Broadband Networks

Prof. Karandikar

Electrical Engineering Department

Indian Institute of Technology, Bombay

Lecture - 28

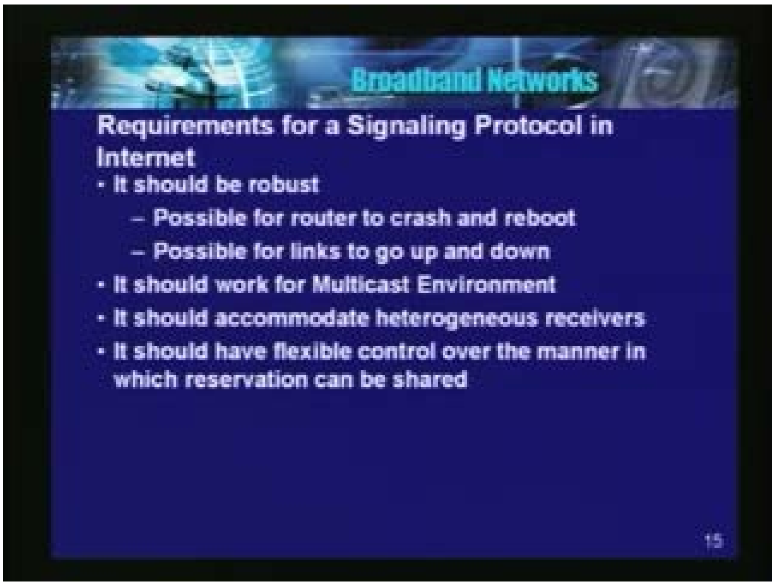
Differentiated Services Internet

So, in the integrated services internet model, as we have seen that resources are explicitly managed through a combination of admission control and packet scheduling algorithms; so what happens is that a source first specifies its traffic characteristics through T specs parameters and also its quality of service attributes through reservation specifications or what is called as R specs and these are specified through a signaling mechanism before a call is set up. And, if the resources, if enough resources are available in the network; the network will run an admission control algorithm and determine whether the resources can be allocated for this particular flow or not. If the resources can be allocated, then the admission control algorithm appropriately sets up the packet scheduling mechanisms in the routers and then the flow starts sending its packet.

So, what we will see today is that the signaling protocol which is used to set up the resource reservation in the Int Serv model which is popularly called as resource reservation protocol or RSVP. In fact RSVP was considered synonism with int Serv for a long period of time but we can see today that RSVP has also been extended for traffic engineering applications in the MPLS networks or for constrained shortest path based algorithm also.

So therefore, actually speaking, RSVP is a general signaling protocol where the source can specify its traffic characteristics and quality of service attributes. It is up to the network to determine whether the call can be accepted or not and this quality of service attributes can be either at the individual application flow level or this quality of service attributes can be at aggregate level also. So, that can be specified as part of the RSVP messages. So, today let us look at in detail, the features of this resource reservation protocol or RSVP.

(Refer Slide Time: 2:57)



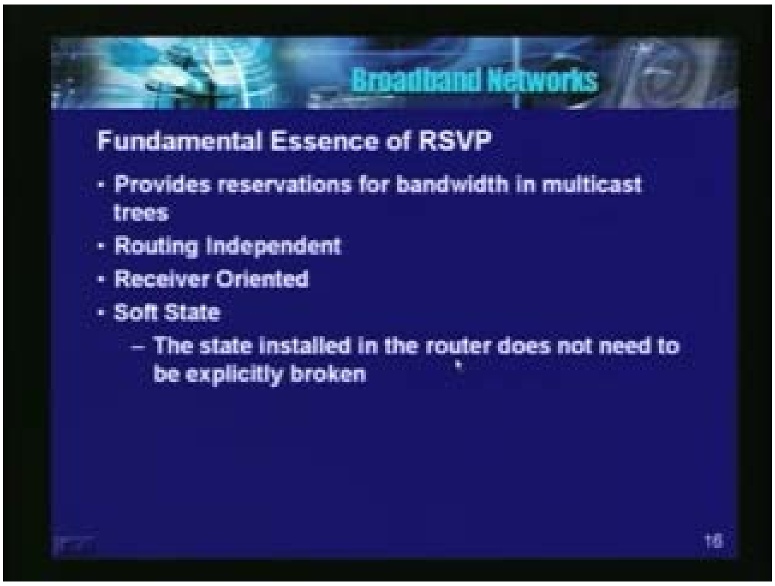
So, this brings us to an important question that what is the difference between a signaling protocol that is required in the internet as opposed to a resource reservation signaling protocol that is available in other QoS networks like a plain old circuit switch networks or a ATM based networks. So, the differences are that the signaling protocol in the internet, it should be a robust.

In the sense that links can come up and go down the routers or the nodes may crash and they may reboot but yet the signaling protocol should operate. Also, the protocol should work for multicast environment. This is very important because some of the earlier multimedia or real-time applications that were envisaged over the internet where essentially a multicast video conferencing or a multicast of a live meeting or let us say a movie or a cricket match was conceived over the internet. So therefore, it was very important that the signaling protocol must work in a multicast environment.

So, when the specifications for this particular signaling protocol were laid down, it was considered that it should scale to a multicast environment also. Then the signaling protocol should also work for heterogeneous receivers. The heterogeneity is a norm in the internet. Different receivers may have different link bandwidths and different capabilities in terms of the receivers or the routers; so therefore, it should be possible to reserve the resources as per the as per the receiver's requirements and not as per the sender's requirements.

So, this is the very fundamental difference. We will just come when we discuss the features of the RSVP; how signaling protocols RSVP is different from the standard signaling protocol let us say in the ATM based networks. Then the last requirement, it should have is that it should have a flexible control over the manner in which reservations can be shared and this we will see that this allows merging of the reservation requests and and so on particularly in the multicast environment.

(Refer Slide Time: 5:29)



So, when these requirements were laid down, a new signaling protocol was formulated and as I pointed out, this was called as RSVP - resource reservation protocol. The one important feature of this RSVP is that it provides reservations for bandwidth in multicast trees. So, it is specifically suited for the multicast environment, the signaling protocol is specifically meant for the multicast environment. Then the second important feature of the RSVP is that it is independent of the routing protocol. So, that is very important feature.

Now, what happens is that what RSVP does is it tries to reserve the resources along the route which has been already chosen by the IP routing protocol. So, the resource reservation protocol does not try to find alternate routes where the resources may be available. It tries to reserve the resources along the path which has been already chosen by an IP routing protocol. It may be possible that enough resources may not be available on the path chosen by the IP routing protocol but resources may be available on alternate path.

However, RSVP does not take that into account. It is essentially routing independent and the problem with the standard IP routing protocol is that they typically choose the path based on some simple metric like shortest path in terms of number of hops. So therefore, it is quite possible that a path which is shorter in terms of number of hops may not have enough resources in terms of bandwidth requirements. However, RSVP does not take that into account, it is essentially a routing independent.

Another important feature of that is RSVP is receiver oriented. Now typically, if you see the signaling protocol, they will be sender oriented. In the sense that the sender will specify its traffic characteristics and its reservation requirements or quality of service requirements and then the intermediate nodes will decide whether the enough resources are available or not.

Now, as opposed to that RSVP is receiver oriented. What does it mean? It means that since internet has heterogeneous receivers, different receivers will have different have different requirements in terms of bandwidth etcetera. So, the receivers will specify what are their qualities of service attributes or quality of service requirements and then the reservations will be done and not based upon what the sender wants to reserve the resources. And, the last important point which RSVP has is that it has a soft state. What does it mean the soft state?

The state, the reservation state which are installed in the router, they does not need to be explicitly broken. RSVP sends periodic refresh messages. Now, that is important because we want the signaling protocol to be robust with respect to the node failures or the link failures. So, if you recall that an IP routing protocol is really robust with respect to the node or the link failures. So, if a particular node or a link fails down, an IP routing protocol will automatically discover an alternate path. So, if the RSVP was designed to be a hard state protocol, then what would have happened is that if a particular link breaks down or so, then that particular QoS call would have been terminated.

However, in this case, what will happen is that if a link breaks down; since IP routing protocol has determined another alternate path in the meanwhile, what will happen is that when the RSVP sends refresh messages, the new reservations state will be installed on the alternate path if enough resources are available on the alternate path. So therefore, RSVP has a soft state as opposed to a hard state. It makes the protocol to be more robust. So, there are certain differences between the RSVP and QoS signaling protocols which are used in networks like ATM.

(Refer Slide Time: 9:55)

ATM has a signaling protocol which is derived from q dot 2931. So, the differences are that RSVP maintains a soft state and on the other hand ATM signaling protocol has a hard state. Another one is the RSVP is independent of route establishment, the route is established by standard IP routing protocol. On the other hand, in ATM, the resource reservation happens concurrent with the route establishment.

So, when you are setting up a virtual circuit path in the ATM networks, you also try to determine whether enough resources are available on that path or not and if enough resources are not available then the path will not be established. The third important point is that RSVP is typically a receiver oriented protocol, on the other hand the ATM is sender oriented.

So, in RSVP receiver specify their resource requirements or their quality of service requirements and then the reservation takes place. On the other hand, in ATM, the sender specifies its traffic characteristics and QoS requirements and then the reservation takes place. A consequence of that is this that in RSVP based signaling protocol it is possible to give different QoS to different receivers depending upon their capability requirements. In ATM, you give uniform QoS to all the receivers.

So, even if some receivers does not have a capability, then it may not make benefit of that quality of service attributes. As a matter of fact, it may not be possible to establish the reservation also in the network if the sender has the quality of service requirements, if the sender is sending the data to meet the quality of service requirements of the best receivers. So, it may not be even possible to establish a QoS call in the ATM based networks.

So, in that sense, the RSVP is more flexible and allows a lot of flexibility in terms of resource reservations to cater to different heterogeneous receivers in the internet. Now, how does a reservation is made in RSVP?

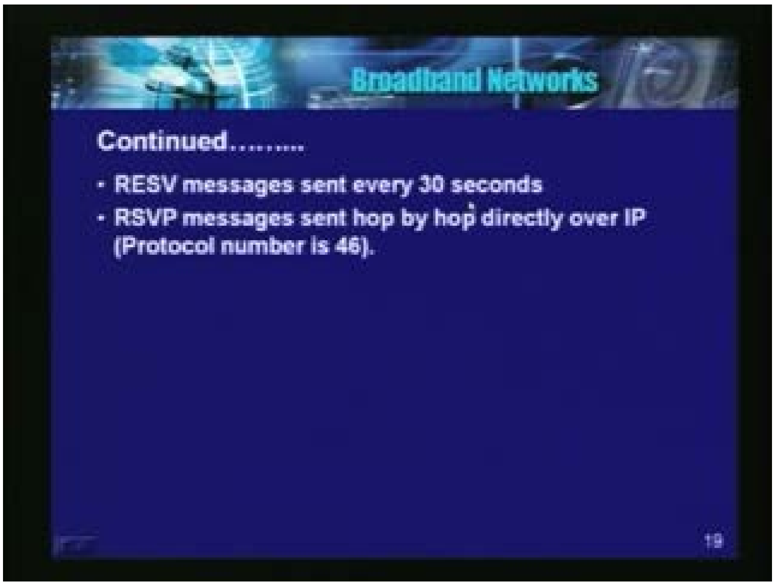
(Refer Slide Time: 12:26)

So, what happens is that a sender will send a path messages. Now, this path messages will typically contain the sender's traffic characteristics which are called as T specs. The T specs will contain traffic characteristics and these traffic characteristics will be specified in terms of token bucket parameter. The routers, they will look at the path messages and they will update the path state table.

This path state table will actually contain what was the previous hop from which these path messages were received because when the router in the reverse direction has to transmit the reservation messages, it will then transmit the reservation messages along the same path over which the path messages were received.

When the path messages, they reach the receiver; then the receiver will then send back the reservation messages and reservation messages will contain the sender's T specs and the receiver's R specs requirements. The routers, the intermediate routers will look at the reservation messages and then they will allocate the resources by using an admission control algorithm and if reservations cannot be done, then error messages will be generated.

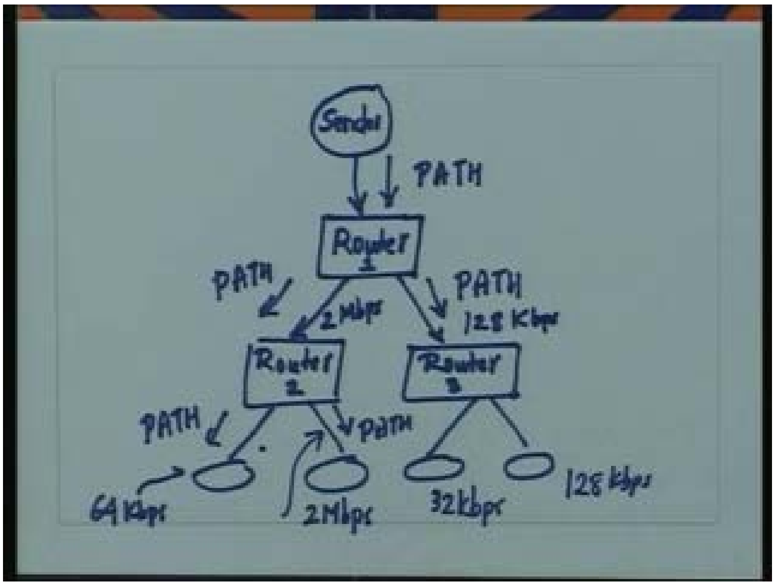
(Refer Slide Time: 13:40)



Reservation messages are sent every 30 seconds. These are like refresh kind of messages and RSVP messages are sent hop by hop directly over a raw IP and for that the protocol number that has been specified by the internet engineering task force IETF is 46.

So, let us look back again, how reservations are done. So, let me just explain here.

(Refer Slide Time: 14:09)



So, here is a sender and these are intermediate routers; so, router 1 and these are hosts. So, what happens is that the path messages travel along this. The path messages will go here, they will travel here, this router will forward the path message here and so on. Then the reservation messages will travel in the upward direction.

Now, let us say that this receiver has a requirement of 64 kbps, this receiver has a requirement of 2 mbps, this receiver has a requirement of 32 kilobit per second and this receiver has a requirement of 128 kilobit per second. So, this receiver in the reservation requests, say that's ask the router 2 to for reserving 64 kbps on this link and this ask the router to reserve 2 mbps on this link.

Now, when the router 2 asks router 1 to set up the link bandwidth here, then it merges these 2 requests and it will ask to set up 2 mbps on this link. Similarly, this will 128 kbps is requirement, so this 128 kbps will be set up here and finally on this link 2 megabits per second will be set up. So, as you can see here, the receivers have different requirements. This receiver has 32 kbps requirement, this receiver has 128 kbps requirement, this receiver has 64 kbps requirement, this receiver has 2 mbps requirement.

Now, you will reserve only 32kbps on this, 128 kbps on this, 64 kbps on this and 2 mbps on this. These reservation requests are merged and finally on this link you will reserve only 2 megabits per second. So, as you can see that the reservation requests can be merged also in the RSVP based signaling protocol and you can have reservation based upon the heterogeneous receiver’s requirements. If it was a sender oriented protocol, then what would have happened is that the 2 megabits per second would have reserved here, 2 mbps here, 2 mbps here.

Even though this router, these receivers cannot tolerate 2 mbps requirement, the 2 mbps message the 2 mbps bandwidth would have been reserved unnecessarily on this link leading to a low resource utilization or wastage of bandwidth. So, as we can see here that it is possible in RSVP by going for receiver oriented reservations to accommodate the heterogeneous receivers in a very scalable and flexible fashion.

So, when the receiver sends a reservation messages, they also send different reservation styles are possible for these merging. So, one merging example I will just illustrate but different merging styles are possible. So, when the reservation messages are sent; they not only send the flow specifications a flow spec or the reservation quality of service attributes, but they also send the manner in which these reservations request can be merged through what is called as filter style or reservation filter styles.

So, what are the reservation filter style options which are available in the RSVP? There are typically 3 reservation filter styles options are available: wild card filter style, fixed filter styles and shared explicit filter style. So, let us look at those filter style option.

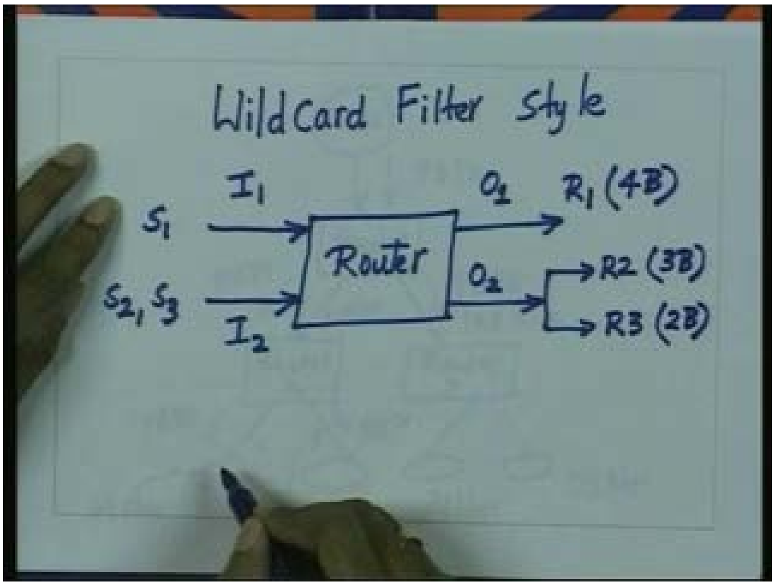
(Refer Slide Time: 18:28)

So, in the wildcard filter style, the receiver requests the reservation from all the flow from all the upstream sender. So, it is you receive the packets from all the upstream senders and the bandwidth reservation is shared. So, it is wildcard because you are receiving the request from the entire upstream sender. So, senders are essentially wildcards and the bandwidth reservation is shared among all the senders.

The second reservation style is a fixed filter style. So, in the fixed filter style, it is just opposite of the wildcard filter style. Here you specify the list of sender, you do not receive packets from all the upstream senders but you actually specify a list of senders and then the bandwidth reservations for each of the senders you also specify because these bandwidth reservations are distinct. So, this is what is called as fixed filter style.

Shared explicit style is like a is like in between the wildcard filter style and fixed filter style. In shared explicit style, you specify explicitly a list of senders. So, the list of senders is explicitly specified. But the bandwidth reservation is shared. This is what is called as shared explicit style. So, just to give you an example I will just show you how the wildcard filter styles works.

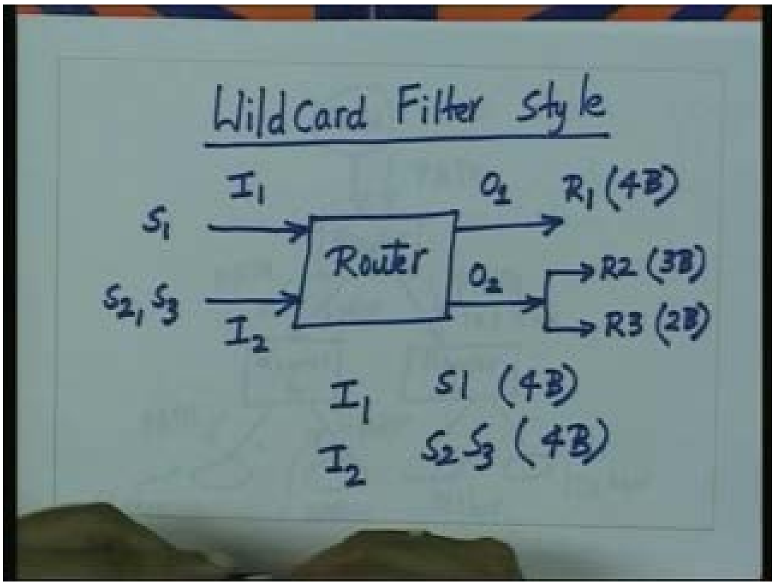
(Refer Slide Time: 20:00)



So, this is, we will see how a wildcard filter styles allows the reservation request. So, let us say here is a router, so here is an input interface I_1 and I_2 and here is an output interface O_1 and O_2. So, on I_1 the sender S_1 and sender S_2 and S_3 are connected to I_2 and on I_1 we have receiver R_1 and there are 2 receivers here - R_2 and R_3.

Now, receiver R_1 requires bandwidth of let us say 4B units, receiver R_2 requires a bandwidth of 3B units, receiver R_3 requires a bandwidth 2B units. So, what does it means is that it is a wildcard filter style, so receiver R_1 will receive request from all the flows - S_1, S_2 but the bandwidth will be shared and that bandwidth amount should be 4B units. Receiver R_2 will receive request from all the flows S_1, S_2, S_3 but the bandwidth will be 3B and similarly R_3 with the bandwidth will be 2B.

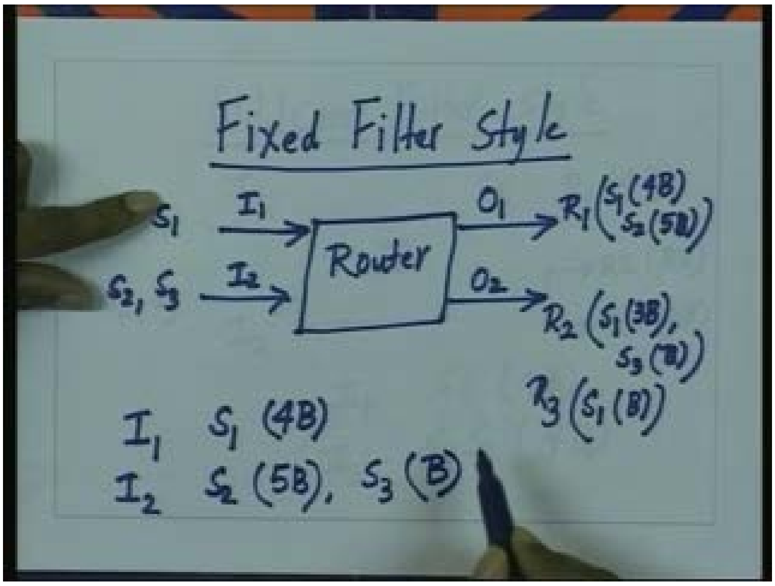
(Refer Slide Time: 21:36)



So, on this link, on I_1 if you see, we will reserve a request for S_1 to be 4B because 4B, this receiver wants to receive the flows from all the streams at 4B units. Similarly, for I_2 - S_2 S_3, we will reserve 4B units only because this receiver again wants to receive from S_2 S_3 for 4B units. Of course, this 4B unit on this link is shared but on this link, we will have to reserve 4B for S_2 and S_3 as well. So, this is what is called as wildcard filter style.

Now, let us look at fixed filter style. So, in fixed filter style what happens is that you specify a list of senders and also you specify the bandwidth reservation. So, the bandwidth reservations are distinct for each sender. So, let us look at fixed filter style.

(Refer Slide Time: 22:41)



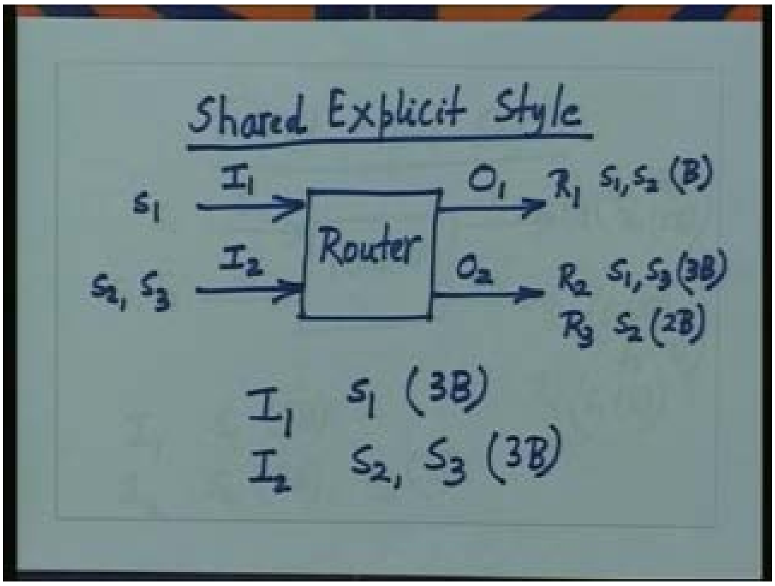
So, here is a fixed filter style. So, we have the same configuration, here is a router again, the input interface I_1 I_2, output interface O_1 O_2. So, here is a receiver R_1, so here of course receiver S_1 S_2 S_3. The receiver R_1 actually wants to receive data from S_1 at a bandwidth of 4B. So, it specifies this and from S_2 at a bandwidth of let us say 5B units and here R_2 wants to receive data from S_1 with a bandwidth of 3B units and it wants to receive data from S_3 with a bandwidth of B units and R_3 wants to receive data only from S_1 with a bandwidth of B units.

So, if you see here on this link; I_1 for S_1 so R_1 wants to receive from S_1 4B units, R_2 wants to receive from S_1 3B units and R_3 wants to receive from S_1 only B units. So, on this link for this S_1, we will reserve the resources for S_1 to be 4B. On this link I_2; now let us look at this, S_2 wants to receive the R_1 wants to receive from S_2 5B units and this bandwidth reservation has to be distinct. So, on this link you will definitely get S_2 5B, on this link for R_2 you will get… so, it does not want to receive anything from S_2. So, no reservation for S_2 is there but on this link now you want for S_2 5B units of bandwidth.

Let us look at S_3. For S_3, you want B units of bandwidth here. So obviously, you will reserve for S three B units of bandwidth over here. So, this is a fixed filter style where you specify a list of senders from which you want to receive the data and the bandwidth reservations that you want to have and the bandwidth reservation are distinct.

Now, the third style is the shared explicit. In shared explicit, you specify explicitly the list of senders but the bandwidth reservations are to be shared just like wildcard filter style. So, bandwidth reservations are shared but you will specify the list of senders. So, here is a shared explicit style.

(Refer Slide Time: 25:40)



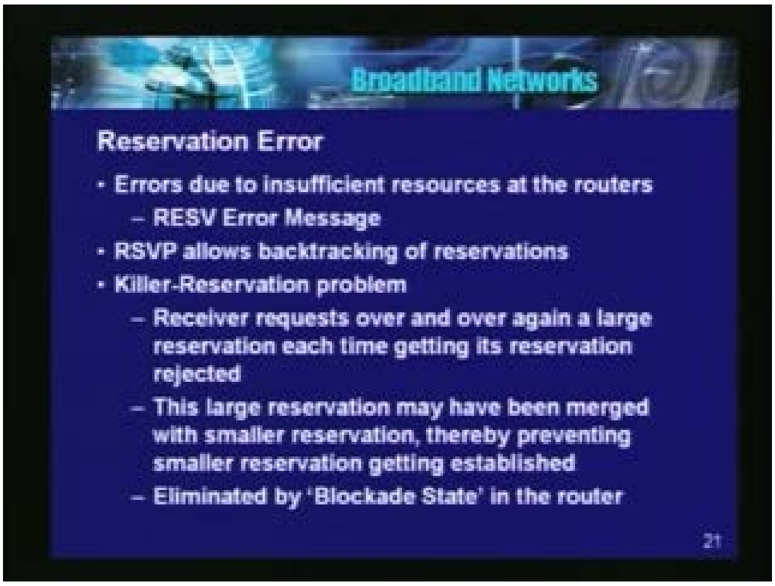
So, again here is the input interface I_1 and input interface I_2. Now, so here are the sources - S_1 S_2 S_3 and here, R_1 now wants to receive from S_1 and S_2 only, not from S_3 and the bandwidth reservation will be shared in this, shared bandwidth is B. R_2 wants to receive from S_1 and S_3 and again now the bandwidth reservation will be shared because it is shared explicit style. So, this bandwidth reservation has to be 3B and R_3wants to receive only from S_2 the bandwidth of 2B.

So, as you can see on the link I_1; how much we will reserve? R_1 wants to receive from S_1but with the bandwidth of B and R_2 wants to receive from S_1 but with the bandwidth 3B. So, on this link S_1 must reserve 3B units of bandwidth. On this link, now the bandwidth reservations are shared, so shared explicit style. So, no explicit reservations need to be made for S_2 and S_3.

So, S_2 and S_3 can share the bandwidth and whichever is the maximum of this and you can see that S_3 actually is having 3B here, S_2 2B and S_2 S_2 B. So, we can actually reserve 3B units of bandwidth which can be shared.

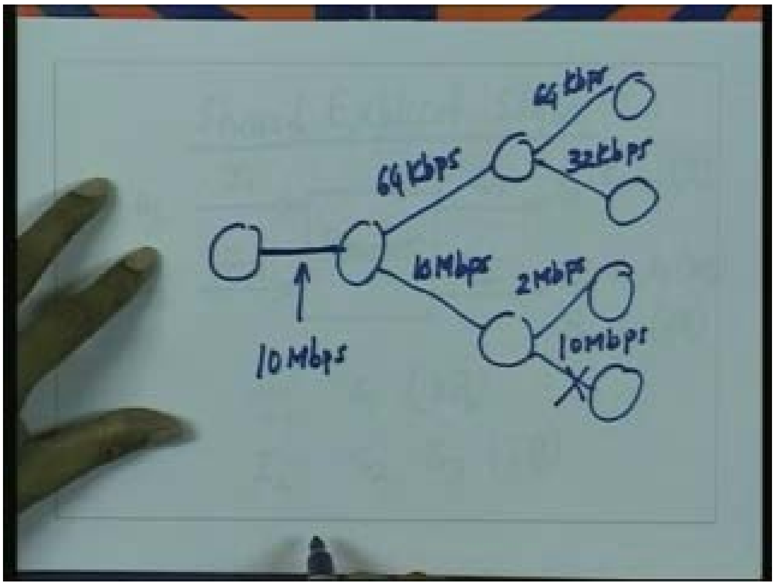
So, this is how a shared explicit style will work. In shared explicit style, you will specify a list of senders and the bandwidth reservations will be distinct. Now, let us look at what happens if there are reservation errors are there.

(Refer Slide Time: 28:09)



So, if you try to send path messages and the receivers try to send reservation messages and if enough resources are not available in the nodes, then there can be an error. So, errors due to the insufficient resources at the routers: so these are called reservation error messages. Actually, RSVP allows backtracking on the reservations also.

(Refer Slide Time: 28:42)

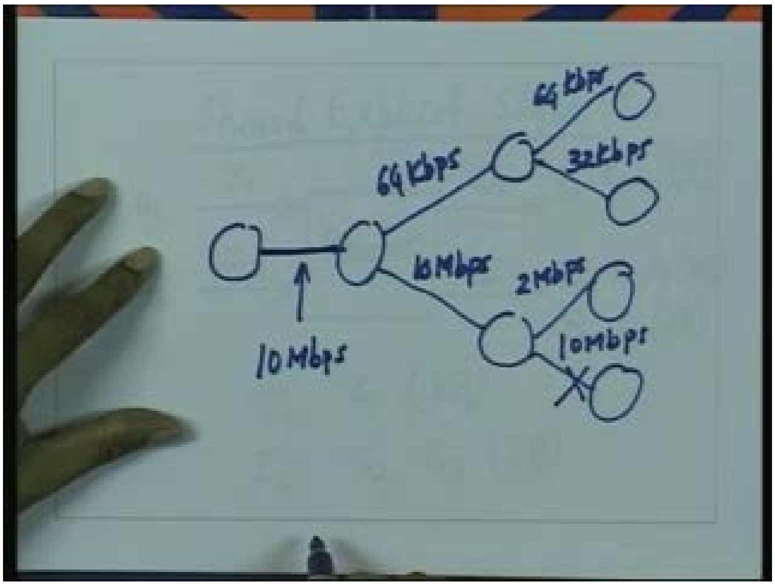


Now, backtracking of the reservation is something like this that if you try to reserve, so let us say here is a node and so here you have 64kbps and this resource requires 32kbps, this requires 2 mbps and this requires 10 mbps. So, these 2 reservation requests can be merged and you will reserve only 64 kb, try to reserve only 64kbps over this link and here you will try to reserve 10 megabits per second and then this will be merged and here you will try to reserve 10 mbps here.

But it may be possible that 10 mbps may not be available here, so you can actually backtrack the reservations and then you may want to remove this node or you may not want to reserve the resources for this node and then allow 2 mbps here and then 2 mbps here; that may be sort of done.

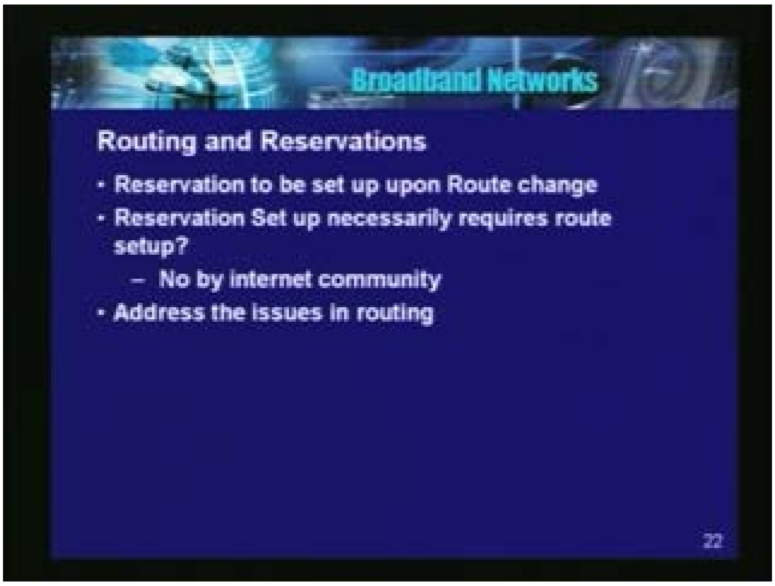
Otherwise, what will happen is that it will lead to what is called as killer reservation problem where what will happen is that the receiver requests over and over again a large reservation and each time getting its reservation request rejected and this large reservations may have been merged with smaller reservations and thereby preventing the smaller reservations also getting established. So, this is eliminated by blockade state in the router.

(Refer Slide Time: 30:11)



As you can see here, in our example, because the 10 mbps bandwidth is not available here and since these have been merged here and only one receiver was there requiring 10 megabits per second; the reservation request actually may get blocked here and blocking therefore, the reception for other receivers as well. So, this may be eliminated by using a blockade state in the routers which will disallow the reservation only for these routers. So, this is possible to do within the RSVP. What about the routing and reservations?

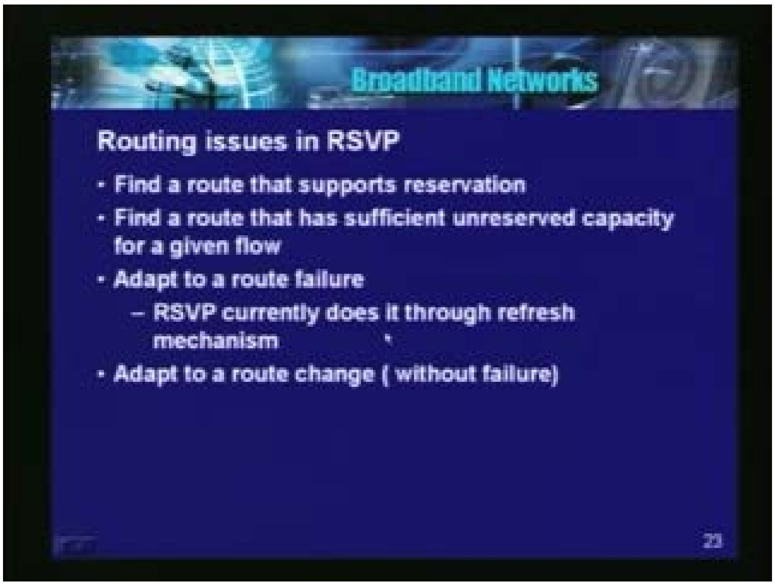
(Refer Slide Time: 30:40)



Reservations, they need to be set up on the route change, as we have already pointed out that resource reservation in the RSVP is independent of the routing protocol. The route will be established by standard IP routing protocols and reservation set up necessarily requires the root setups. So, it has been as I said, no by the internet community and RSVP has not been linked with the root reservations.

So, all the issues are addressed in the routing itself if there is a node or a link failures and the RSVP takes care of that by using refresh messages which are sent every 30 seconds or so.

(Refer Slide Time: 31:21)



So, you find a route that supports reservation or you find a route that has sufficient unreserved capacity for a given flow or you adapt to a route failure. So, currently as we have just pointed out RSVP is doing it through the refresh mechanisms and adapts to the route changes. So, this actually concludes our discussion on the resource reservation protocol RSVP. Some of the important features of RSVP we have pointed out.

In fact, it was some people thought that RSVP has scalability. So, RSVP has also been applied to the traffic engineering for the MPLS networks. So, that what is called as RSVP TE. New objects have been defined in the RSVP protocol. Some people had argued that RSVP has a scalability problem because RSVP was considered similar to the Int Serv but as we have just seen that RSVP is just a simple plain signaling protocol which can be used within the frame work of the integrated services internet to reserve the resources. So therefore, all the scalability problem that we have actually debated in the internet community, were associated more with the Int Serv framework rather than the RSVP.

RSVP can be used to reserve the resources either in the integrated services internet or even in the differentiated services framework or even for load balancing or traffic engineering in the MPLS based networks. Its fundamental features are that it is routing independent, it is based on the soft state protocol and it is receiver oriented and it caters to the multicast. So, with these 4fundamental features which are so unique to the internet problem which distinguishes signaling for resource reservation in the internet from signaling for resource reservation in another network let us say, a virtual circuit based ATM networks.

So, these features actually distinguish the requirements of a signaling protocol which has to be there in an internet. So, now as we have already seen both the integrated services framework and the signaling protocols for that RSVP; let us now see why the Int Serv model was not popular in the internet community and what are the real problems that are associated with this and how the internet community is coping up with those challenges.

(Refer Slide Time: 33:50)

So, with that evolved what we call as the Diff Serv or the differentiated services framework. So, we will discuss today some of the features of the differentiated services framework.

(Refer Slide Time: 34:00)

Now, what were the problems with the integrated services model and why the Int Serv model was not popular in the internet? Let us look at those problems first. So, one problem was and one of the most important problems was the scalability issues in the internet. Now, Int Serv is basically required to maintain the per-flow state for each application flow and if you today look at the core routers, they will approximately see something like 250 k to 1 and in fact sometimes one million flows also in the core routers today, today that number is actually one million.

So, that means if we have to maintain the reservation states for one million flows, we have to maintain large number of reservation state in the core routers and that is clearly a sort of impossible. So, this was one scalability problem associated with the Int Serv networks that did not make the Int Serv frameworks so popular or successful.

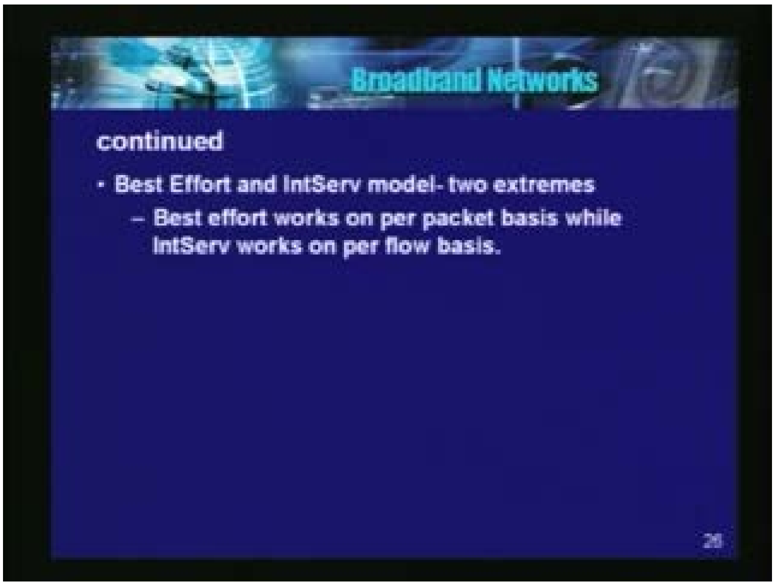
Another problem with the Int Serv framework was that it relied on the end to end service paradigm. So, the Int Serv model was working on the fact that the source will initiate the resource reservation, reservation states will be installed in all the nodes which come along the path and the quality of service will be available on the end to end basis. Now, clearly this required that all the nodes on the path must support the reserve reservation mechanisms; otherwise, the end to end service model cannot work and clearly in internet which has exploded or which has grown to such a large extent, the QoS paradigm can only be implemented in a phase wise or in a stage wise manner and as a result it is impossible from day one to give end to end QoS service.

So therefore, Int Serv model could not cope up with the evolutionary frame work of quality of service guarantees. So, that was one of another reasons why the Int Serv model did not succeed apart from, with related to the scalability issues and also it required all the intermediate nodes must support because the paradigm was based on end to end service model.

Then, another flexibility was that the Int Serv provides a small number of service model and those service models are end to end. We actually need some kind of qualitative services and third one was that the Int Serv model actually relied on the signaling protocol, hosts have to undergo a signaling mechanisms in the Int Serv. So, they have to first specify their traffic characteristics and quality of service guarantee and so on and some kind of a dynamic signaling procedure is required in the internet.

So, these were the fundamental challenges which were there, which became the bottleneck in the implementation of the Int Serv architecture in the internet framework. So, to address these challenges a new paradigm that was established and that was called as the differentiated services framework.

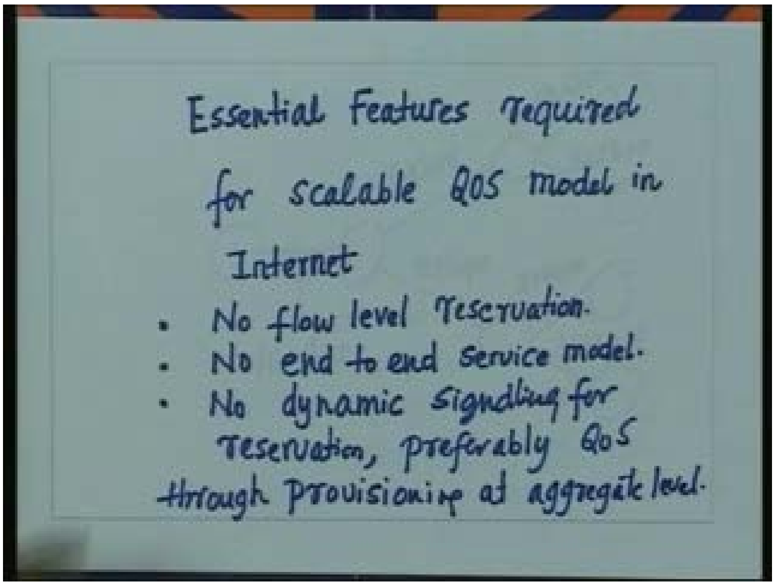
(Refer Slide Time: 37:21)



If you really see, before actually we discussed how we come up with a differentiated services framework; best effort internet is one extreme which does not give any quality of service guarantees which does packet forwarding by looking at the destination IP address. So, that is sort of one extreme. When Int Serv was proposed that went to the other extreme that tried to give resource end to end resource reservation on a per micro flow basis; so that was another extreme.

So, what is required is to come up with a framework or a mechanism which has certain distinguishing features, which are required to provide quality of service guarantees in the internet. So, what are those features? Let us look at some of the essential features which are required to give a scalable QoS model in the network.

(Refer Slide Time: 38:17)



So, what are those essential features required for scalable QoS model in internet? First thing is that no flow, no flow level reservation. You must reserve the resources at the aggregate level; it should not be on a per-flow level.

Second; no end to end model, it is important because what is really happening is that when we are gradually introducing the quality of service framework in the internet in an evolutionary manner, we cannot from day one start to give quality of service guarantees on an end to end basis. So, it must gradually evolve, the framework must allow for that.

Third thing is that no dynamic, it should not be dependent on any dynamic signaling. It should not be dependent on any dynamic signaling like RSVP in the Int Serv where you have to first undergo a signaling procedure … based. It should be more dependent upon some kind of configuration or management or provisioning kind of thing.

So therefore, it will be preferable that no dynamic signaling for reservation; preferably, QoS through provisioning at aggregate level. So, with these characteristics actually, the differentiated services or the Diff Serv paradigm was developed. So, the thirst of the Diff Serv paradigm is that first of all you will aggregate the flows, no micro flows.

The aggregation of flows is possible in the Int Serv also because by you can change the definition of flow. Int Serv allows you to change the definition of the flow, the flow need not be an application flow in the Int Serv. It can be all the packets coming from the particular entity or a source IP address also. So, that is possible.

Most important thing however in the differentiated services frameworks is that you are provisioning the resources for a particular aggregate of flow through SLA conformance management configuration and provisioning mechanisms rather than by dynamic signaling mechanism which was there in the Int Serv. That is one thing.

Second most important thing in the differentiated services frameworks is that while in the integrated services, you are setting up an end to end service; in the differentiated services

framework, we are not defining any end to end service. We will define what is called as forwarding treatment given to the particular aggregates packet by a specific node.
So, we will define only the forwarding treatment and what forwarding treatment should be given to the aggregates packet will be determined by information contained in the packets header only. It will not be through a flow state table. So, no state will be maintained in the core routers, it should be a stateless mechanism, no state should be maintained.

So, as a result, the whole paradigm becomes scalable. It can be gradually implemented in an evolutionary manner and thirdly by combination of these forwarding treatments, a service provider can offer and build various qualitative services that he can give it to different customers. So, that way a differentiated services paradigm was actually evolved.

(Refer Slide Time: 43:04)

So, what are the requirements of therefore differentiated services architecture is that it needs to accommodate a very wide variety of services and provisioning policies. It needs to decouple the traffic conditioning and service provisioning from forwarding behavior. So, forwarding we will define the forwarding behavior of each individual node by some rules and those rules will be typically carried in the packet in the form of some code point.

No end to end service will be defined and no per flow or per customer or per aggregate state will be maintained in the core router. Core routers will only look at the index or label in the packet and do a very simple packet classifications. The resources will not be reserved by any hop by hop signaling or so. In fact, there will not be any concept of resource reservation. There will be a concept of only resource provisioning.

(Refer Slide Time: 44:04)

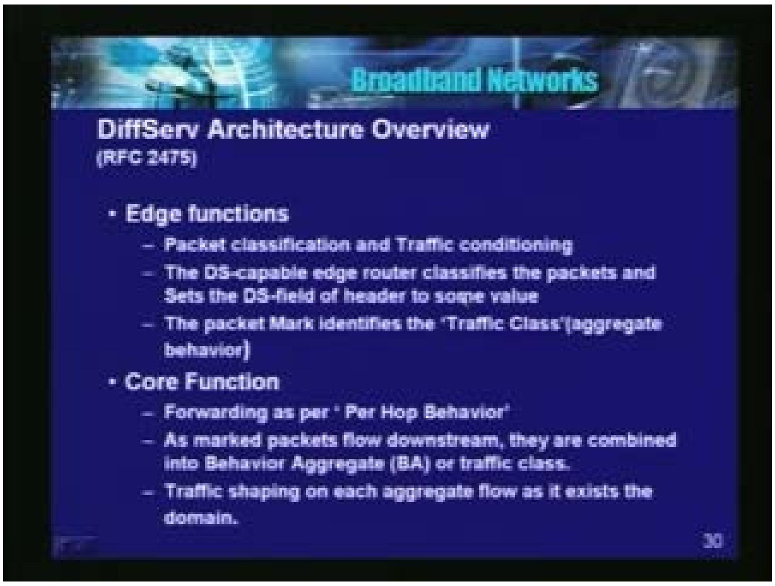
So, the basic principle is that the resource allocation will be done on aggregated traffic. Forwarding behaviors are defined in the differentiated services rather than services. Now, what is forwarding behavior? There is a certain difference between the forwarding behavior and service. Forwarding behavior or forwarding treatment, it typically refers to an externally observable behavior of a particular mechanism or an algorithm which has been implemented in the routers. On the other hand, the service refers to the overall performance that a customer traffic will receive.

Now, just to give you an example; for example, we may implement a forwarding behavior which is called priority in the router. So, we can implement a high priority forwarding treatment. So, packets which belong which are supposed to be receiving high priority forwarding treatment will be put in the high priority queue and they will be served. But that does not necessarily mean that the aggregate traffic which has subscribed to this high priority treatment will necessarily receive good service.

It may so happen that the load of the aggregate traffic on the high priority traffic queue may be so high where actually the customers in the low priority traffic may be so low that customers in the low priority traffic may be actually receiving better service. So as a result, in order that the high priority traffic actually perceives better service, you also require a combination of admission control to control the amount of high priority traffic that can be injected into the network.

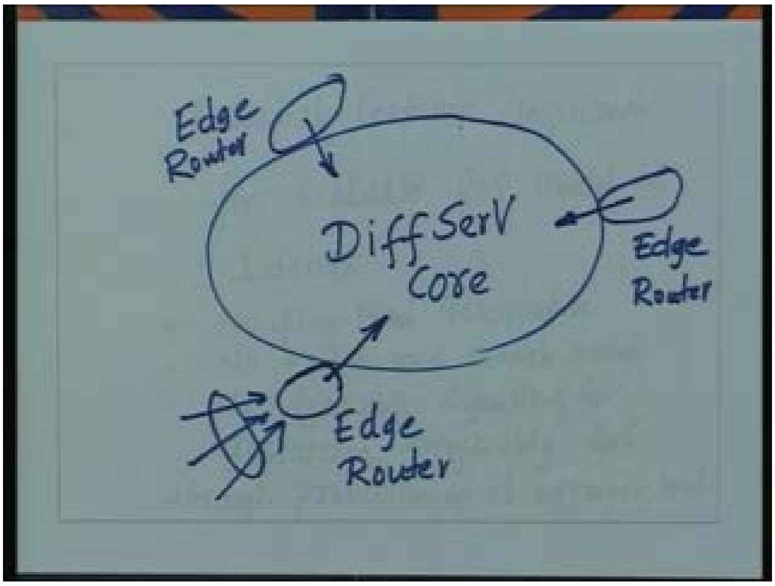
So, a combination of admission control and this forwarding treatment will give you a better service to high priority traffic. So, this is how a difference between forwarding treatment or a service that a particular traffic aggregate will receive. So, the service actually refers to overall performance that a customer traffic… So, a service provider may actually build different services by different forwarding treatments and as we have just said that the QoS is done by provisioning rather than by reservation and this QoS provisioning is done through SLA contracts rather than through the dynamic signaling mechanisms.

(Refer Slide Time: 46:36)



So, the Diff Serv architecture was actually propounded in the IETF's RFC 2475. It discusses in great detail how the architectural framework of a differentiated services… So, now to incorporate these paradigm principles that is we want a simple packet classification, no state to be maintained in the core, the core routers will give forwarding treatment by looking at the packet label only, no dynamic signaling etcetera; so to take care of this fact, the architecture defines actually the network into 2 parts - one is the edge routers and another one is the core routers. So, the differentiated services network will look something like this.

(Refer Slide Time: 47:22)



So, here is the here is the core of the Diff Serv networks. So, here is the Diff Serv core and here is the these are the edge routers, so these are edge routers. Now, what happens at the edges is that edge routers are receiving several traffic flows, edge routers will aggregate these traffic flows into one traffic aggregates.

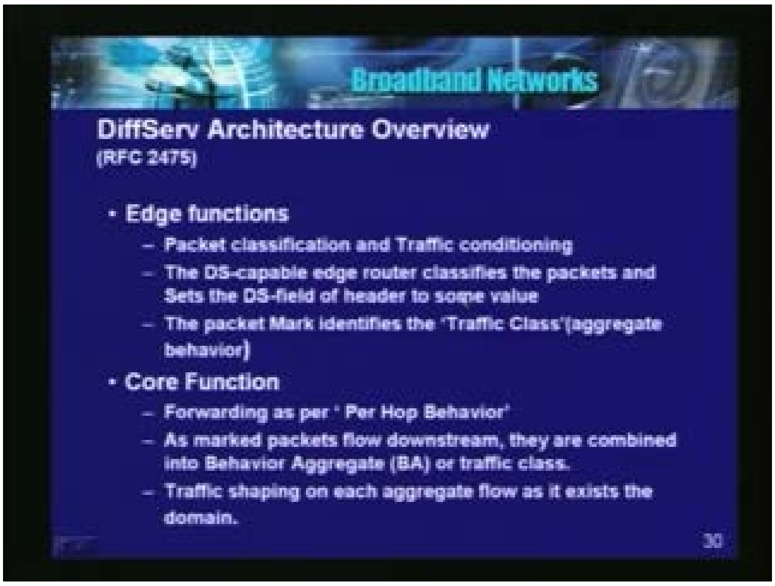
So similarly, they will inject them into the core. They will mark these packets with specific labels and the packets will be marked with specific labels depending upon the forwarding treatment they should receive in the core routers. So, all the packet classification is done at the edge routers, the edge routers will look at the packet attributes and depending upon the rules; we will classify them into forwarding classes.

So, now we have forwarding classes and the number of these forwarding classes is typically limited. It is not as high as number of flows which were there in the Int Serv routers. After we have classified these packets into forwarding classes, each packet is tagged with some kind of label or some kind of index which will indicate that it belongs to which forwarding class and then in the core router, in the core, the router will only look at that index and will apply the corresponding forwarding treatment. So, that is the basic principle of the differentiated services.

Now, the particular index which has been chosen by the differentiated services framework is the IETF architecture has actually made use of IP TOS field, IP TOS field which was actually not used in the earlier internet, only in some application it was used to give the precedence. So, those IP TOS field type of service field which were actually, originally meant to give some kind of priority to the IP traffic.

However, it was never commercially exploited. So, the differentiated services architecture has redefined those field to indicate to which one of the forwarding classes this particular packet would belong to. Now, if you look at now architecture overview; as we have said that there are 2 functions, edge functions, the edge routers will perform the packet classifications and the traffic conditioning.

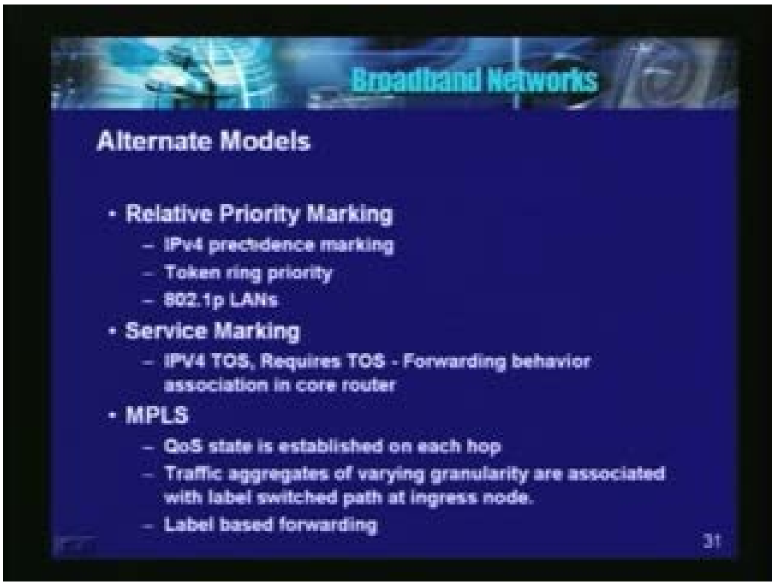
(Refer Slide Time: 50:13)



So, it will classify the packets into forwarding classes. So, it classifies the packet and it sets the DS field or the differentiated services field of the packet header to some value which indicates to which packet it belongs. So, this DS field identifies the traffic class and each traffic class has a specific forwarding treatment associated with it which is also called as aggregate behavior.

Now, as far as the core routers are concerned, they will do the forwarding as per the per hop behavior which is indicated by this differentiated services or DS field and as marked packets they flow downstream, they are combined into behavior aggregates or traffic class and traffic shaping is done on each aggregate flow as it exists the differentiated services domain. So, these are the functions of edge routers and these are the functions of the core routers.

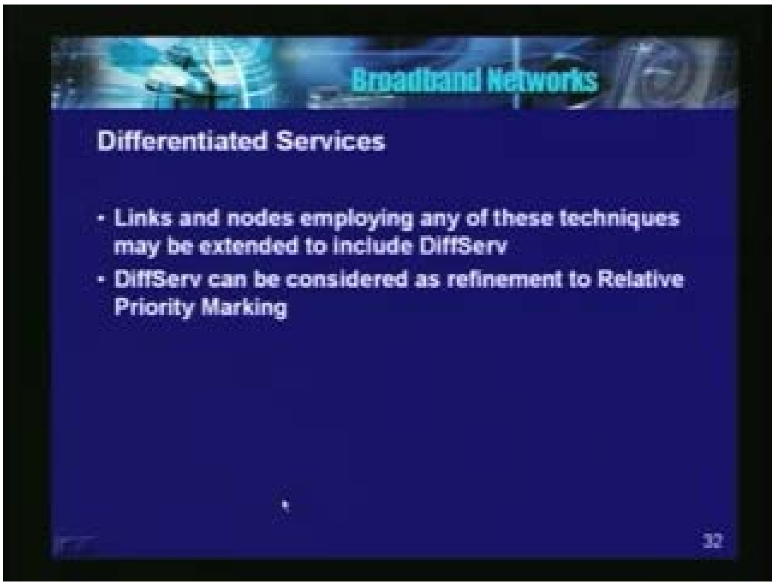
(Refer Slide Time: 51:22)



If you see, how differentiated services model differs from the alternate models is; so basic model essentially is that the packets contain, the packets header actually indicate that what kind of forwarding treatment the routers or the node should give. So, the packet indicates. If you recall that similar models exist in other areas also, in other context also which is as we are just speaking about that the IPV4 has this specific marking. Even in Ethernet based LANs we have a 802.1p priority queues.

So, in MPLS for example, the QoS state is established on each hop, the label can or the exp 3bits can indicate and when you do the label based forwarding also you can indicate the quality of service the quality of service guarantees.

(Refer Slide Time: 52:21)



However, how differentiated services differs from all these things is that in differentiated services, you do the you assign these differentiated services this index at the edge function and after you have classified the packets into these traffic aggregates; in the differentiated services domain, you are provisioning the resources for this particular traffic aggregates. So, what you are doing is that while it is true that forwarding treatments nodes will do, on top of that through the SLA contracts, you are giving some kind of some kind of service which will be built over the top of this forwarding treatment.

So, in that respect it is different from simple priority treatment of 802.1p Ethernet priority queuing. So, there is an concept of there is a concept of an SLA provisioning on the aggregate basis within the differentiated services domain. So, that is how it distinguishes from simple priority queuing of the 802.1p where that concept does not exist. It is only forwarding, it is only a treatment or a priority treatment which will be given to the queue; there is no admission control mechanism, there are no SLA conformances or provisioning.

So therefore, it may be difficult to build a service over such priority queuing mechanisms, simple priority queuing mechanisms. So, differentiated services framework actually allows to build you a service by having a good combination of forwarding per hop behavior as well as by through some admission control mechanism and SLA contracts.