multi cast group; see over a network, there can be several multi cast groups that might be operating right. Now, for every multi cast group, there will be each router will store one path state. So, for every multi cast group, it will store different path states, as many multi cast groups are operating on the network, that many path states at the router has to be maintained, and another information is the reservation state. The reservation state is basically, the resource reservation, information about resource reservation. So, for different destination, what kind of resources needs to be reserved? We have said that different destinations might have different quality of service requirements, and it is typically maintained in one or more tables.

(Refer Slide Time: 13:00)

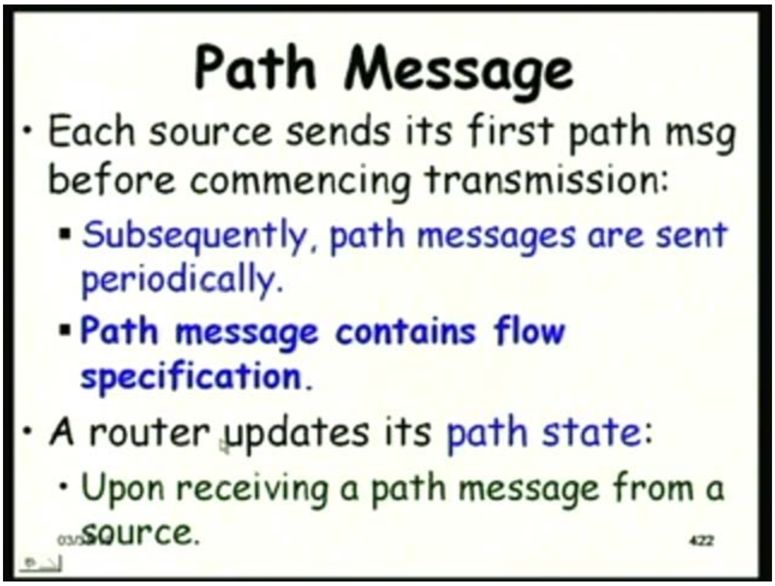


So, now let us look at the working of RSVP, and here we will also discuss about the messages, the two types of messages which we which we told; actually, this is a receiver oriented protocol, because it supports heterogeneous receivers, multiple of them; and each receiver has its own quality of service demand. Now, if we look at any one router in the network, it can receive messages from the downstream link in the multi cast tree. So, from the sink tree, the destination sends a message to a router and that in turn forwards it to the router on the upstream.

So, the one that is connected, the router that is connected to the destination will first receive the message from the destination, and which will be forwarded upstream that is what it says that each router can receive messages from each downstream link in the multi cast tree, because there are several receivers. Now, each router passes only one reservation message to its upstream link.

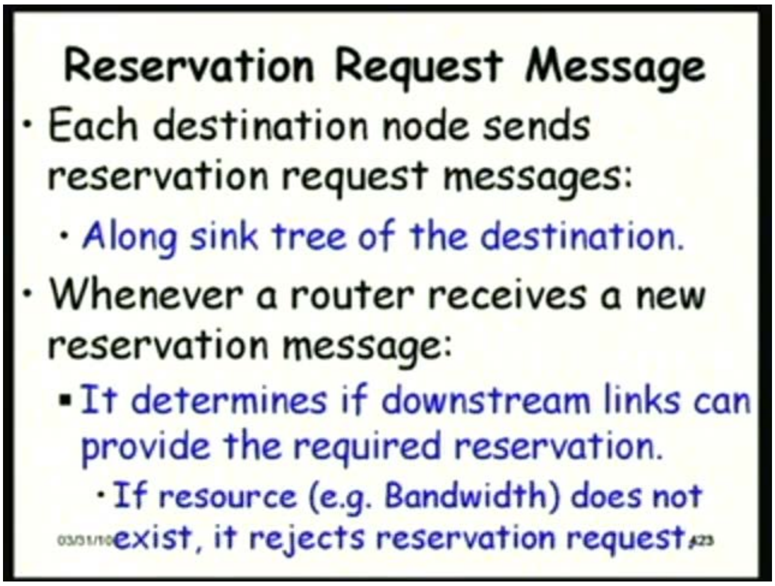
So basically, even though there are multiple data, it might receive through different links, it just uses only one to avoid duplication. It just sends one reservation message to its upstream link. So, this is the reservation message and we had also told about a path message. A path message is a control message, both path and reservation message are control messages; and the path message is sent by the source and is forwarded by the routers in the multi cast tree. So every router in the multi cast tree would receive the path message.

(Refer Slide Time: 15:24)



Each source before it starts to transmit, once it needs to transmit, it sends the first path message; and after sending the first path message, it subsequently path messages are sent periodically; why is that? Is it not enough that a path is established and it need not send any further; why does it need to send periodically? We will see the answer, but the main idea here is that we are talking of a dynamic network where different multi cast trees become active, inactive, their quality of service requirements change and so on. So, the path message contains flow specification. It is a traffic specification and we had seen that earlier and a router would update its path state upon receiving the message form a source. So there may be several sources and the router would get path messages from each of the source, and it will keep this in a separate path state. So the path message is used to provide path information; the path state up to the routers.

(Refer Slide Time: 17:03)



Now each destination node, see once the source sends the path message, the first path message is sent, and then periodically it sends. So once a destination receives a path message, it might want to receive that information. So, that stream of information may be needed by the destination. So, what it will do is, it will send a reservation request message. So, once the destination node after getting a path message, decides to receive the stream, it sends a reservation request message; so this will be sent along the sink tree of the destination. So, to the router connected to these and then on to the upstream and each router in the upstream sends only one one reservation request to the upstream to its upstream, and whenever a router receives a new reservation message, it checks that whether the link can provide the required reservation.

So, it gets a reservation message, and checks whether the link that is connected to it. The downstream link can provide the required reservation. So, are you clear, why it is a downstream link and not upstream link? Just think about it. So, it determines, if the downstream link can provide the required reservation; just think about it.

(Audio Not Clear. Refer Time: 18:56 to 19:01) we have to decide whether the receiving node is ready to…

Not really it is the link. It is not the node see the downstream link…

The the the receiving node may be giving fragment also that time they cannot be reserved also.

No, see for receiving node only we are… see, if you are wanting both sender and receiver to work in both modes; that is a sender can send and receive, and the receiver can send and receive, then we will have to multi cast trees, separate multi cast trees. But here we are just talking of one multicast tree where there are specific senders and specific receivers right; and it is the downstream link, because see the destination sends the reservation message to the edge router that is connected to it right, and that edge router will check whether the link to the receiver can support the required quality of service, because it has control over that. It does not have control over the incoming link to it. It has only control over the outgoing link right. So, that is the reason.

So, if the resource, for example, bandwidth does not exist, it rejects the reservation request. So, every router keeps track of the resources that are being used, and by resources we mean the bandwidth and so on, that are on its outgoing link right, incoming link, it does not keep track of anything; it just receives messages on that right packets on that.

(Audio Not Clear. Refer Time: 20:44 to 20:47)

No, no, see it does not have control over, for example, bandwidth allocation on the incoming link.

No, for outgoing link?

For the outgoing link, it provides reservation if it exists. So, if the bandwidth has already been allocated to different multi cast groups, then it will reject it right. So, that is what it says; it checks if the downstream link can provide the required reservation, so let us not right now confuse with rate control and so on, because as I can see that your confusion is with rate control, rate control is not the responsibility of the reservation module right. So, we are just talking of RSVP, whose sole responsibility is to provide the required reservation of the required resources; it neither establishes the path nor does admission control and so on. It only depending on the requirement of the destination, it just makes the specific reservations, if available.

(Refer Slide Time: 22:01)

The destination sends its first reservation message, when it receives a path message from a source, whose message stream it wishes to receive; and we had said that the source actually sends the first path message, before it starts to transmit; and then periodically it sends path messages and receiver at any point of time after receiving a path message. It might want to receive that stream and then it sends a, the first reservation message; and then either the reservation is or not right, reservation could successfully be done or it may not be done, and once the reservation is successful, to maintain the reservation the destination periodically sends reservation messages; why is that because, is not it enough that the reservation was made and should not it work, why should it send periodically reservation messages. The answer is that, the applications that are there can be dynamic in nature; they stop to receive, they might change their quality of service requirement and so on.

So, the applications that we are considering, some of them can be dynamic in nature and therefore, it is necessary for all connections, all receivers to send periodic receiver messages, just to hold their reservation. Even that was made, just to keep them in a reserved state they need to send periodically reservation messages, but what if the receiver at the end of the communication, sends a message that do not need the reservation, please deallocated will that not work; see, that is the alternative scheme; see, here it is periodically sending a message saying that hold the reservation, hold the reservation and so on.

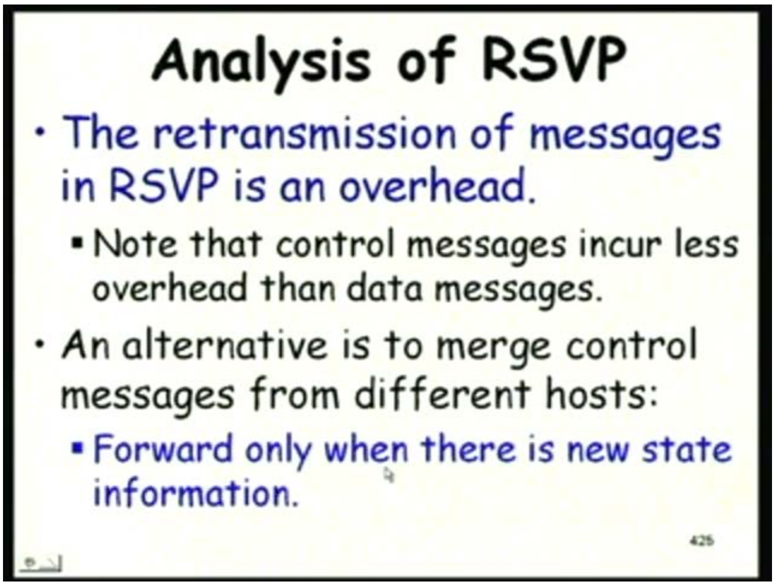
But cannot it just work, if it is whenever there is a change in the reservation requirement, either it does not need any further packets or it need changes its quality of service. Cannot it just send a reservation message and that would save lot of overhead, what do you think? Yes will it work or not work? Is the… What I am saying is it clear? I am saying that instead of periodically sending reservation messages, would not it be ok; if the receiver does not send it periodically, but only when there is a change of state, either it does not need any further packets or its quality of service requirements changes, would not it work, because that will reduce the overhead.

(Audio Not Clear. Refer Time: 25:16 to 25:25)

No, that the receiver knows; event delivery is not a problem, because the receiver knows that when it is not going to receive further message. No, I do not think so. See, the main problem here is that what if the receiver just dies right; the receiver node, there is a link failure or something to the receiver or may be the receiver just crashed. So, what will happen? The reservation will be held forever right that will be wastage of resource.

So, to avoid such problems needs to periodically send messages that it still needs to hold the reservation. It is still interested in receiving the packets, and this information is the soft state about the receiver, because it you need to send reservation messages to keep it alive. It is not that you have the reservation state, and then it just goes on until you say that you are not interested. It is a soft state if you do not refresh it, then it will be de-allocated. So, the protocol by itself is very simple; only two messages are there and two information need to be maintained and reservation is made.

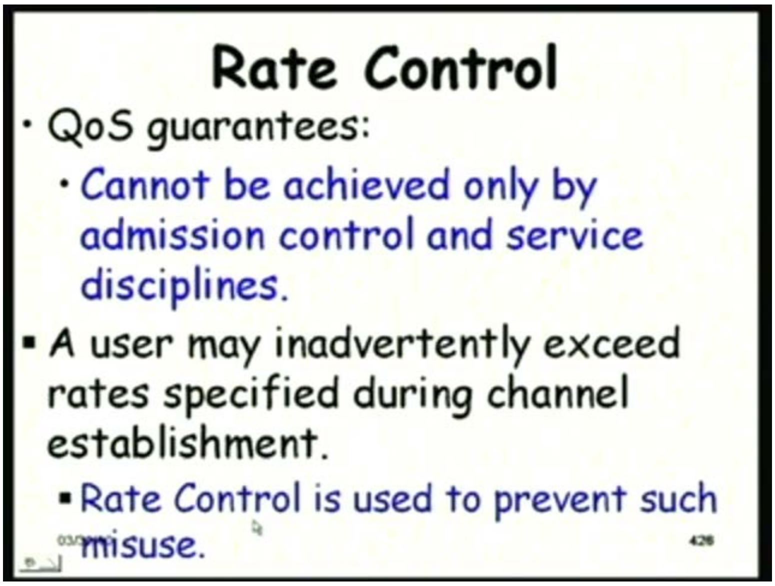
(Refer Slide Time: 27:04)



Now, let us try to analyze the RSVP. So, here the retransmission of messages in RSVP is an overhead. Actually, the path messages are sent periodically; and also the reservation messages are sent periodically; but one thing we must understand that a control message incurs much less overhead than data messages. So, the data messages are bulky, right the packets basically, the data messages are basically the packets; these are bulky and these have lot of overhead, if we you know keep on sending packets, but control messages are small information, and it incurs much less overhead the sink tree; one possibility would be to merge the control messages from different hosts and just change just forward only when there is a new state information.

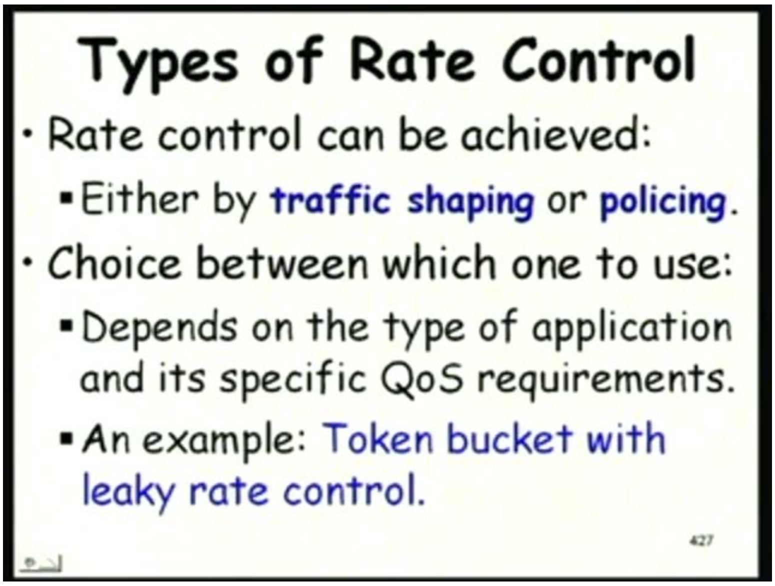
So, the different hosts they keep on sending messages and the intermediate one just checks, whether there is a change, if there is a change, it propagates that otherwise it does not. So, that is a possibility, but this is a the RSVP is the simplest resource generation protocol and is used widely most popular. We can very easily claim that, it is a very popular protocol. Anywhere you talk about resource reservation in a packet switched network, we will have to consider RSVP

(Refer Slide Time: 28:58)



Now, let us look at the rate control; we had said that the quality of service guarantees cannot be provided just by doing admission control and the service disciplines. Why is that, because the user might inadvertently or may be maliciously exceed the rate specified during the connection established, and rate control is used to prevent such an issues. But even otherwise, even when the user is perfectly well behaved, but still the packet might loose its shape. Sorry the traffic that is generated by a connection can loose its shape and it will become unacceptable, even though the sender was well behaved, why is that? We will just examined that shortly that even when the senders are well behaved, the traffic can get distracted and it will not be possible to provide the guaranteed quality of service. So, we will just examine that issue in a minute or so later; now let us see the types of rate control that are applied.

(Refer Slide Time: 30:22)



One rate control is called as traffic shaping; here, the traffic becomes out of shape, becomes busty, and then we need to reshape it to the original traffic specification; the other is policing, where we check whether the traffic meets the required characteristics for the traffic, I mean the agreed characteristics for the traffic; if not, reject it; the ones that are not satisfying reject them.

Now, the choice between which one to use, which type of rate control depends on the application and the specific quality of service requirement and the characteristics of the application, whether it can tolerate delay; whether it cannot tolerate delay; whether it can tolerate jitter and so on. So, we will check different traffic shaping policies; we will discuss the token bucket with leaky rate control, but let us check the advantages, relative advantages and disadvantages of traffic shaping versus policing.

(Refer Slide Time: 31:37)

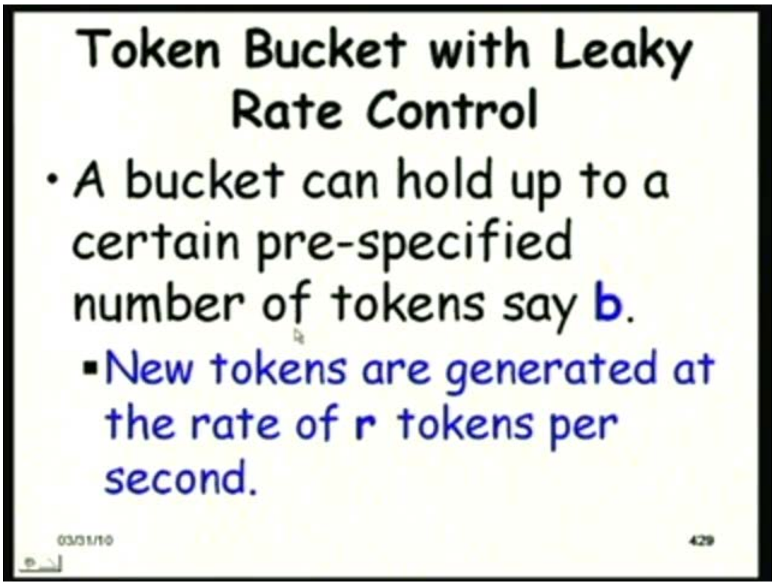
Traffic Shaping vs Policing

	Traffic Shaping	Policing
Objective	Buffers packets that are above committed rates.	Drops excess packets over committed rates. Does not buffer.
Handling bursts	Uses a leaky bucket to delay traffic, achieving a smoothing effect.	Propagates bursts.
Advantage	Avoids retransmission due to dropped packets.	Avoids delay due to queuing.
Disadvantages	Can introduce delays due to queuing.	Can reduce throughput of affected streams.

The objective of shaping is to buffer the packets that are above the committed rates. So, if a burst comes, it does not transmit them, it just buffers them, and then it smoothens out the traffic; so that, it meets the required traffic characteristic. But policing it just drops the packet over the agreed rate, there is no buffering involved in policing; now in traffic shaping typically a leaky bucket algorithm is used to delay the traffic; and the leaky bucket is basically like, you have a bucket, and then you want to transmit the data at a constant rate right. The the packets come and get accumulated in the bucket, but the leak or the one that is being transmitted from the bucket is a constant bit rate traffic. So that is the essential idea behind a leaky bucket traffic; a leaky bucket algorithm for delaying the traffic to achieving a smoothing effect. The policing on the other hand, it does not do any smoothing, as long as it is in the… Within the agreed maximum rate it just transmits. So, even if a burst has come, it will transmit the burst, as long as it is within the agreed maximum rate off transmission. So, bursts are not handled by policing well the traffic can become bursty.

Now, the advantage of a traffic shaping is that, it avoids retransmission due to drought packets; it does not drop packets actually it just delays them. On the other hand, if you have policing, there can be delays due to retransmission, I am sorry, it avoids delay due to queuing, but there can be retransmissions, see queuing delays are not there, but retransmission delays can be there; and here, we can think that when it is queued, it is delayed right; when we queue a packet it gets delayed, but here in the policing, the throughout of the streams are affected, because there are retransmissions right. So, these also has delays here it avoids queuing delays, but there can be retransmission delays, and again there is a overhead retransmission, and it can reduce the throughput of the network, whereas here the throughput is not affected, but there can be delays due to queuing. So, these are the choice that we have to make based depending on the network and depending on the application; whether a traffic shaping rate control would be suitably or a policing rate control will be suitable or a mixture of both.

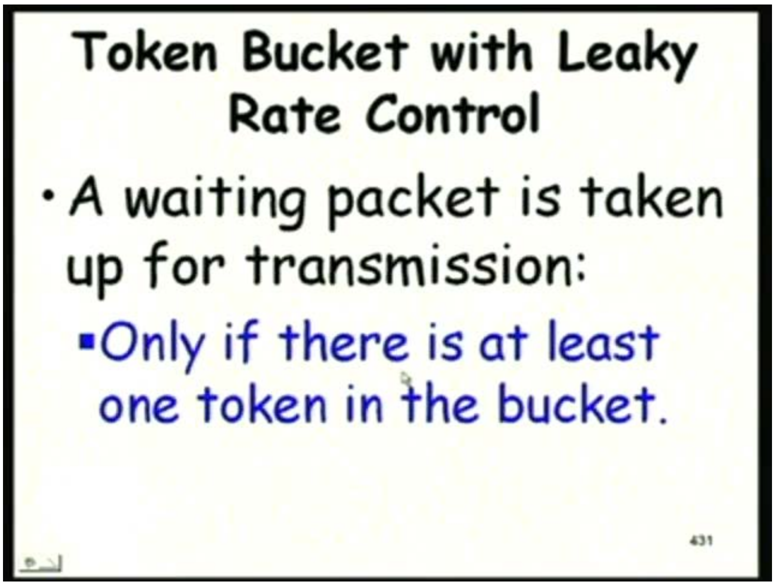
(Refer Slide Time: 34:57)



Now, let us look at the token bucket with leaky rate control; this we are saying is a very popular traffic shaping algorithm, and different variations of this algorithm are used for traffic shaping. Now, the main idea behind this algorithm is that there is a concept of a bucket, and the bucket can hold certain number of tokens, let us say b tokens it can hold. Now the network generates, let me not say network; it is the because these the buckets are there with every router, right the rate control is applied at every router. So, each router generates r tokens per second; these are parameters to be optimized based on the kind of network we are talking of, kind of traffic we are talking of, and the kind of application etcetera. So, b and r are… Can be based on the designers choice

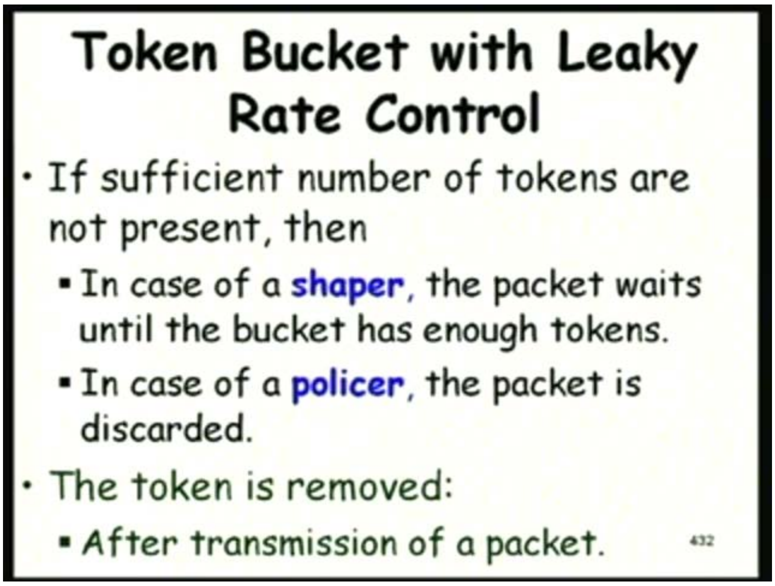
The bucket size is b can hold up to b tokens, and the router generates r tokens per second; and whenever it generates a new token it tries to put in the bucket, but if there is less then b tokens that is some tokens have been used up, then it will fill the bucket, it is added to the bucket, if the bucket contains less than b tokens, but if the bucket is already full, there are already b tokes, then the generated token is ignored.

(Refer Slide Time: 37:03)



A waiting packet is taken up for transmission, only if there is at least one token in the bucket.

(Refer Slide Time: 37:17)



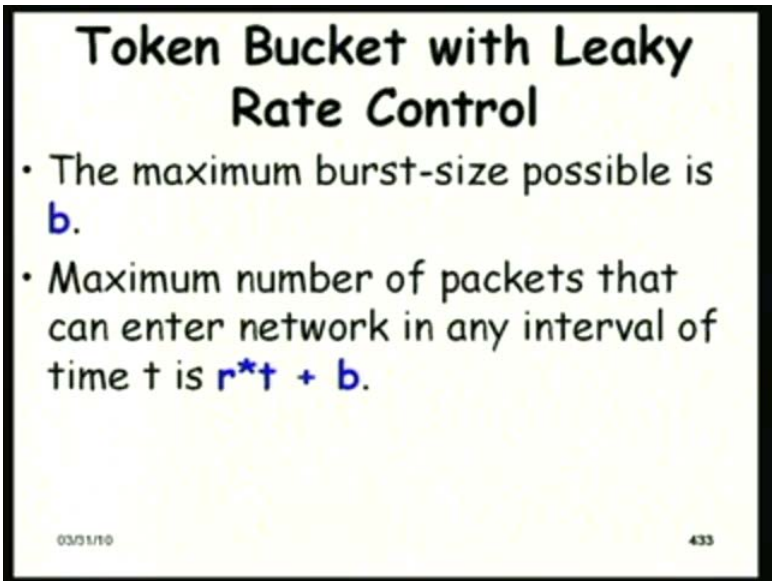
But what if, if sufficient tokens are not present. In a shaper, the packet will wait until the bucket has enough token. So, the same token bucket can even be used as a policer, where if the token is not there, the packet is discarded; and after the packet has been transmitted, the token is removed. So, what do you think, b indicates here, we discussed the protocol more or less. So, what is the role of b here? And what is the role of r here? Yes, can anybody answer? That we almost discussed the token bucket with leaky rate control where the bucket can store b tokens and the rate of generation of tokens is r.

So, what can you say about the traffic that will be generated by this algorithm whether you are…

(Audio Not Clear. Refer Time: 38:36 to 36:43)

Exactly. So, basically it says that r is the average rate of transmission right; r packets can be transmitted per second, but there can be if you look at a long term, if you look at a long term over a large period of time, the average packet that can be transmitted will be less than r; do you agree with to that? And the maximum burst size, as he says will be b right; the burst size will be restricted to b; let us proceed.

(Refer Slide Time: 39:18)



So, that is what it says the maximum burst size possible is b, and the average rate of transmission is r; r packets can be transmitted per second; so, the maximum number of packets that can be transmitted over any interval is r into t plus b. Do you agree with this? During any interval the number of packets is less that r into t plus b; whether you are taking of long term or short term, the number of packets during a interval t will be less than r into t plus b.

(Audio Not Clear. Refer Time: 40:04 to 40:08)

No, see b is already the…

(Audio Not Clear. Refer Time: 40:09)

Exactly. b packet… The token is already full with b, you know, so at any instant, b packets can be transmitted one after other right, but r is the rate of generation. So, after every second, r tokens will get added; not after every one second, r tokens are added at I mean periodically or equally spaced r equally spaced tokens are added per second. So, if you look at the burst size is b right; so, if you consider a time t, how many tokens will be generated by that time? r into t, right that many packets will be transmitted due to the newly generated tokens.

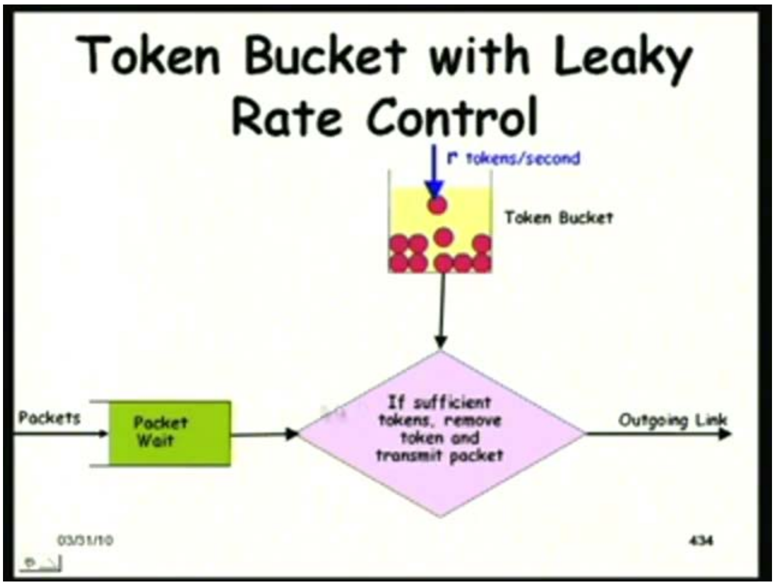
And the old tokens that were there in the bucket, the b tokens, that will be b right. So, it will be r into t plus b, does that appear alright?

(Audio Not Clear. Refer Time: 41:05 to 41:11)

No, no even b can go out. See, r t is the new token that has come in to the bucket, and b tokens were already there in the bucket. So, the b can go out, and the r into t can go out the packet, the bucket can become empty right ok.

So, this is important observation that due to the token pocket with leaky rate control, the average rate becomes r and the burst size becomes b and over any interval of time, the maximum number of packets that can get transmitted is r into t plus b. This we just discussed, we have effectively set a limit on the long term average and the burst size.

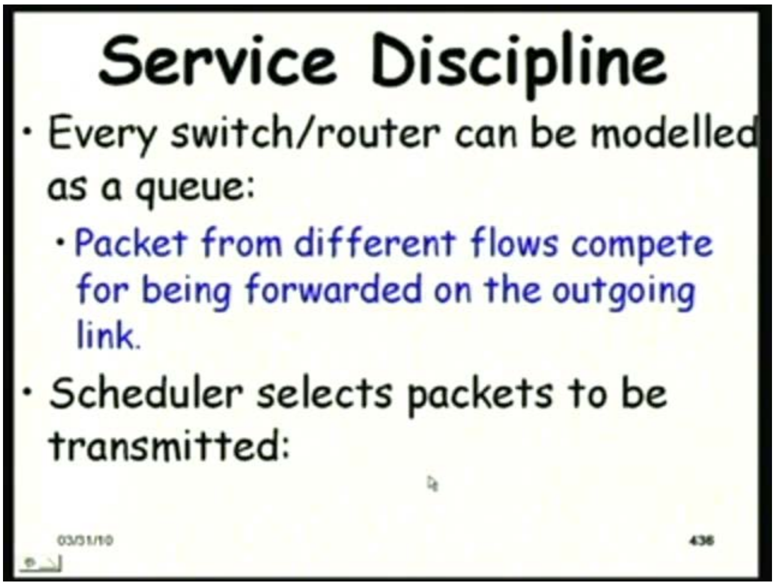
(Refer Slide Time: 42:05)



So, this is the schematic of the working. We have this token bucket, where r tokens are getting generated per second; just keep on generating periodically and if you look at one second r tokens would have got added here. If the bucket has less than b tokens, but if it is already full, then the tokens will be discarded; now as the packets come, they wait here; and if sufficient tokens are available, they are transmitted and the token is removed, but of course, if it was a policing that we are taking off, the traffic shaping by policing, then the packet will not wait; it will be discarded here.

So, this is how it will work. So, there are the switch or the router, here the output buffers are there and as the packets come in the rate control is applied, and then they are sent to the output buffers right, depending on the link on which there to be sent, by applying the rate control sent on to the output links, and then the service discipline here, just check this the red one is the service discipline that will check, how much resource has been reserved for that connection; and then it will transmit on the link for that buffer in the agreed rate or the reserved rate. So, it might be connected to a destination or it can be connected to another switch.

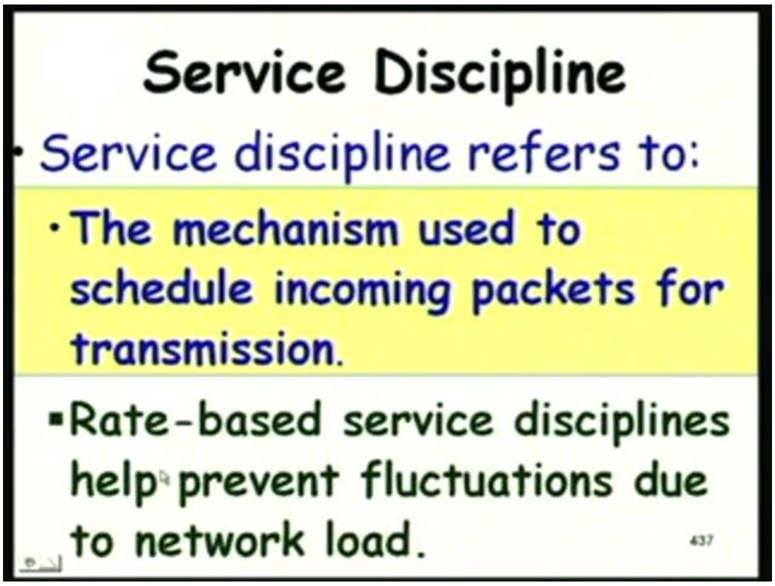
(Refer Slide Time: 44:08)



Now let us look at the service discipline; so far, we had looked at the rate control and also we have looked at the reservation, how reservation is made, resource reservation is made and also the rate control. Now, let us look at another important module in the quality of service framework, which is the service discipline. Now, to understand the service discipline, we can think of every switch or router as a queue, where the packets come and get queued. This is the output buffer, what we had shown in the previous diagram. So, here the rate control has been applied and the packets are queued on the output buffer. But for the outgoing link we need to remember that there can be different connections or different flows, packets form different flows or connection they might complete.

So, the task of the service discipline is to, for every flow for the reserved resource it tries to transmit on the outgoing link. It honors the reserved resource, even if there are too many packets for one flow are queued upon the output buffer, it does not transmit them unless it has sufficient reservation; otherwise it transmit at a slower rate. So, you can think of these as a scheduler, the service discipline is actually a scheduler, which looks at different flows, I mean the reservations made for different flows, and then checks the output buffer, and then puts it on to the output link, if sufficient resource reservation exists for that.

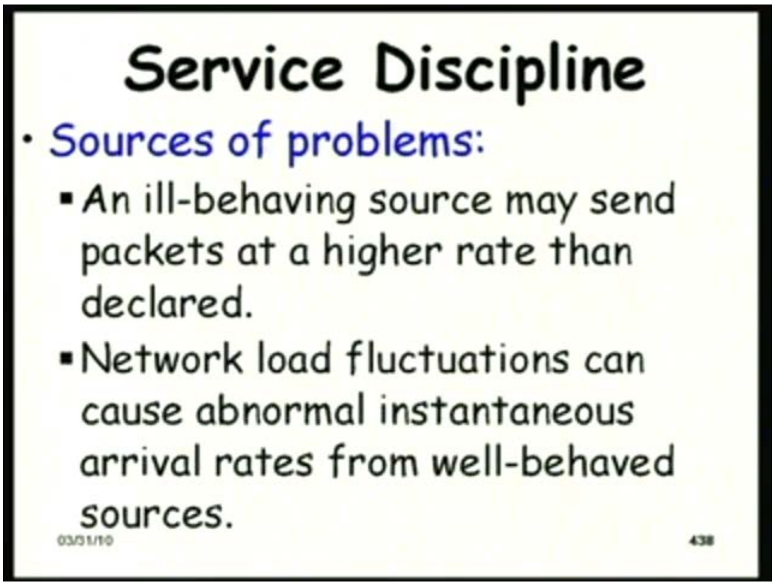
(Refer Slide Time: 46:06)



So, we can think conceptually that a service discipline is the mechanism used to schedule incoming packets for transmission. So, the packets that are incoming to a router, after the rate control is applied, they are put on the output buffer, and then they are taken up on transmission based on the service discipline. Now, we will discuss few service discipline in the subsequent slide, but one thing that we must remember is that all of them will be rate based service disciplines.

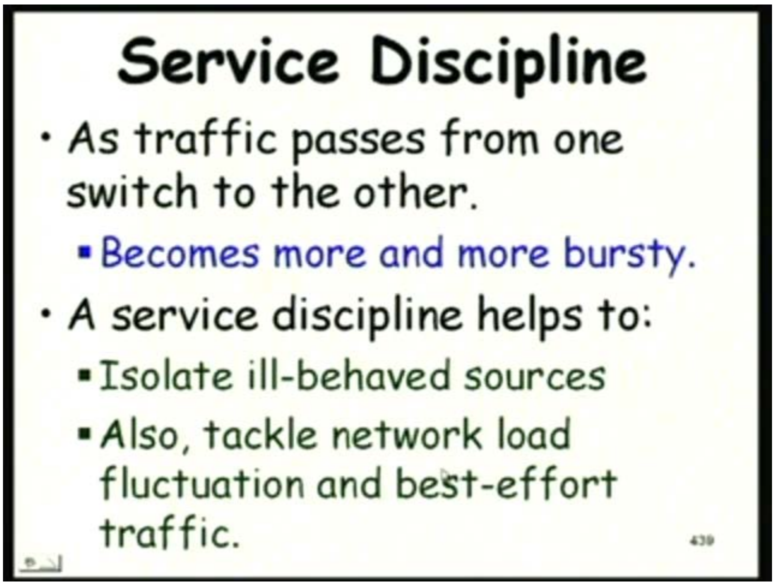
So, they will schedule the packets take the incoming packets from the output buffer and transmit them based on certain rates. So, we can think of as a simple model that for the different flows or connection specific rates are fixed, right they can be transmitted only up to that rate there are too many of them. They will keep on waiting on the output buffer.

(Refer Slide Time: 47:19)



Now, what is the complexity in the service discipline? One is that ill behaving sources may send packets at a higher rate than declared, but of course, the policing and shaping will take care of them, but still some packets I mean, they might… The maximum that was agreed up on can come on to the output buffer right, but this is if we have a rate control, proper rate control this is not a problem actually. The ill behaving sources can be handled. But the main problem here is the network load fluctuation can cause abnormal instantaneous arrival rates from even well behaved sources. So, we are just trying to explain this few minutes earlier that even when the all the sources are well behaved. Do not pump data at any higher rate than what was agreed, but still the traffic will loose itself.

(Refer Slide Time: 48:34)



Or in other words, any traffic as it passes from one switch to other becomes more and more bursty. So, traffic that was bursty only to a small extent, will increase its burst size; and finally, after few switches it will become really bursty; the burst size would have increased ten times.

Sir,

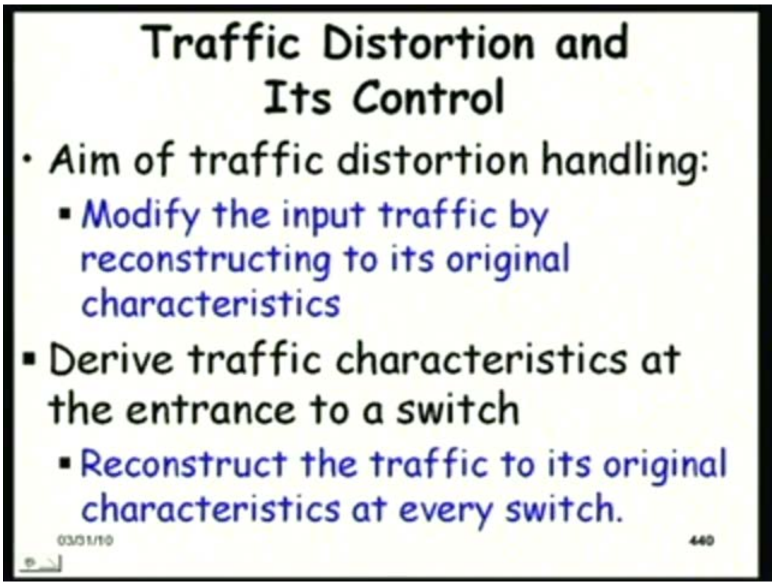
Yes.

We will design each maximum rate of each source such that the traffic the in extremely should not becomes more bursty. So, if all the process are behaving well, when is the traffic will burst.

Good question. See, he says that see the admission we rechecking that see they are injecting less than r packets per minute or something, maximum size are depending on some agreed traffic. So, why should that traffic become busty as it proceeds that is the question precisely. So, we will just discuss that with some mathematics

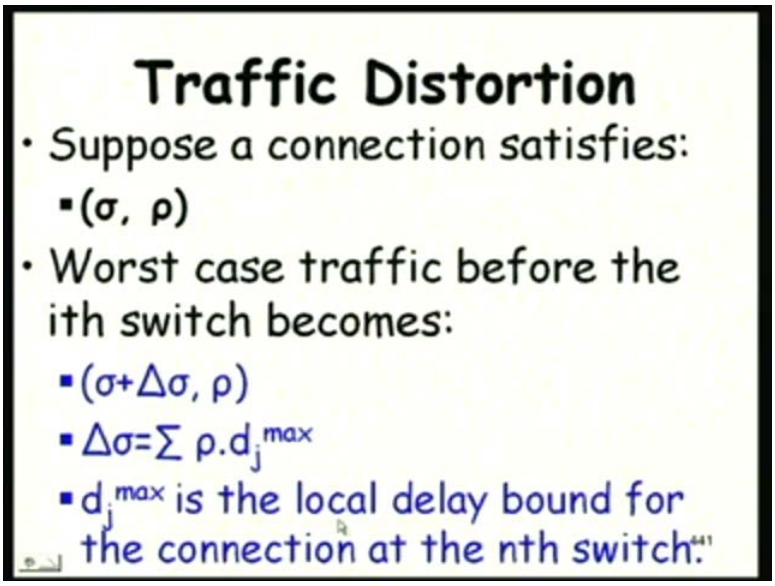
So, a service discipline can help to isolate ill behaved sources and tackle this is the most important point actually; the tackle the bursty traffic as if the traffic becomes distorted even form well behaved sources it tackles them. So, let us first understand the question that he was asking that why does the traffic get distorted, the perfectly correct traffic that was given by the sources; why should it become distorted as it proceeds on the network and how will it be controlled?

(Refer Slide Time: 50:14)



We will we will just discuss that I think in the next slide, but here just talking of the distortion handling, which we will try to reconstruct the traffic to original characteristics given that the traffic has the loose lost its shape; it has become distorted. The distortion handle mechanism will try to reconstruct the traffic. It will derive the traffic characteristics at the entrance of the switch, and reconstruct the traffic to the original characteristic that we are agreed upon.

(Refer Slide Time: 50:58)



Now, let us see why the traffic gets distorted. Now, let us say a connection satisfies sigma comma rho model. So, where rho is the average rate of transmission, and sigma is the burst size above above the average rate. This model we had seen right, rho is the average traffic and sigma is the burst size. Now, as the traffic proceeds though a different switches, just before the i th switch, it will become sigma plus delta sigma comma rho; the average becomes remains the same, because that is the rate at which the traffic is being pumped; rho is the average rate at it is the traffic is coming in, so that does not get change; but the bursts size increased the traffic has become bursty after every switch and the delta sigma is rho into d j max summation over all the switches. So, the delta sigma here is the average traffic into d j max for all the switches. What is d j max - is the delay bound for the connection at the n th switch.

So, let us see why this occurs? d j max to think of it is the maximum delay, that the packet can undergo at the switch right that is the agreed protocol. Now, what happens is that the packet once it comes gets queued and within d j max, it will be transmitted out right. So, the traffic that was coming in here got queued, and let us say this what burst has come and it has got queued, and then all of them are transmitted together, right; suddenly they were transmitted, they were all delayed till the burst came and they were transmitted.

So, we can see here that it has increased the burst size, because it had queued them; see these were coming one after other well separated out this was the average rate. But here it just queued them for d j max and then suddenly transmitted all the packets. So, the burst size has increased here. See here, just after one switch it will be sigma into d 1max where d 1 is the first router. So, it has delayed all of them that were coming nicely one after other; delayed them, and then transmitting in a burst the traffic has become bursty and after the n th switch you will find that it is the sum of all this.

So, the traffic becomes more and more bursty, as it crosses one switch after another. I mean, it may not be equal to this burst size, but this is the worst case situation, we are talking of that. In the worst-case, the maximum burst size can become this the actual burst size may be slightly less than this so we just saw how traffic gets distorted, a perfectly well behaved traffic gets distorted as it is transmitted on the network, and what is the amount of distortion that also we saw. I hope you all agree with this computation right you model this traffic like this, what will be the increase in the burst size, agree to that; and then we will see how the service discipline can help to handle the traffic distortion.

(Audio Not Clear. Refer Time: 52:02 to 52:04)

Exactly, the distortion is due to queue. So, let us stop now, and we will continue form this point. Thank you.