e-PG Pathshala

Subject : Computer Science Paper: Computer Networks

Module: QoS – IntServ & RSVP Module No: CS/CN/27

Quadrant 1 – e-text

We know that there are many techniques used to provide QoS – call admission, signalling, QoS-aware routing, classification, traffic shaping & policing and scheduling. We will now see how these techniques are actually implemented in the protocols that have been proposed for providing QoS. In this module, we will go into the details of the IntServ and RSVP protocols.

**Learning Objectives:**

- What are the possible approaches to provide QoS?
- How does the IntServ-RSVP combination provide QoS?
- How successful has this approach been?

## 27.1 QoS approaches

In the Internet, two different approaches have been proposed for providing QoS: Fine-grained approach, and coarse-grained approach. *Fine-grained approach* provides QoS to individual applications or flows, while the coarse-grained approach provides QoS to large classes of data or aggregated traffic. Integrated services mechanism (IntServ), along with Reservation protocol (RSVP), belongs to the former category, while Differentiated Services (DiffServ) belongs to the latter category. We focus on the IntServ mechanism in this module.

## 27.2 IntServ - Overview

IntServ refers to a body of work that was produced by the IETF around 1995–97. It provides specifications for a number of *service classes designed to meet the needs of different* application types. It also defines how RSVP could be used to make reservations using these service classes. With the help of RSVP, it supports dynamic reservation of resources.

In IntServ, a " flow " is the basic management unit. Thus it aims to provide accurate quality control. All the QoS related control functions are provided in each router. That is,

all routers in the network must be IntServ aware.

IntServ can be said to comprise of four components: Service models, Service Interface, Packet Scheduling, Signaling to/from network. These four components answer the following questions.

What does the network promise?

- Answered by service models supported.

How does the application describe what it wants?

- Answered by the service interface.

How does the network meet promises?

- Answered by packet scheduling mechanism.

How is the promise communicated to/from the network?

How is admission of new applications controlled?

- Answered by the signaling mechanism.

### *27.2.1 Service models*

IntServ primarily supports two Service models for QoS: Guaranteed service (RFC 2212), and Controlled Load service (RFC 2211). The Guaranteed Service (GS) model targets hard real-time applications. It seeks to guarantee bandwidth and end-to-end delay bounds for flows. In GS, the user specifies traffic characteristics and a service requirement. The network should guarantee that the delay experienced is less than a specified maximum value. This requires admission control at each of the routers, to be able to guarantee.

The Controlled load model emulates a lightly loaded network even though the network as a whole may in fact be heavily loaded. It targets applications that can adapt to network conditions within a certain performance window. In CL service, the user specifies traffic characteristics and bandwidth. This also requires admission control at each of the routers.

### *27.2.2 Service Interface*

A session must first declare its QoS requirement and characterize the traffic it will send through the network. This is done by specifying what is called as "*flowSpec". FlowSpec* has two components, T-spec, and R-spec.

T-spec: defines the traffic characteristics of sender (typically, as a leaky bucket with rate *r* and buffer size *b*).

R-spec: defines the QoS being requested by receiver (typically, a rate *r*).

A signaling protocol is needed to carry the R-spec and T-spec to the routers where reservation is required. RSVP is the signaling protocol used for this purpose.

### *27.2.3 Packet scheduling*

For guaranteed service, the traffic is characterized by a token bucket filter with a rate r and bucket depth b. Weighted Fair Queuing (WFQ) is used at the routers to control the bandwidth allocation, and manage the delay bounds.

### *27.2.4 Call Admission*

Routers will admit calls based on their R-spec & T-spec and on the current resource allocated at the routers to other calls as depicted in Fig. 27.1.

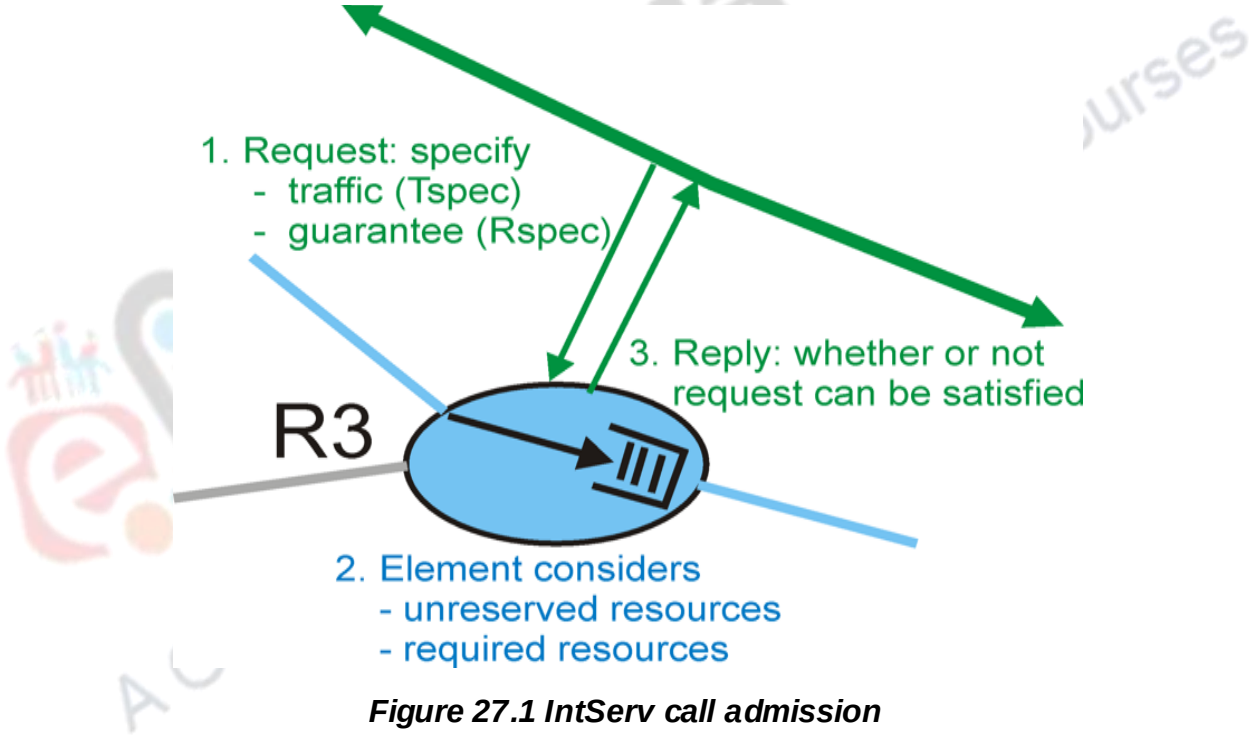



***Figure 27.1 IntServ call admission***

Effectively, there are two key components that comprise IntServ – flowSpec, and RSVP. FlowSpec defines the information that we provide to the network, and RSVP is a mechanism to exchange such information, along with requests for service, and admission control decisions. We look at these below.

## 27.3 Flowspec

As mentioned above, there are two separable parts to the flowspec:

Tspec, which describes the flow's traffic characteristics; and

*RSpec,* which describes the service requested from the network. Rspec is service

specific and easy to describe. In case of controlled load service, no parameters are given. In case of guaranteed service, a delay target or bound is specified.

Tspec, on the other hand, is tricky to specify, as it is often not a constant. For most applications, the bandwidth is not a single number. It varies constantly. For instance, a video application will generate more bits per second when the scene is changing rapidly than when it is still. Hence, just knowing the long term average bandwidth may not be sufficient.

For example, consider a situation where 10 flows arrive at a switch on separate ports and they all leave on the same 10 Mbps link. If each flow is expected to send no more than 1 Mbps, there would be no problem at all. However, if these are variable bit applications such as compressed video, it is possible that they will send at rates higher than the average. If enough sources send more than average rates, then the total rate at which data arrives at the switch will be more than 10 Mbps. This excess data will be queued, leading to delays. And if such a condition persists, the longer the queue will get, and we may not be able to be within the delay bound. Hence, we need a better mechanism to define the traffic characteristics of the source.

*A Token Bucket Filter* is used to define the bandwidth characteristics of a source. The filter is described by two parameters: Token rate *r &* Bucket depth *B.*

Remember the way token bucket works. To be able to send a byte, a token is needed. To send a packet of length *n*, *n* tokens are needed. Initially there are no tokens. Tokens are accumulated at a rate of *r* per second. No more than *B* tokens can be accumulated.

Let us see how this characterization can be used to specify different types of traffic. Consider the two flows given in Fig. 27.2.

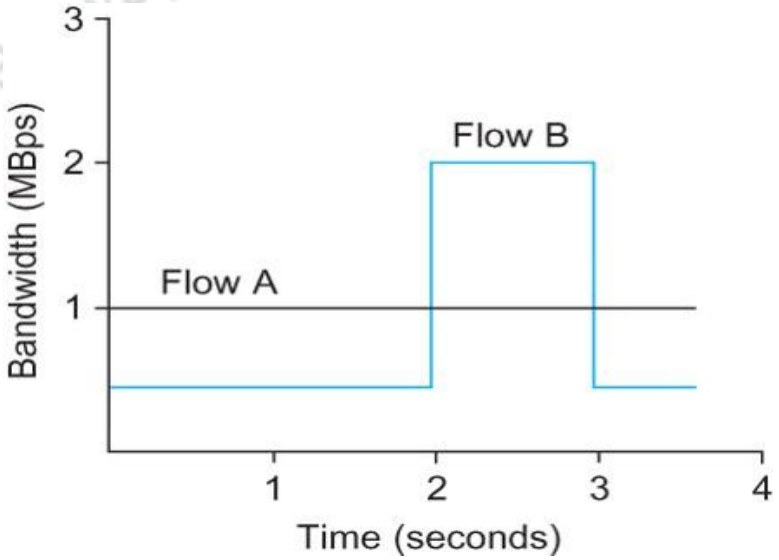


***Figure 27.2 TSpec Example***

Assume each flow can send data as individual bytes rather than as packets. Flow A generates data at a steady rate of 1 MBps. It can be described by a token bucket filter with a rate *r* = 1 MBps and a bucket depth of 1 byte. This means that it receives tokens at a rate of 1 MBps but it cannot store more than 1 token, it spends them immediately.

Flow B sends at a rate that averages out to 1 MBps over the long term, but does so by sending at 0.5 MBps for 2 seconds and then at 2 MBps for 1 second. Since the token bucket rate *r* is a long term average rate, flow B can be described by a token bucket with a rate of 1 MBps.

Unlike flow A, flow B needs a bucket depth *B* of at least 1 MB, so that it can store up tokens while it sends at less than 1 MBps, which can be used when it sends at 2 MBps. For the first 2 seconds, it receives tokens at a rate of 1 MBps but spends them at only 0.5 MBps. So it can save up $2 \times 0.5 = 1$ MB of tokens, in the first two seconds, which it spends at the 3rd second.

Thus, different traffic characteristics are specified using the token bucket filter mechanism.

**27.4 Data Plane in an IntServ Router**

Putting all these together, the data plane in an IntServ router will perform flow-based classification, flow-based queuing, traffic policing using token bucket filter, and WFQ scheduling, as shown in Fig.27.3.

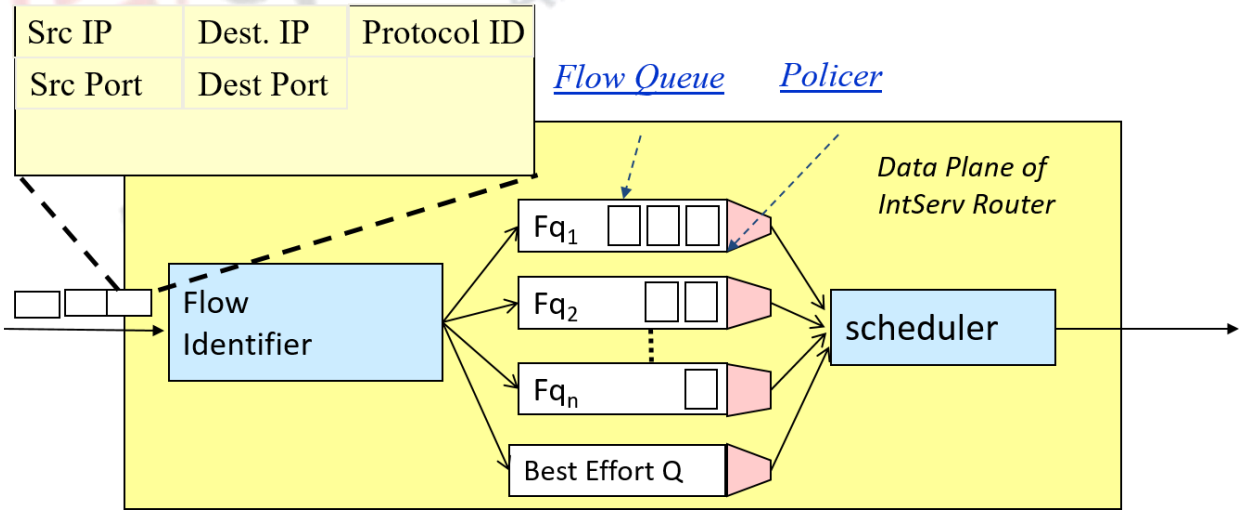



***Figure 27.3 Data plane in an IntServ router***

The flow identification is performed using the 5 fields – Source IP, Source Port, Destination IP, Destination Port, and the Protocol ID. There is a per-flow queue, and a

default best-effort queue. Policing is done as per the TSpec. The WFQ scheduler then sends the packets out.

The control plane is used to admit the flows, and the protocol used for that purpose is RSVP. Let us take a look at RSVP now.

## 27.5 Reservation protocol (RSVP)

There are a few requirements that are expected of a reservation protocol, to coexist with the existing IP protocol. It should not detract from the robustness that we find in today's connectionless networks. Because connectionless networks rely on little or no state being stored in the network itself, it is possible for routers to crash and reboot and for links to go up and down while end-to-end connectivity is still maintained. This robustness should not be compromised. It should not replicate routing functionality, and should co-exist with route changes.

RSVP tries to maintain this robustness by using the idea of soft state in the routers. It is not a routing protocol, and only makes reservations.

It should provide support for multicast. In a multicast scenario, different receivers have different capabilities and want different QOS. All of them must be supported seamlessly. Changes in group membership should not be expensive. Reservations should be aggregate – i.e. each receiver in group should not have to reserve. It should also be possible to switch allocated resource to different senders.

RSVP handles these requirements by using ***receiver-initiated*** reservations, as opposed to sender-oriented reservations. The reservation messages are sent on the multicast tree and this helps to have aggregate reservations for the group.

### *27.5.1 RSVP operation*

RSVP rides on top of unicast/multicast routing protocols. It must be present at sender(s), receiver(s), and routers. It carries resource requests all the way through the network. At each hop, it consults admission control and sets up the reservation. It informs the requester in case of failure.

The receiver needs to know the sender's TSpec. It also needs to know what path the packets will follow from sender to receiver, so that it can establish a resource reservation at each router on the path. Both these requirements are met by sending a message (PATH message) from the sender to the receiver that contains the TSpec.

Each router looks at this PATH message as it goes by, and it figures out the *reverse path that will be used to send* reservations from the receiver back to the sender. Having received a PATH message, the receiver sends a reservation back "up" the multicast tree in a RESV message.

This message contains the sender's TSpec and an RSpec describing the requirements of this receiver. Each router on the path looks at the reservation request and tries to allocate the necessary resources to satisfy it. If the reservation can be made, the RESV request is passed on to the next router. *If not, an error message is returned to the receiver who made the request.*

If all goes well, the correct reservation is installed at every router between the sender and the receiver. As long as the receiver wants to retain the reservation, it sends the same RESV message about once every 30 seconds.

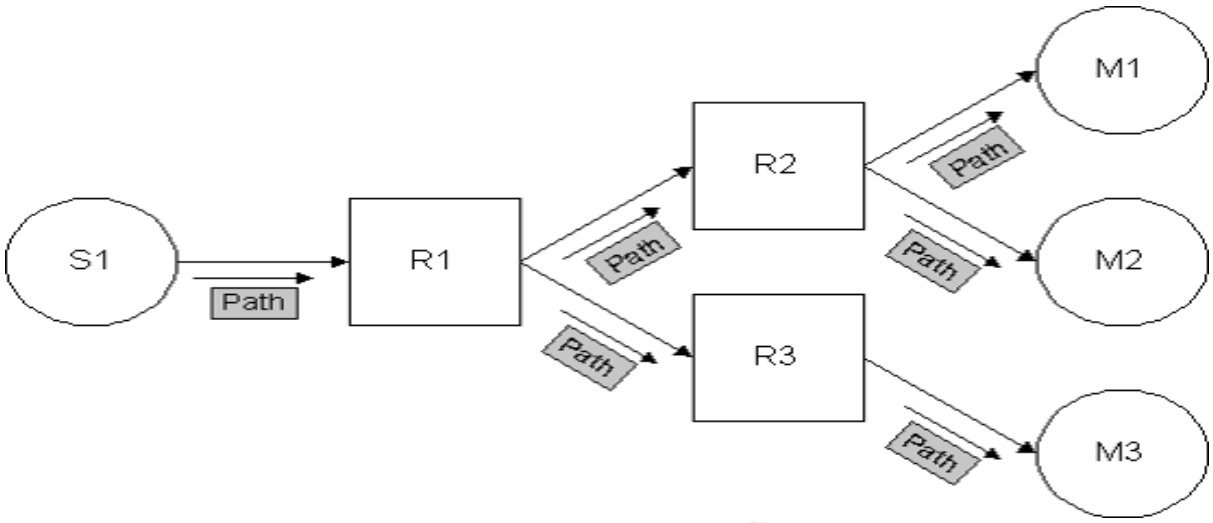
### *27.5.2 RSVP details*

Traffic in RSVP is defined in terms of sessions. A session is identified by the tuple <Destination IP address - Unicast/Multicast, Destination port number>. Parameters used to describe the traffic are Filter Spec (Type of filter), and Flow Spec (Sender : Tspec, Rx : Rspec).

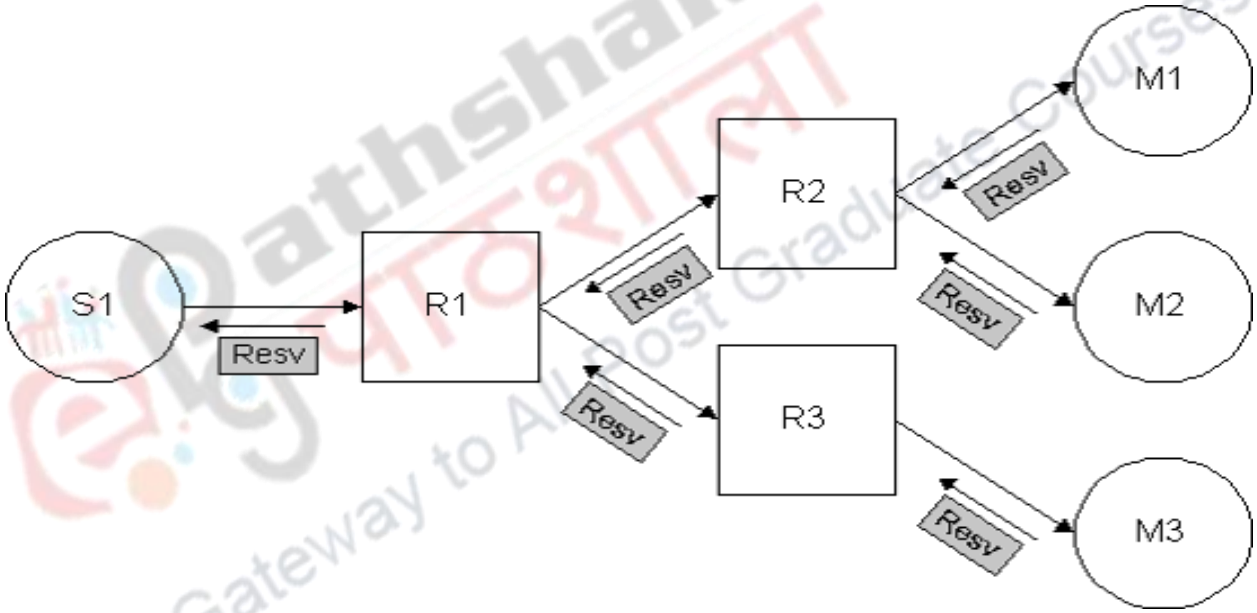
*RSVP Messages used are the following:*

(i) PATH message
(ii) RESV message
(iii) CONFIRMATION message
   a. Generated only upon request.
   b. Unicast to receiver when RESV reaches node with established state.
(iv) TEARDOWN message
**(v)** ERROR message (if PATH or RESV fai**ls)**

The PATH message is sent periodically by sender towards all destinations. This message helps to set up path from the sender to each destination. It contains *TSpec,* based on token bucket model – with parameters maximum bandwidth, token bucket size, and Maximum packet size. Fig 27.4(a) shows an example of sending PATH messages.




***(a) PATH message***




***(b) RESV message***

***Figure 27.4(a) PATH message (b) RESV message in RSVP***

Receivers request for resources using RESV message. The RESV messages are sent from the receivers to the source upstream. These messages are sent along the route set by PATH messages. If there are no senders, no reservation can be made. The RESV information is merged as the message proceeds upstream.

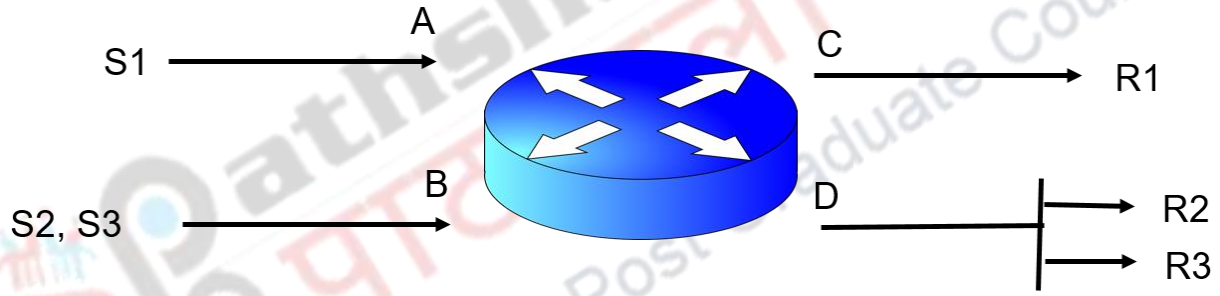
RESV messages are propagated upwards only if the reservation at that particular router is less than the requested QoS parameters. This helps in conserving resources in a multicast setting, by suppressing redundant reservation messages. An example of RESV message being sent up a multicast tree is shown in Fig 27.4(b).

There are two types of tear down messages – *pathtear,* which is initiated by the sender; and *resvtear,* which is initiated by the receiver.

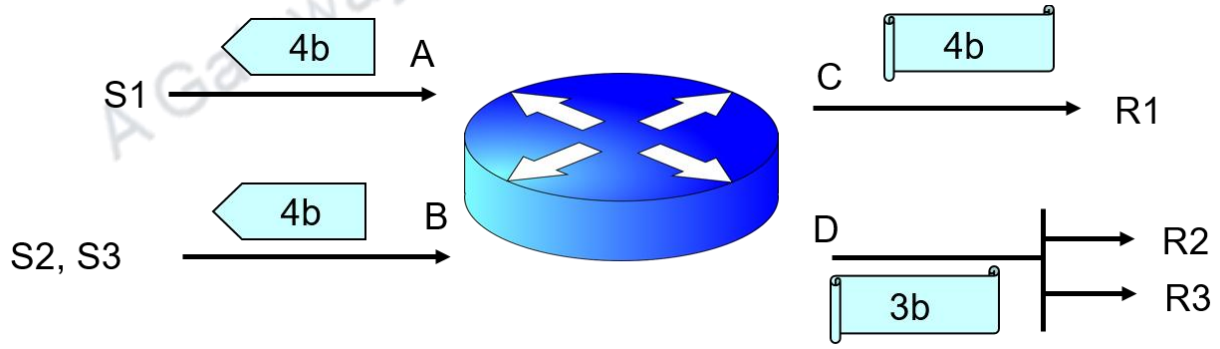
RSVP supports three different reservation styles: Wildcard/No filter, Fixed filter, and Dynamic filter (or shared filter).

Wildcard filter does not specify a particular sender for a reservation. In Fixed filter, a sender is explicitly specified for a reservation. Dynamic (shared) filter allows reservation to be shared among valid senders. The senders can also be changed over time. The concept of these three filters is illustrated in the example given in Fig.27.5.

Consider three senders S1, S1, S3, and 3 receivers R1, R2, R3, connected to router interfaces A,B,C,D, as shown in Fig.27.5(a). Fig27.5(b) shows the reservations for a Wildcard Filter, where R1, R2, and R3 want to reserve 4b, 3b, and 2b, respectively (b is given rate). The reservations towards both S1, and S2 is the maximum of the reservations, i.e., 4b.



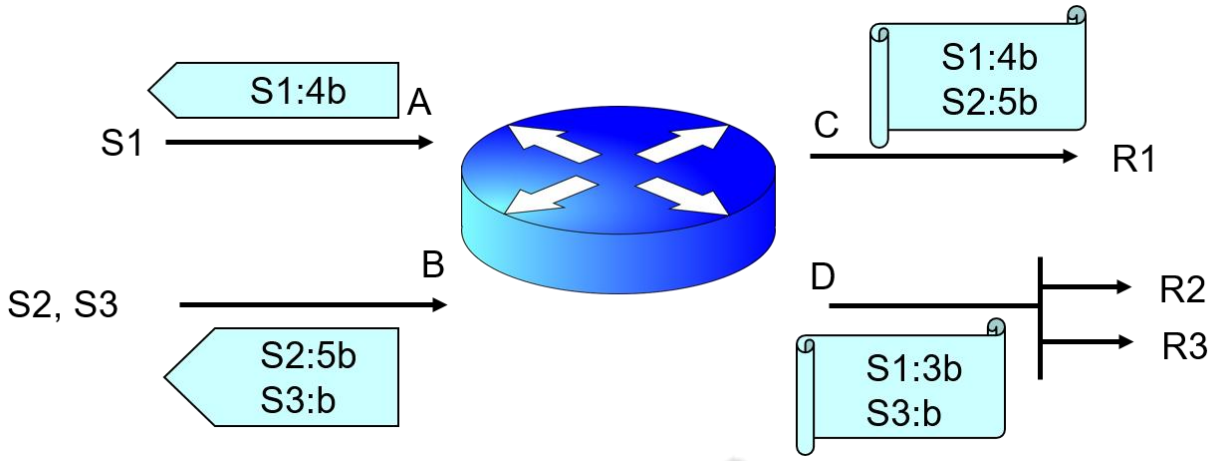

***(a) Example for filters***




***(b) Wildcard Filter reservation***

A Fixed Filter reservation, where R1 wants to reserve 4b for S1 and 5b for S2, R2 wants to reserve 3b for S1 and b for S3; and R3 wants to reserve b for S1, is shown in Fig.27.5(c). The maximum aggregated reservation for S1 is 4b, and for S2 is 5b, and S3

b. These are sent towards S1, and S2,S3 through the interfaces A and B respectively.




***(c) Fixed Filter Reservation***

A shared filter reservation, where, R1 wants to reserve b for S1 and S2 (shared); R2 wants to reserve 3b for S1 and S3 (shared), and R3 wants to reserve 2b for S2, is shown in Fig.27.5(d). On combining the requests for S1, a value of 3b is reserved for S1, and 3b for (S2,S3).

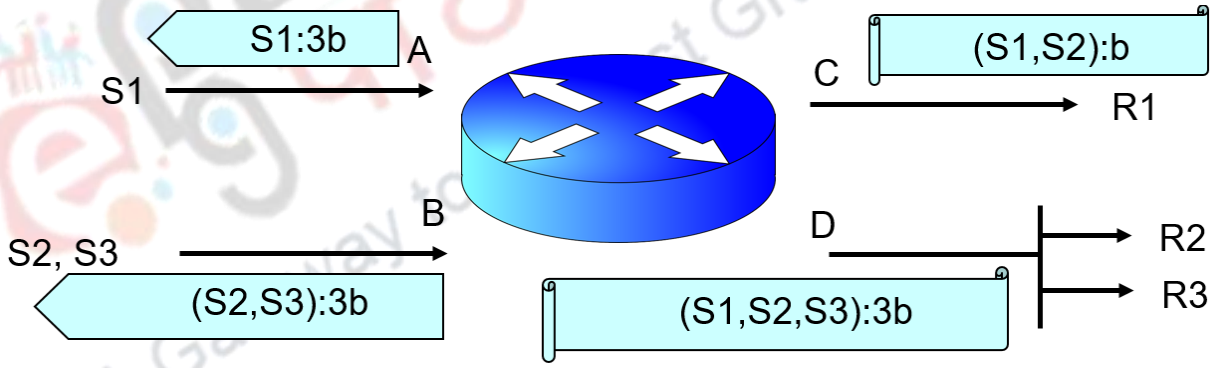



***Figure 27.5(d) Shared Filter***

*27.5.2.1 Soft state maintenance*

RSVP uses the idea of soft-state, i.e., the reservations are not made permanently, but need to be refreshed periodically. Routers along path would remove reservations based on timeouts. Hence, the PATH and RESV messages have to be sent periodically, to keep the reservation alive. Although this results in message overhead, the advantage is that network resources are not reserved forever in case of node failure.

***27.5.3 RSVP based Traffic Engineering***

Although RSVP was proposed to work with IntServ, it has actually been used for other

purposes, along with other protocols as well. In practice, IntServ has not been very popular, due to the high overheads in maintaining per-flow state at all the routers.

One of the major applications of RSVP is in Traffic Engineering (TE). It is used to establish LSPs (Label Switched Path) in MPLS (MultiProtocol Label Switching) networks. It essentially enables source routing. Once a path is specified in the core, routers route packets based on labels. It is also used in optical networks.

### 27.6 Summary

To summarize, we have examined two protocols in this module - IntServ and RSVP. We have seen that IntServ is a fine-grained approach, where flow related QoS is specified using FlowSpec comprising of TSpec, and RSpec. We have discussed the two types of services provided by IntServ, namely, guaranteed service & controlled load service.

We have examined the principle of RSVP, the soft-state idea and its operations including its messages, and filter specifications.

### Acknowledgements & References


1. *Computer Networking: A Top Down Approach Featuring the Internet*, 6th edition. Jim Kurose, Keith Ross Addison-Wesley, 2012.
2. Computer Networks: A systems Approach, 4th edition, David Peterson, Davie, Morgan Kauffman, 2012.
3. Computer Networks An Open Source Approach, Ying-Dar Lin, Ren-Hung Hwang, Fred Baker, McGraw Hill, 2012.
4. Abhay K. Parekh and R. Gallager, “A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single Node Case” IEEE Transactions on Networking, June 1993.