

The Integrated Services in the Internet: State of the Art

Author: Paul P.White, Jon Crowcroft

Department of Computer Science
University College London
Gower Street
London
WC1E 6BT

Tel: 44 171 419 3701, +44 171 380 7296

Email: p.white@cs.ucl.ac.uk, j.crowcroft@cs.ucl.ac.uk

Abstract

This paper is about the evolution of the Internet from a simple data network into a true multiservice network that can support the emerging multimedia applications and their protocols with appropriate performance and costs. The real-time delivery and specific bandwidth requirements of these multimedia applications have created a need for an Integrated Services Internet in which traditional best-effort datagram delivery can coexist with additional enhanced Quality of Service (QoS) delivery classes. The Integrated Services Internet will be able to commit to meet bandwidth, packet loss and delay specifications for individual data flows by using the Resource ReSerVation Protocol (RSVP), together with appropriate packet forward scheduling policies.

Keywords

Controlled-Load Service
Data Flow
Guaranteed Service
Integrated Services (IS)
One Pass with Advertising (OPWA)
Quality of Service (QoS)
ReSerVation setup Protocol (RSVP)
Real Time Protocol (RTP)
Random Early Detection (RED)
Multicast

1.0 Introduction

The Internet has evolved from a classical, closed community data network into the infrastructure for the Global Information Society. Users are coming to trust the Internet for business, including not only the burgeoning use of the World Wide Web, and traditional applications such as Email and File Transfer, but increasingly for “real-time” multimedia such as audio, video and shared applications (such as a whiteboard, or meeting planner or shared document editor).

The original design goal of the Internet was to serve as a highly fault tolerant data network for the defence community. To this end, the amount of state shared between the network and end systems is minimised, being merely sufficient information to calculate a set of routes, and, for each packet sent by an end system, to determine the best current route. In its original implementation, a great deal of attention was paid to this distributed, dynamic route calculation, and not too much to the performance aspects of packet forwarding. It was not anticipated that the typical military users would *share* the network with other users, or let it become overloaded. Because of this, early router implementations used FIFO (First In First Out) queues for all traffic, whose behaviour under overload conditions was simply to drop the latest arriving packets¹. The only reason for this was simplicity.

Over time, it became apparent that traffic management, for different applications, for different users, and for overloaded networks, would be essential. A number of stages in the evolution of the protocols and implementations of the Internet have followed:

1. The first service differentiation was by application, and the idea of *Type of Service* forwarding was proposed, albeit only ever deployed in a few research networks[21]. Conceptually, routers distinguish traffic either by examining special bits in the IP packet header, or by application protocol (deduced by spying at high level protocol fields, specifically, the transport protocol port numbers- this is often feasible since routers also commonly implement filters for security reasons based on these same fields). ToS routing, as it is often known, is a **win-win** enhancement for some mixes of applications - for example, bulk transfer versus interactive terminal traffic, can see a mutual benefit by allowing higher throughput for large file transfers, but lower delay for light traffic flows such as telnet typically offers. Although some commercial routers have a ToS forwarding feature, few applications currently take advantage of it..
2. Telnet, FTP and WWW's application protocol, HTTP, all operate over TCP, the Transmission Control Protocol, which enhances the inherently unreliable Internet Protocol, to provide ordered, reliable delivery. In 1988, TCP was re-engineered, a decade after its original design, to provide an end-to-end congestion control. The net effect of ToS routing and TCP Congestion Control is to offer a network that provides a **fair share** to the set of concurrent users. [In fact, it is largely only an approximation to a fair share, but more recent work on slightly different queueing strategies may improve this - we discuss this later.] Because all TCP users achieve only a fair share of the capacity available at any moment, it has been common not to collect time-based usage charges, since the time to transfer any item of data depends on the number of other active users. An access leasing system is operated

¹ UDP was experimentally used for carrying voice traffic even in the early ARPANET days, but the network was typically not overloaded, so that packets encountered no significant queues, and there was no need for any traffic control. We discuss this more later.

by most commercial Internet Service Providers, and this leads to very efficient cost recovery.

3. Group communication is very useful for applications that involve multiple simultaneous senders and receivers. The *Mbone* is a virtual overlay on the current Internet that provides multicast [17], which results in a massive reduction of load on the network for such applications. Initially, it was expected that this might be used for replicated transactions, but in the absence of transport (end-to-end) protocols to support these, what was deployed more quickly was a family of applications based on the User Datagram Protocol, namely the audio and video conferencing programs.
4. Typical applications based on UDP offer an approximately fixed rate of packets to the network.² Fundamentally, to support arbitrary numbers of users of these applications, and not suffer overload, the Internet had to be enhanced in some way. A number of possibilities exist[22].
 - i) Over Engineering
 - ii) Resource Reservation
 - iii) Usage Based Charging

The first approach may work for a large class of applications, but many researchers and practitioners believe that there will always be some application that can dominate the network capacity, and so ii) and perhaps iii) as well are needed.

The second of these has received most attention in the protocol standards development arena, and we discuss this at length in the rest of this paper. The third of these is perceived as complex in the Internet community and possibly very hard to deploy - typically, commercial Internet Providers find this difficult to install in the absence of a set of global agreements on tariffs.

2.0 Application Requirements

While computers have infinite patience, humans have limited tolerance for delay. This is reflected mostly in the throughput requirements for traditional applications in the Internet, such as FTP, Email and WWW. With multimedia applications, it is more important that the quality of communication is maintained. This means that latency must be bounded and jitter must be controlled, but also that both loss and a minimum throughput are tolerable. With sufficient engineering [19], applications may be highly loss tolerant, but there are still limits.

It is possible that either simple congestion, or pricing alone will cause people to cease to use applications when a network is overloaded. However, it depends very much on the mix of end-to-end protocol behaviour in a FIFO queuing system with "tail-drop", just exactly what mix of applications are running at the stable operating point of a full network. Provided that all applications use the same scheme to adapt to a bottleneck, there is no problem in achieving reasonable fairness. However, multicast UDP based

² While there are some adaptive multimedia applications, most notably Steve McCanne's work on Vic, Ian Wakeman's work on Ivs and the INRIA Free Phone and UCL's Robust Audio Tool [18], these are currently a minority of approaches. In any case, an audio or video application does have a limited acceptable range over which it can adapt.

applications cannot use the feedback scheme used in TCP, and as yet, such congestion control is not well understood.³

One alternative approach is to alter the network to keep some kind of minimal state *by observation* of current traffic, and to penalise so-called badly-behaved sources. One such scheme is Random Early Detection [20], which is extremely promising. As the name suggests, Random *Early* Detection is a mechanism to give feedback at the onset of congestion, rather than after congestion has occurred. Essentially, the mechanism uses the persistent presence of a number of packets in a bottleneck queue as an indication of non-co-operative behaviour, and penalises such sources with a higher loss rate. Under normal conditions, adaptive sources adapt sooner, so that the queues are shorter than without RED operating. This leads to more buffer being available for legitimate short term bursts. A bias against persistent long term bursts must still be retained to prevent hogging.

However, it must be assumed that the “gene pool” of congestion control, or adaptive applications, may increase to more than 1 (namely TCP’s), and that at such a point, the service differentiation task will become more complex. In any case, RED only alleviates congestion. It cannot guarantee a minimum packet rate that will be forwarded. In such cases, it is believed that dynamic specification of a source (or receiver) requirement is preferable. To this end, RSVP can be used, for both unicast, and multicast, to request “special treatment” for a traffic flow.

There are two main classes of multimedia applications that might make such a request:

1. Adaptive applications, with users that may benefit from some minimum capacity being guaranteed at some reasonable probability.
2. Legacy, constant rate applications with a requirement for strong guarantees of service.⁴

In the next three sections, we describe the emerging system for supporting these services in detail.

3.0 IETF Integrated Services

In response to the growing demand for an Integrated Services Internet, the Internet Engineering Task Force (IETF)[14] set up an Integrated Services (intserv) Working Group[15] which has since defined several service classes that if supported by the routers traversed by a data flow⁵ can provide the data flow with certain QOS

³ “Drop Tail” is the term sometimes used to describe the behaviour of a FIFO queuing system when it runs out of buffer. Packets arriving at the full buffer are dropped. This is in contrast to RED and related schemes, where packets are dropped from other (perhaps random) places in the overloaded queue to reflect more fairly the different contribution of different sources to the overload circumstance.

⁴ In the ATM standards, there is a variety of network services targeted at a range of applications that is quite similar (although extending to other application models too). The specification of the application behaviour, and the network service are also separated in a way that is quite similar to the Integrated Service Internet, as we shall see below.

⁵ A data flow identifies the set of packets to receive special QOS. It is defined by a ‘Session’ comprising the IP address, transport layer protocol type and port number of the destination along with a list of specific senders to that Session that are entitled to receive the special QOS. Each sender

commitments. By contrast best-effort traffic entering a router will receive no such service commitment and will have to make do with whatever resources are available. The level of QOS provided by these enhanced QOS classes is programmable on a per-flow basis according to requests from the end applications. These requests can be passed to the routers by network management procedures or, more commonly, using a reservation protocol such as RSVP which is described in section 4. The requests dictate the level of resources (e.g. bandwidth, buffer space) that must be reserved along with the transmission scheduling behaviour that must be installed in the routers to provide the desired end-to-end QOS commitment for the data flow.

In determining the resource allocations necessary to satisfy a request the router needs to take account of the QOS support provided by the link layer in the data forwarding path. Furthermore, in the case of a QOS-active link layer such as ATM or certain types of LAN the router is responsible for negotiations with the link layer to ensure that the link layer installs appropriate QOS support should the request be accepted. This mapping to link-layer QOS is medium-dependent and the mechanisms for doing so are currently being defined by the Integrated Services over Specific Lower Layers (issll) Working Group of the IETF[16]. In the case of a QOS-passive link-layer such as a leased-line the mapping to the link-layer QOS is trivial since transmission capacity is handled entirely by the router's packet scheduler.

Each router must apply admission control to requests to ensure that they are only accepted if sufficient local resources are available. In making this check, admission control must consider information supplied by end applications regarding the traffic envelope⁶ that their data flow will fall within. One of the parameters in the traffic envelope that must be supplied is the maximum datagram size of the data flow, and should this be greater than the MTU of the link then admission control will reject the request since the Integrated Services models rely on the assumption that datagrams receiving an enhanced QOS class are never fragmented.

Once an appropriate reservation has been installed in each router along the path, the data flow can expect to receive an end-to-end QOS commitment provided no path changes or router failures occur during the lifetime of the flow⁷, and provided the data flow conforms to the traffic envelope supplied in the request. Service-specific policing and traffic reshaping actions as described in sections 3.1 and 3.2 will be employed within the network to ensure that non-conforming data flows do not affect the QOS commitments for behaving data flows. The IETF has considered various QOS classes such as [1][6][10][12] although to date only two of these, Guaranteed Service[10] and Controlled-Load Service[12], have been formally specified for use with RSVP[13]. First, we will discuss the simpler of these services, namely Controlled Load.

3.1 Controlled-Load Service

Controlled-Load Service[12] provides approximately the same quality of service under heavy loads as under light loads,. A description of the traffic characteristics

is identified by source address, and port number while it's protocol type must be the same as for the Session.

⁶ Think of the packet arrival process as having a pattern like a waveform. The envelope is its typical shape.

⁷ Due to refresh timers used by RSVP, a flow may recover its QoS commitment without taking any special action. With the advent of QoS aware routing, it may be that there is not even a gap in the perceived provision of the service contract.

(the Tspec, described in 3.2) for the flow desiring Controlled-Load Service must be submitted to the router as for the case of Guaranteed Service although it is not necessary to include the peak rate parameter. If the flow is accepted for Controlled-Load Service then the router makes a commitment to offer the flow a service equivalent to that seen by a best-effort flow on a lightly loaded network. The important difference from the traditional Internet best-effort service is that the Controlled-Load flow does not noticeably deteriorate as the network load increases. This will be true regardless of the level of load increase. By contrast, a best-effort flow would experience progressively worse service (higher delay and loss) as the network load increased. The Controlled-Load Service is intended for those classes of applications that can tolerate a certain amount of loss and delay provided it is kept to a reasonable level. Examples of applications in this category include adaptive real-time applications.

Controlled Load has some fairly simple implementations, in terms of the queueing systems in routers. It also functions adequately for the existing Mbone applications, which can adapt to the modest (small) scale end-to-end delay and variations and jitter that it may introduce, through the use of adaptive playout buffering [19]. It is not suited to applications that require very low latency (e.g. distributed VR systems and so forth). Next we discuss the service provided where the user requires some commitment to a delay guarantee, namely the Guaranteed Service.

Example of Controlled Load Usage/Requirement

One possible example of the use of the controlled load service is that of Mbone applications, over a private so-called Intranet, where traffic conditions and global policies can be managed such that a statistical throughput guarantee is enough, and propagation delays will be low enough that for most users, interactive software based multimedia conferencing tools will perform adequately. A more interesting example might be the provision of SNA or DEC LAT tunnelling across a public Internet Service Provider's backbone network. SNA and DEC LAT are both somewhat delay sensitive due to their detailed protocol operations, although not as much as some real time systems are. However, using the same Internet path to carry them with arbitrary interference from other applications' flows would not work well (or at all).

3.2 Guaranteed Service

Guaranteed Service[10] provides an assured level of bandwidth, a firm end-to-end delay bound and no queuing loss for conforming packets of a data flow. It is intended for applications with stringent real-time delivery requirements such as certain audio and video applications that have fixed "play-out" buffers and are intolerant of any datagram arriving after their playback time. Guaranteed Service really addresses the support of "legacy" applications that expect a delivery model similar to traditional telecommunications **circuits**.

Each router characterises the Guaranteed Service for a specific flow by allocating a bandwidth, R and buffer space, B that the flow may consume. This is done by approximating the "fluid model" of service[8][9] so that the flow effectively sees a dedicated wire of bandwidth, R between source and receiver. In a perfect fluid model a flow conforming to a token bucket of rate, r and depth, b will have its delay bounded by b/R provided $R \geq r$. To allow for deviations from this perfect fluid

model in the router's approximation⁸, two error terms, C and D are introduced. These errors arise from the finite packet sizes that are being dealt with. For example, any packet may experience an excess delay as it is forwarded due to the size of the packets in the same queue, and due to inaccuracies in scheduling from packets (of a possibly different size) in other queues bound for the output link. These terms are derived for Weighted Fair Queueing schedulers in Parekh's seminal work [8][9]. Consequently the delay bound now becomes $b/R + C/R + D$. However with Guaranteed Service a limit is imposed on the peak rate, p of the flow which results in a reduction of the delay bound. In addition, the packetisation effect of the flow needs to be taken into account by considering the maximum packet size, M ⁹. These additional factors result in a more precise bound on the end to end queuing delay as follows:

$$Q_{delay_{end2end}} = \frac{(b - M)(p - R)}{R(p - r)} + \frac{(M + C_{tot})}{R} + D_{tot} \quad (\text{case } p > R \geq r) \quad (1)$$

$$Q_{delay_{end2end}} = \frac{(M + C_{tot})}{R} + D_{tot} \quad (\text{case } R \geq p \geq r) \quad (2)$$

The composed terms, C_{tot} and D_{tot} represent the summation of the C and D error terms respectively for each router along the end-to-end data path. In (1), there are three delay terms, made from the contributions of the burst of packets, b , (the bucket depth) sent at the peak rate p , and serviced at the output link rate R , plus the sum over all hops, of errors introduced at each hop due to a packet size worth of fluid flow approximation, plus a third term, made up of cross traffic scheduling approximation contributions. In (2), the first term is absent, since the link rate is greater than the peak, so there are no packets queued from this flow itself.

In order for a router to invoke Guaranteed Service for a specific data flow it needs to be informed of the traffic characteristics of the flow, T_{spec} , along with the reservation characteristics, R_{spec} . Furthermore to enable the router to calculate sufficient local resources to guarantee a lossless service it requires the terms C_{sum} and D_{sum} which represent the summation of the C and D error terms respectively for each router along the path since the last re-shaping point(see below).

T_{spec} parameters

p = peak rate of flow (bytes/second)

b = bucket depth (bytes)

r = token bucket rate (bytes/second)

m = minimum policed unit (bytes)¹⁰

M = maximum datagram size (bytes)

R_{spec} parameters

R = bandwidth, i.e. service rate (bytes/second)

S = Slack Term (ms) (see section 4.6)

⁸ Among other things the router's approximation must take account of the medium-dependent behaviour of the link layer of the data forwarding path.

⁹ While the Internet Protocol permits, in principle, a wide range of packet sizes, in practice, the range supported makes stating this upper limit practical and realistic.

¹⁰ Policing will treat any IP datagram less than size m as being of size m .

Guaranteed Service traffic must be policed at the network access points to ensure conformance to the Tspec, so that traffic does not interfere with other flows and cause them to miss their contract. We discuss this more below.

In addition to policing of data flows at the edge of the network, Guaranteed Service also requires reshaping of traffic to the token bucket of the reserved Tspec at certain points on the distribution tree. Any packets failing the reshaping are treated as best-effort and marked accordingly if such a facility is available. Reshaping¹¹ must be applied at any points where it is possible for a data flow to exceed the reserved Tspec even when all senders associated with the data flow conform to their individual Tspecs. Such an occurrence is possible in the following 2 cases.

1. At branch points in the distribution tree where the reserved Tspecs of the outgoing branches are not the same. In this case the reserved Tspec of the incoming branch is given by the ‘maximum’¹² of the reserved Tspecs on each of the outgoing branches. Consequently some of the outgoing branches will have a reserved Tspec that is less than the reserved Tspec of the incoming branch and so it is possible that in the absence of reshaping, traffic that conforms to the Tspec of the incoming branch might not conform when routed through to an outgoing branch with a smaller reserved Tspec. As a result, reshaping must be performed at each such outgoing branch to ensure that the traffic is within this smaller reserved Tspec.
2. At merge points in the distribution tree for sources sharing the same reservation since in these cases the sum of the Tspecs relating to the incoming branches will be greater than the Tspec reserved on the outgoing branch. Consequently when multiple incoming branches are each simultaneously active with traffic conforming to their respective Tspecs it is possible that when this traffic is merged onto the outgoing branch it will violate the reserved Tspec of the outgoing branch. Hence reshaping to the reserved Tspec of the outgoing branch is necessary.

This reshaping will necessarily incur an additional delay (essentially, to smooth a collection of peaks over some troughs of traffic flow must entail slowing down early packets, since one obviously cannot speed up later packets).

Killer Reservation Problems

When merging heterogeneous reservation requests from receivers onto the tree flowing from the same source, there is an additional problem known as the “Killer Reservation” problem, which manifests itself in two ways:

1. A large reservation made subsequent to an existing smaller reservation may fail. If this is the case, a naïve implementation will cause the entire reservation to fail. The solution to this is to introduce extra states into the reservation protocol such that subsequent failures do not break existing reservations.
2. A receiver may continually attempt to make a large reservation, retrying quickly after every failure. This may continually block a smaller reservation request that might otherwise succeed. Again, a merge point might keep state concerning recent

¹¹ Traffic conforming to a leaky bucket specification has still some degrees of freedom to take different shapes within the envelope. Shaping can improve the way such traffic mixes and improve its buffering requirements.

¹² Maximum according to rules defined in [3].

failed reservations and favour new ones that are more likely to succeed over retries for ones that have recently failed.

Example of use/requirement for Guaranteed Service.

There are a number of applications in the military and commercial worlds which have hard delay bound requirements. For example, the Distributed Interactive Simulation program has a scenario with 100,000 participants in an online war game simulation, where the applications send messages to each other that represent events between objects in the real world. Participants should see progress (missiles hitting tanks) in an ordered, and timely fashion. A somewhat difference scenario, but with remarkable similar network guarantee requirements, is that of share dealer networks. Here, share price advertisements are multicast to the dealer terminals, with hard delay bounds on delivery delays (and rates), and delay bounds on the response times, since the price in the *real world* is varying, possibly very rapidly, and the requirement is that a bid to buy at a price does not encounter an offer to sell that is significantly out of date.

3.3 Policing and Conformance of Controlled Load and Guaranteed Services

Routers implementing the Controlled-Load and Guaranteed Services must check for conformance of data flows to their appropriate reserved Tspecs. This is known as policing. Any non-conforming data flows must not be allowed to affect the QOS offered to conforming data flows or to unfairly affect the handling of best-effort traffic. Within these constraints the router should attempt to forward as many of the packets of the non-conforming data flow as possible. This might be done by dividing the packets into conforming and non-conforming groups and forwarding the non-conforming group on a best effort basis. Alternatively, the router may choose to degrade the QOS of all packets of a non-conforming data flow equally.

The usual enforcement policy is to forward non-conforming packets as best-effort datagrams¹³.

An additional requirement for policing over that of meeting the global traffic contract, is that there are possible consequences for pricing if excess traffic is not seen to receive a lesser service guarantee.

3.4 Instantiating Integrated Services on Given Link Technology

Assuming a backbone network is implemented as a set of routers connected together by point-to-point circuits, integrated services must be implemented by putting in place the appropriate queueing strategies in the routers. Typically, theory tells us that Weighted Fair Queueing (or hierarchical round robin service) will provide at least a baseline for implementation. In fact, for controlled load, simple priority queueing

¹³Action with regard to non-conforming datagrams should be configurable to allow for situations such as traffic-sharing where the preferred action might be to discard non-conforming datagrams. This configuration requirement also applies to reshaping. If and when a marking facility (e.g. a bit in the Ipv6 header to indicate that a packet has exceeded its flow parameters) becomes available these non-conforming datagrams should be marked to ensure that they are treated as best-effort datagrams at all subsequent routers

schemes may suffice. There are a number of other service disciplines being researched.

Where routers are interconnected by other types of links, particularly shared media (LANs, Satellite channels etc.), or switches, then the interconnect technology must be controlled as well. In the standards work, specifications are emerging for mapping the Guaranteed Service and Controlled Load to run over Token Ring, SMDS, Frame Relay, and a variety of Transfer Capabilities and QoS Classes on switched ATM networks.

For non-deterministic (but popular) subnetworks such as Ethernet, the technology must be enhanced somehow. This is a matter for current research.

4.0 Resource ReSerVation Protocol (RSVP)

The Resource ReSerVation Protocol RSVP[3] was designed to enable the senders, receivers and routers of communication sessions(either multicast or unicast) to communicate with each other in order to set up the necessary router state to support the services described in sections 3.1 and 3.2. It is worth noting that RSVP is not the only IP reservation protocol that has been designed for this purpose. Others include ST-II[11] and ST-II+[5] which incidentally contain some interesting architectural differences to RSVP such as the use of hard-state and sender-initiated¹⁴ reservations rather than soft-state¹⁵ and receiver-initiated reservations as in RSVP. However for the rest of this paper the only reservation protocol we consider is RSVP since currently this has the most industry support. For further discussion on the mentioned alternatives the interested reader can refer to [7].

RSVP is a novel signaling protocol in at least 3 ways:

1. It accommodates multicast, not just point-to-multipoint (one-to-many) reservations. To this end, the receiver driven request model permits heterogeneity, in principle, and the filter mechanism allows for calls that reserve resources efficiently for the aggregate traffic flow (e.g. for audio conferencing).
2. It uses *soft state*, which means that it is tolerant of temporary loss of function without entailing fate-sharing between the end systems and the network nodes. This means that QoS routing can be deployed separately (in more than one way!).
3. RSVP is quite straightforward in packet format and operations, and so is relatively low cost in terms of implementation in end systems and routers. One thing that RSVP is *not* is a routing protocol. RSVP does not support QoS-dependent routing itself (in other words, such routing is independent of RSVP, and could precede or follow reservations).

RSVP identifies a communication session by the combination of destination address, transport layer protocol type and destination port number. It is important to note that each RSVP operation only applies to packets of a particular session and as such every RSVP message must include details of the session to which it applies. For the remainder of this tutorial it will be assumed that any discussion is for a single session only. In addition, although RSVP is applicable to both unicast and multicast sessions

¹⁴ ST-II+ permits both sender and receiver-initiated reservations, ST-II permits sender-initiated reservations only.

¹⁵ With hard-state the network is responsible for reliably maintaining router state whereas with soft-state the responsibility is passed to the end-systems which must generate periodic refreshes to prevent state timeout.

we concentrate on the more complicated multicast case. We do not discuss RSVP policy issues such as authorization, authentication, and questions relating to charging for reservations.

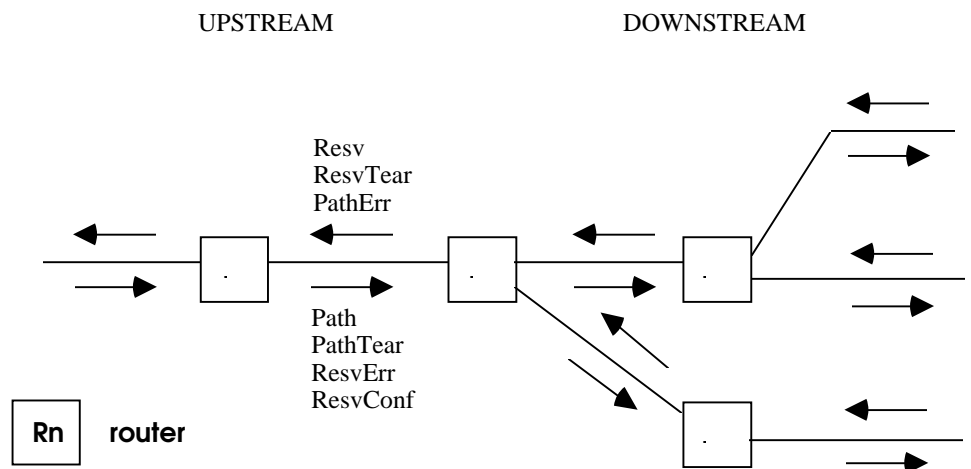


Figure 1: Direction of RSVP messages.

RSVP is not a routing protocol, it is a signaling protocol; it is merely used to reserve resources along the existing route set up by whichever underlying routing protocol is in place. Figure 1 shows an example of RSVP for a multicast Session involving one sender, S1 and three receivers, RCV1 - RCV3. The primary messages used by RSVP are the Path message which originates from the traffic sender and the Resv message which originates from the traffic receivers. The primary roles of the Path message are firstly, to install reverse routing state in each router along the path and secondly, to provide receivers with information about the characteristics of the sender traffic and end-to-end path so that they can make appropriate reservation requests. The primary role of the Resv message is to carry reservation requests to the routers along the distribution tree between receivers and senders. Returning now to Figure 1, as soon as S1 has data to send it begins periodically forwarding RSVP Path messages to the next hop, R1 down the distribution tree. RSVP messages can be transported ‘raw’ within IP datagrams using protocol number 46 although hosts without this raw I/O capability may first encapsulate the RSVP messages within a UDP header.

4.1 Reservation Styles and Merging

Associated with each reservation made at a router’s interface is a Filterspec describing the packets to which the reservation applies along with an effective Flowspec. Both the Filterspec and effective Flowspec are obtained from a merging process applied to selected Resv messages arriving on the router’s interface. The rules for merging are dependent upon the reservation Style of each Resv message as described below. In addition the router calculates the Filterspec and Flowspec of Resv messages to be sent to the previous hop(s) upstream by applying Style-dependent merging of stored reservation state. Any changes to stored reservation state that result in changes to the Resv messages to be sent upstream will cause an updated Resv message to be sent upstream immediately. Otherwise Resv messages are created

based on stored reservation state and sent upstream periodically. As for path state all reservation state is stored in routers using soft-state and consequently relies on periodic refreshes via Resv messages to prevent state timeout¹⁶. In addition just as a PathTear message exists to explicitly tear down path state, a ResvTear message exists to explicitly tear down reservation state. Currently 3 reservation Styles are permissible as described below and illustrated in Figures 2-6 where the convention Style(Filterspec{Flowspec}) is used to summarise the requests made by the Resv messages. It should be noted that the merging processes described below apply only to packets of the same Session(This is true of any RSVP process). Also merging can only occur between messages with the same reservation style. Details of the reservation styles are as follows where it is assumed that each interface I in Figures 2-4 is routable to each of the router's other interfaces.

Fixed Filter (FF) (distinct reservation and explicit sender selection)

The Filterspec of each FF reservation installed at an interface consists of a single sender only. The effective Flowspec of the reservation installed is the maximum of all FF reservation requests received¹⁷ on that interface for that particular sender. The Flowspec of the FF Resv message unicast to the previous hop of a particular sender is given by the maximum Flowspec of all reservations installed in the router for that particular sender.

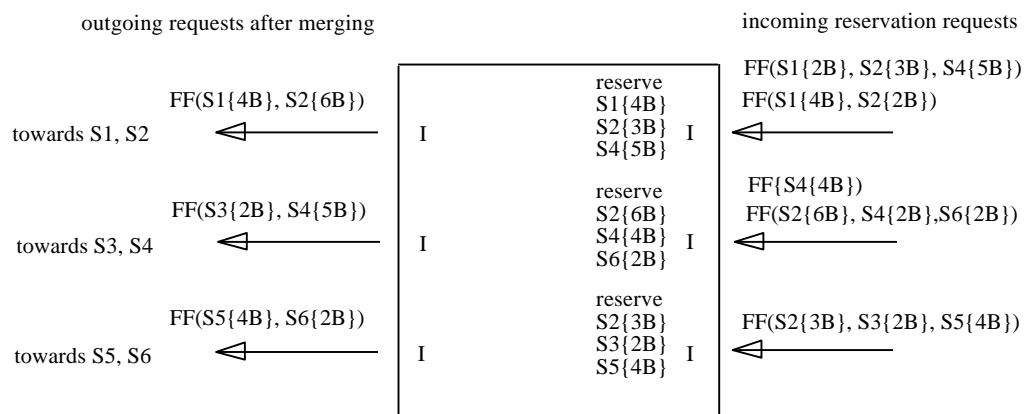


Figure 2: Fixed Filter Reservation Example.

Wildcard Filter (WF) (shared reservation and wildcard sender selection)

The Filterspec of each WF reservation installed at an interface is wildcard and matches on any sender from upstream. The effective Flowspec installed is the maximum from all WF reservation requests received on that particular interface. The

¹⁶ The periodic Resv message is necessary and sufficient to prevent reservation state timeout. Of course, if a router crashes, a Path message is necessary after reboot so that the Resv can rendezvous with it.

¹⁷ In cases where the interface connects to a shared medium LAN Resv messages from multiple next hops may be received.

Flowspec of each WF Resv message unicast to a previous hop upstream is given by the maximum Flowspec of all WF reservations installed in the router¹⁸.

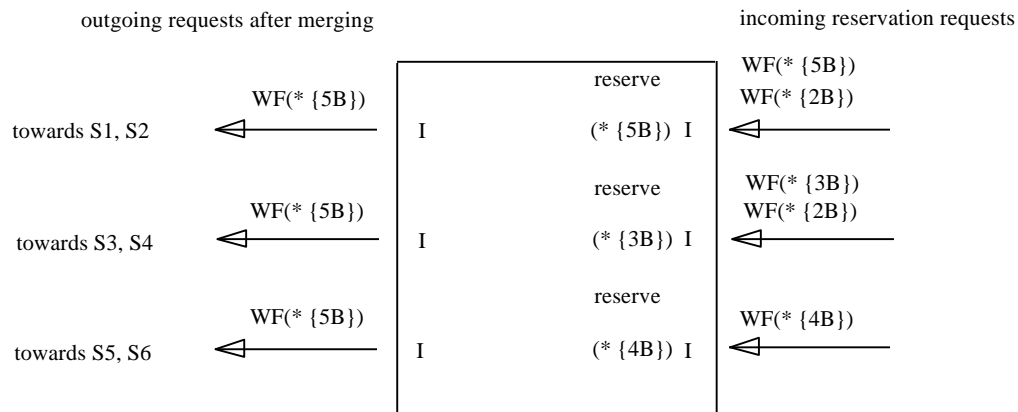


Figure 3: Wildcard Filter Reservation Example.

Shared Explicit (SE) (shared reservation and explicit sender selection)

The Filterspec of each SE reservation installed at an interface contains a specific set of senders from upstream and is obtained by taking the union of the individual Filterspecs from each SE reservation request received on that interface. The effective Flowspec installed is the maximum from all SE reservation requests received on that particular interface. The Filterspec of a SE Resv message unicast out of an interface to a previous hop upstream is the union of all senders whose previous hop is via that interface and who are contained in the Filterspec of at least one SE reservation in the router. Likewise the Flowspec of this SE Resv message is given by the maximum Flowspec of all SE reservations whose Filterspecs contain at least one sender whose previous hop is via that interface.

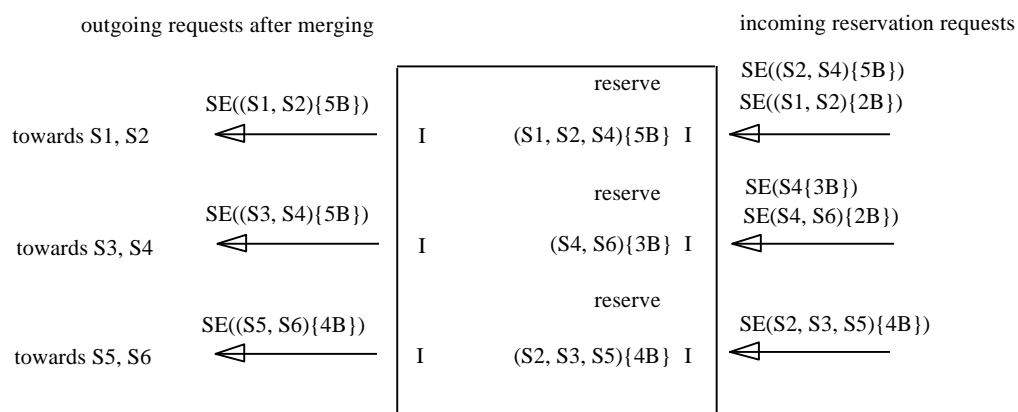


Figure 4: Shared Explicit Reservation Example.

¹⁸Strictly speaking, only WF reservations whose ‘Scope’ applies to the interface out of which the Resv message is sent are considered for this second merging process. Scope details are required for WF reservations on non-shared trees to prevent looping. Further details can be found in [3].

SE and WF styles are useful for conferencing applications where only one sender is likely to be active at once in which case reservation requests for say twice the sender bandwidth could be reserved in order to allow an amount of over-speaking.

Although RSVP is unaware of which service(Controlled-Load or Guaranteed) reservations refer to, RSVP is able to identify those points in the distribution tree that require reshaping in the event that the reservations are for Guaranteed Service as described in section 3.2. Consequently at all such points RSVP informs the traffic control mechanisms within the appropriate router accordingly although such action will only result in reshaping if the reservation is actually for Guaranteed Service.

4.2 Path Messages

Each Path message includes the following information:

- Phop, the address of the last RSVP-capable node to forward this Path message. This address is updated at every RSVP-capable router along the path.
- The Sender Template, a filter specification identifying the sender. It contains the IP address of the sender and optionally the sender port(in the case of IPv6 a flow label may be used in place of the sender port).
- The Sender Tspec defining the sender traffic characteristics.
- An optional Adspec containing OPWA information(see sections 4.4 and 4.5) which is updated at every RSVP-capable router along the path to attain end-to-end significance before being presented to receivers to enable them to calculate the level of resources that must be reserved to obtain a given end-to-end QOS.

4.3 Processing and Propagation of Path Messages by Network Routers

Each intermediate RSVP-capable router along the distribution tree intercepts Path messages and checks them for validity. If an error is detected then the router will drop the Path message and send a PathErr message upstream to inform the sender who can then take appropriate action. Assuming the Path message is valid the router does the following:

- Update the path state entry for the sender identified by the Sender Template. If no path state exists then create it. Path state includes the Sender Tspec, the address, Phop of the previous hop upstream router and optionally an Adspec. The Phop address needs to be stored in order to route Resv messages in the reverse direction up the tree. The Sender Tspec provides a ceiling to clip any inadvertently over-specified Tspecs subsequently received in Resv messages
- Set cleanup timer equal to cleanup timeout interval and restart timer.
- Associated with each path state entry is a cleanup timer, the expiration of which triggers deletion of the path state. Expiration of the timer will be prevented if a Path message for the entry is received at least once every cleanup timeout interval. This is the so-called RSVP “soft state” mechanism and ensures that state automatically times out if routing changes while subsequent Path messages install state along the new routing path. In this way the use of soft-state rather than hard-state helps to maintain much of the robustness of the initial Internet design concepts whereby all flow-related state was restricted to the end systems[4].

The router is also responsible for generating Path messages based on the stored path state and forwarding them down the routing tree making sure that for each outgoing interface the Adspec(see section 4.4) and Phop objects are updated accordingly. Path messages will be generated and forwarded whenever RSVP detects any changes to stored path state or is informed by the underlying routing protocol of a change in the set of outgoing interfaces in the data forwarding path. Otherwise, a Path message for each specific path state entry is created and forwarded every refresh period timeout interval in order to refresh downstream path state.

The refresh period timeout interval is several times smaller than the cleanup timeout interval so that occasional lost Path messages can be tolerated without triggering unnecessary deletion of path state. However it is still recommended that a minimum network bandwidth be configured for RSVP messages to protect them from congestion losses.

Although all path state would eventually timeout in the absence of any refreshes via Path messages, RSVP includes an additional message, PathTear to expedite the process. PathTear messages travel across the same path as Path messages and are used to explicitly tear down path state. PathTear messages are generated whenever a path state entry is deleted and so a PathTear message generated by a sender will result in deletion of all downstream path state for that sender. It is recommended that senders do this as soon as they leave the communications session. Also, deletion of any path state entry triggers deletion of any dependent reservation state(see section 4.1).

4.4 Adspec

The Adspec is an optional object that the sender may include in its generated Path messages in order to advertise to receivers the characteristics of the end to end communications path. This information can be used by receivers to determine the level of reservation required in order to achieve their desired end to end QOS. The Adspec consists of a message header, a Default General Parameters fragment and at least one of the following; Guaranteed Service fragment, Controlled-Load Service fragment. Omission of either the Guaranteed or Controlled-Load Service fragment is an indication to receivers that the omitted service is not available. This feature can be used in a multicast Session to force all receivers to select the same service. At present RSVP does not accommodate heterogeneity of services between receivers within a given multicast Session, although, within the same service model, the parameters may differ for receivers in the same session, so the core objective of supporting heterogeneity is *mainly* met.

The Default General Parameters fragment includes the following fields which are updated at each RSVP-capable router along the path in order to present end-to-end values to the receivers.

- Minimum Path Latency (summation of individual link latencies). This parameter represents the end-to-end latency in the absence of any queuing delay. In the case of Guaranteed Service, receivers can add this value to the bounded end-to-end queuing delay to obtain the overall bounded end-to-end delay.
- Path Bandwidth (minimum of individual link bandwidths along the path)

- Global Break bit - This bit is cleared when the Adspec is created by the sender. Encountering any routers that do not support RSVP will result in this bit being set to one in order to inform the receiver that the Adspec may be invalid.
- Integrated Services(IS) Hop count - incremented by one at every RSVP/IS-capable router along the path.
- PathMTU - Path Maximum Transmission Unit (minimum of MTUs of individual links along the path).

Correct functioning of IETF Integrated Services requires that packets of a data flow are never fragmented¹⁹. This also means that the value of M in the Tspec of a reservation request must never exceed the MTU²⁰ of any link to which the reservation request applies to. A receiver can ensure that this requirement is met by setting the value of M in the Tspec of it's reservation request to the minimum of the PathMTU values received in 'relevant' Path messages. The value of M in each generated reservation request may be further reduced on the way to each sender if merging of Resv messages occurs(see section 4.2). The minimum value of M from the Tspec of each Resv message²¹ received by the sender should then be used by the sending application as the upper limit on the size of packets to receive special QOS. In this way fragmentation of these packets will never occur. It is worth noting that [13] recommends that the value of M in the Sender Tspec, which has played no part in the above MTU negotiation process, should be set equal to the maximum packet size that the sender is capable of generating rather than what it is currently sending.

The Guaranteed Service fragment of the Adspec includes the following fields which are updated at each RSVP-capable router along the path in order to present end-to-end values to the receivers.

- Ctot - end to end composed²² value for C.
- Dtot - end to end composed value for D.
- CSum - composed value for C since last reshaping point.
- DSum - composed value for D since last reshaping point (CSum and Dsum values are used by reshaping processes at certain points along the distribution tree).
- Guaranteed Service Break Bit - This bit is cleared when the Adspec is created by the sender. Encountering any routers that do support RSVP/IS but do NOT support Guaranteed Service will result in this bit being set to one in order to inform the receiver that the Adspec may be invalid and the service cannot be guaranteed.
- Guaranteed Service General Parameters Headers/Values - These are optional but if any are included then each one overrides the corresponding value given in the Default General Parameters fragment as far as a receiver wishing to make a Guaranteed Service reservation is concerned. These override parameters could for example be added by routers along the path that have certain service-specific requirements. For example a router may have been configured by network management so that Guaranteed Service reservations can only take up a certain amount, Bgs of the outgoing link bandwidth. Consequently if the Default Path bandwidth value in the Adspec to be sent out of this interface is greater than Bgs

¹⁹ It might be possible to devise a scheme to support QoS for fragmented traffic, but the key problem of how loss of fragment results in loss of overall datagram is hard to work around!

²⁰ MTU Discovery can be employed to avoid this.

²¹ In cases where the last hop to a sender is a shared medium LAN the sender may receive Resv messages across the same interface from multiple next hop routers.

²² Composed, as explained above, from the sum of each router/hop's estimate of the error.

then a Guaranteed Service Specific Path bandwidth header and value equal to Bgs may be included in the Adspec. As for Default General Parameters, any Service-Specific General Parameters must be updated at each RSVP hop.

The Controlled-Load Service fragment of the Adspec includes the following fields which are updated at each RSVP-capable router along the path in order to present end-to-end values to the receivers.

- Controlled-Load Service Break Bit - This bit is cleared when the Adspec is created by the sender. Encountering any routers that do support RSVP/IS but do NOT support Controlled-Load will result in this bit being set to one in order to inform the receiver that the Adspec may be invalid and the service cannot be guaranteed.
- Controlled-Load Service General Parameters Headers/Values - As for the Guaranteed Service fragment, override Service-Specific General Parameters may be added to the Controlled-Load Service fragment.

4.5 Making a Reservation using One Pass with Advertising (OPWA)

One Path With Advertising(OPWA), refers to the reservation model for the case where the sender includes an Adspec in its Path messages to enable the receiver to determine the end-to-end service that will result from a given reservation request. If the sender omits the Adspec from its Path messages then the reservation model is referred to simply as One Pass in which case there is no easy way for the receiver to determine the resulting end-to-end service. The objective of this aspect of the RSVP reservation model, both for One Pass and One Pass with Advertising is to minimise the latency (in terms of number of handshakes between senders and recipients) before a reservation is in place. Here we consider the OPWA case.

Let us assume that the sender omits the Controlled-Load Service data fragment from the Adspec thereby restricting each receiver to reservation of Guaranteed Service only. Upon receiving Path messages the receiver extracts the following parameters from the Sender Tspec contained therein: r, b, p, m. In addition the following are extracted from the Adspec: Minimum Path Latency, Ctot, Dtot, PathMTU, Path Bandwidth.

In a way similar to incremental route calculation, OPWA permits incremental accumulation of the delay for a reservation. The required bound on end-to-end queuing delay, Qdelreq is now calculated by subtracting the Minimum Path Latency from the value of end-to-end delay required by the receiver's application. Typically, the receiver would then perform an initial check by evaluating equation (2) for R equal to the peak rate, p. If the resultant delay was greater than or equal to Qdelreq then equation (2) would be used for calculation of the minimum value of R necessary to satisfy Qdelreq. Otherwise equation (1) would be used for this purpose. This minimum value of R is then obtained by inserting Qdelreq into either equation (1) or (2) along with M(given by PathMTU), Ctot, Dtot, r, b, p, as appropriate. If the obtained value of R exceeds the Path Bandwidth value as obtained from the Adspec of the received Path message then it must be reduced accordingly. The receiver can now create a reservation specification, Rspec comprising firstly the calculated value, R of bandwidth to be reserved in each router, and secondly a Slack Term that is

initialised to zero²³. The Rspec can now be used in the creation of a Resv message which also includes the following:

- An indication of the reservation style which can be FF, SE or WF (see section 4.1)
- A filter specification, Filterspec (omitted for the case of WF reservation style). This is used to identify the sender(s) and the format is identical to that of the Sender Template in a Path message.
- A flow specification, Flowspec comprising the Rspec and a traffic specification, Tspec. Tspec is usually set equal to the Sender Tspec except M will be given by PathMTU obtained from the received Adspec.
- Optionally a reservation confirm object, ResvConf containing the IP address of the receiver. If present this object indicates that the node accepting this reservation request at which propagation of the message up the distribution tree finishes should return a ResvConf message to the receiver to indicate that there is a high probability²⁴ that the end-to-end reservation has been successfully installed.

The Resv message is now sent to the previous hop upstream as obtained from the stored path state. Upon reaching the next upstream router the Resv messages can be merged with other Resv messages arriving on the same interface according to certain rules as described in section 4.1 to obtain an effective Flowspec and Filterspec. The following action is then taken.

- The effective Flowspec is passed to the traffic control module within the router which applies both admission control and policy control to determine whether the reservation can be accepted. Admission control is concerned solely about whether enough capacity exists to satisfy the request while policy control also takes into account any additional factors that need to be considered (e.g. certain policies may limit a users reserved bandwidth even if spare bandwidth exists).
- If the reservation attempt is denied then any existing reservations are left unaltered and the router must send a ResvErr message downstream.
- If the reservation request is accepted then reservation state is set up in accordance with the effective Flowspec and Filterspec as described in section 4.1. In accepting the request it may be permissible to alter the Rspec associated with the reservation from (Rin, Sin) to (Rout, Sout) in accordance with the rules described in section 4.6. The resultant reservation may then be merged with other reservations in accordance with the rules in section 4.1 to obtain a new Resv message that is sent to the next router upstream, the address of which is obtained from the stored path state.

4.6 Slack term

When a receiver generates an Rspec for a Resv message to be sent for a Guaranteed Service reservation request it must include a Slack Term, S(ms) as well as the amount of bandwidth, R to be installed in each router along the path. S represents the amount by which the end-to-end delay bound will be below the end-to-end delay required by

²³In some cases even with R set to the minimum permissible value of r the resultant end-to-end queuing delay as given by eqs (1) and (2) will still be less than Qdelreq in which case the difference can be represented in a non-zero slack term. In addition there are other scenarios explained in section 4.6 in which the slack term may not be initialised to zero.

²⁴In practice there are certain scenarios in which a ResvConf message might be received by a receiver only for the request to be rejected shortly afterwards.

the application assuming each router along the path reserves R bandwidth according to the Guaranteed Service fluid approximation. Inclusion of a non-zero Slack Term offers the individual routers greater flexibility in making their local reservations. In certain circumstances this greater flexibility could increase the chance of an end-to-end reservation being successful. Some routers have deadline based schedulers that decouple rate and delay guarantees. Such a scheduler may sometimes be unable to meet it's deadline requirement for Guaranteed Service in which case it might still be able to accept the reservation providing the Slack Term is at least as large as the excess delay. The excess delay would then be subtracted from the Slack Term before unicasting the Resv message to the previous hop upstream. Similarly a rate based scheduler might be able to admit a reservation request by reserving less than the requested bandwidth and unicasting the reduced reservation request to a previous hop upstream provided it could extract enough slack. Any router using available slack to reduce it's reservation must conform to the rules in equation (3) to ensure that the end-to-end delay bound remains satisfied.

$$S_{out} + \frac{b}{R_{out}} + \frac{C_{toti}}{R_{out}} \leq S_{in} + \frac{b}{R_{in}} + \frac{C_{toti}}{R_{in}} \quad r \leq R_{out} \leq R_{in} \quad (3)$$

where:

C_{toti} is the cumulative sum of the error terms, C for all the routers that are upstream of, and including the current element, i.

(R_{in}, S_{in}) is the reservation request received by router, i.

(R_{out}, S_{out}) is the modified reservation request unicast to the previous hop router upstream.

An example of how intelligent use of the Slack Term can increase the probability of an end-to-end reservation request being accepted is illustrated in Figures 5 and 6. Suppose the token bucket rate (as defined earlier) of the data to be sent is 1.5Mb/s and the receiver has calculated from the Tspec and Adspec parameters in received Path messages that the desired end-to-end delay can be achieved by a reservation of $(R=2.5\text{Mb/s}, S=0)$ which is then requested in Figure 5. However because R3 only has 2Mb/s of unused bandwidth and there is no slack available the reservation is denied. In Figure 6 the reservation is increased to $R=3\text{Mb/s}$ and the amount by which such a reservation would be within the required delay bound is put in the Slack Term $(S>0)$. R5 and R6 reserve the requested 3Mb/s. R3 can only reserve a value of 2Mb/s which if used as the new reservation value in the propagated Resv message will cause an increase in the end to end delay bound. R3 can calculate this increase, d_i and if it is less than the value of the Slack Term, S_1 in the received Resv message then the request can be accepted and a reservation of 2Mb/s installed in R3. R3 will then set the Rspec in the Resv message to $(R=2\text{Mb/s}, S_2=S_1-d_i)$ before unicasting it to the next hop upstream which results in R2 and R1 also reserving 2Mb/s. The end-to-end delay bound of the reserved path is now no greater than for a reservation of 2.5Mb/s in every router if that were possible.

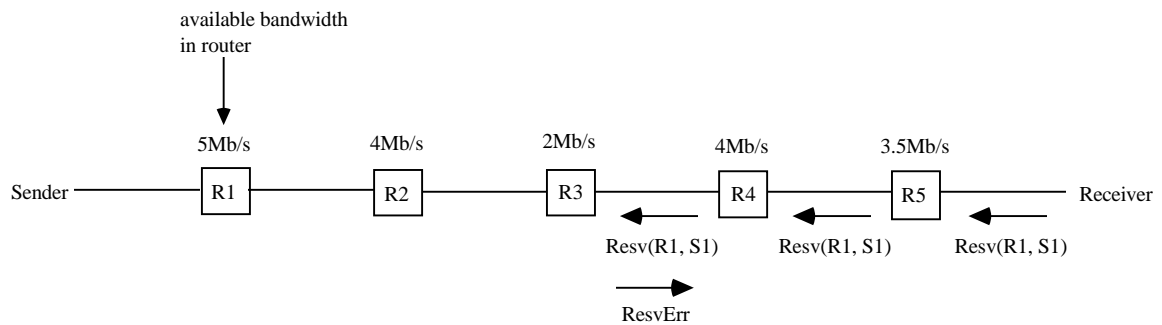


Figure 5 $R=2.5\text{Mb/s}$, $S1=0$. Reservation request denied.

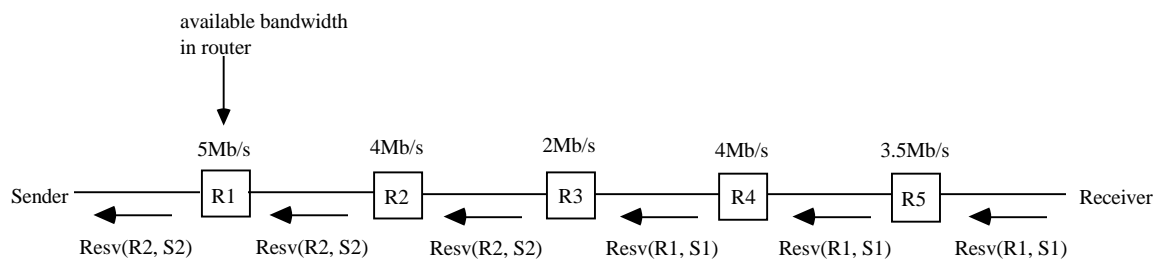


Figure 6 $R1=3\text{Mb/s}$, $S1>0$, $R2=2\text{Mb/s}$, $S2<S1$. Reservation accepted.

5.0 Summary

In this paper we have looked at the evolution of the Internet from Best Effort, FIFO, Destination Routed, Unicast network, to a multi-service, QoS Routed, multicast capable system.

We have seen that very detailed progress has been made in that if supported by the routers along an end-to-end data path, RSVP can permit end systems to request Integrated Services that provide end applications with enhanced QoS commitments over conventional best-effort delivery. RSVP can be used by end applications to select and invoke the appropriate class and QoS level. In addition if the OPWA reservation model is used with RSVP then the requesting application is able to determine the resultant end-to-end QoS in advance of making the reservation.

Without RSVP, a fall back service of best effort is still available from the unused capacity. In the near future, some research needs to be carried out in a number of areas:

1. Accounting and Billing needs to be integrated into the model in a scaleable way.
2. Aggregation of non-specifically related reservations (or flows) would be useful - in the same way that ATM provides Virtual Paths as well as Virtual Circuits, we might like to build virtual private Internets using, for example, the address aggregation mechanism CIDR, to be used within a reservation (the extension has been proposed in the RSVP working group to allow the generalised port to be accompanied by masks, in the same way that routing protocols distribute updates with masks).
3. Authentication of users of RSVP is clearly essential if we are to incur bills when we use it.

4. The usage accounting model must accommodate mirror servers in some way.
5. Some scheme to permit settlements or something akin to them will need to be evolved to allow deployment of RSVP and Integrated services across paths that entail more than a single *Intranet* or commercial Internet Service Provider.
6. Lastly, simply experience of using a multiservice networks is needed to see which pieces of this complex system are really used frequently, since it is not at all clear that the entire edifice is all either necessary, or sufficient.

The current status of RSVP in the IETF is that it is undergoing standardisation as of the writing of this paper.

6.0 Acknowledgements

The authors would like to acknowledge the generous support of British Telecommunications, and the feedback from the anonymous reviewers, as well as the many helpful comments and suggestions from Bur Goode.

7.0 References

- [1] F. Baker, R. Guerin and D. Kandlur. Specification of Committed Rate Quality of Service, Internet Draft, June 1996, <ftp://ds.internic.net/internet-drafts/draft-ietf-intserv-commit-rate-svc-00.txt>.
- [2] R. Braden, D. Clark and S. Shenker. Integrated Services in the Internet Architecture: an Overview, Request for Comments, July 1994, <ftp://ds.internic.net/rfc/rfc1633.txt>.
- [3] R. Braden, L. Zhang, S. Berson, S. Herzog and S. Jamin. Resource Reservation Protocol (RSVP) - Version 1 Functional Specification, August 12, 1996. Available via <http://www.ietf.org/html.charters/intserv-charter.html>.
- [4] D. Clark. The Design Philosophy of the DARPA Internet Protocols, ACM SIGCOMM '88, August 1988.
- [5] L. Delgrossi and L. Berger. Internet Stream Protocol Version 2 (ST2) Protocol Specification - Version ST2+, Request for Comments. August 1995, RFC1819, <ftp://ds.internic.net/rfc/rfc1190.txt>.
- [6] J. Heinanen. Protected Best Effort Service, Internet Draft, February 1996, <ftp://ds.internic.net/internet-drafts/draft-heinanen-pbe-svc-01.txt>.
- [7] D. Mitzel, D. Estrin, S. Shenker, L. Zhang. An Architectural Comparison of ST-II and RSVP, Proceedings of Infocom '94, <http://www.isi.edu/div7/rsvp/pub.html>.
- [8] A. Parekh and R. Gallager. A Generalized Processor Sharing Approach to Flow Control-The Single Node Case, In IEEE/ACM Transactions on Networking, 1(3), pp. 366-357, 1993.
- [9] A. Parekh and R. Gallager. A Generalized Processor Sharing Approach to Flow Control-The Multiple Node Case, In IEEE/ACM Transactions on Networking, 2(2), pp. 137-150, 1996.
- [10] S. Schenker, C.Partridge, R.Guerin. Specification of Guaranteed Quality of Service, Internet Draft, August 1996, <ftp://ds.internic.net/internet-drafts/draft-ietf-intserv-guaranteed-svc-06.txt>.
- [11] C. Topolcic. Experimental Internet Stream Protocol, Version 2 (ST-II), Request for Comments, October 1990, RFC1190, <ftp://ds.internic.net/rfc/rfc1190.txt>.
- [12] J. Wroclawski. Specification of the Controlled-Load Network Element Service, Internet Draft, August 1996, <ftp://ds.internic.net/internet-drafts/draft-ietf-intserv-ctrl-load-svc-03.txt>.
- [13] J. Wroclawski. The Use of RSVP with IETF Integrated Services, Internet Draft, August 1996, <ftp://ds.internic.net/internet-drafts/draft-ietf-intserv-rsvp-use-00.txt>.
- [14] IETF Home Page, <http://www.ietf.cnri.reston.va.us>.
- [15] Integrated Services Charter, <http://www.ietf.org/html.charters/intserv-charter.html>.
- [16] Integrated Services over Specific Link Layers (issl) Charter, <http://www.ietf.org/html.charters/issl-charter.html>.
- [17] Multicast Routing, Steve Deering, PhD Thesis, Stanford, 1988

- [18] Scalable Compression And Transmission of internet Multicast Video, Steven McCanne, Report No. UCB/CSD 96-928, Dec 1996. PhD Thesis
- [19] The Robust Audio Tool, Colin Perkins, Vicky Hardman, Angela Sasse, to appear, in Communications of the ACM, 1997.
- [20] Random Early Detection Gateways, Floyd, S., and Jacobson, V., Random Early Detection gateways for Congestion Avoidance, IEEE/ACM Transactions on Networking, V.1 N.4, August 1993, pp. 397-413.
- [21] Wakeman, Ghosh, Crowcroft, Jacobson and Floyd, "Implementing Real Time Packet Forwarding Policies using Streams", in Proceedings of the Usenix Conference, New Orleans, January 1995.
- [22] Pricing in Computer Networks: Reshaping the Research Agenda, S. Shenker, D.Clark, D.Estrin, S.herzog,in Internet Economics, Lee W. McKnight and Joseph P.Bailey, MIT Press,1996
- [23] IEEE Standard for Information Technology - Protocols for Distributed Interactive Simulation Applications, IEEE Std. 1278-1993.