ABSTRACT

The growing use of multimedia communication applications with specific bandwidth and real-time delivery requirements has created the need for an integrated services Internet in which traditional best-effort datagram delivery can coexist with additional enhanced quality of service (QoS) delivery classes. Such classes provide data flows with QoS commitments with regard to bandwidth, packet loss, and delay through the reservation of network resources along the data path, which can be done using the Resource Reservation Protocol (RSVP). This article is a tutorial on how RSVP can be used by end applications to ensure that they receive the end-to-end QoS that they require.

RSVP and Integrated Services in the Internet: A Tutorial

Paul P. White, University College London

The current Internet consists of a multitude of networks built from various link-layer technologies and relies on the Internet Protocol (IP) to interwork between them. IP makes no assumptions about the underlying protocol stacks and offers an unreliable, connectionless network-layer service that is subject to packet loss, reordering, and packet duplication, all of which, together with queuing delay in router buffers, will increase with network load. Because of the lack of any firm guarantees, the traditional IP delivery model is often referred to as "best-effort" with an additional higher-layer end-to-end protocol such as the Transmission Control Protocol (TCP) required to provide end-to-end reliability. TCP does this through the use of such mechanisms as packet retransmission, which further adds to the overall information transfer delay.

For traditional non-real-time Internet traffic such as File Transfer Protocol (FTP) data, the best-effort delivery model of IP has not been a problem. However, as we move further into the age of multimedia communications, many real-time applications are being developed that are delay-sensitive to the point where the best-effort delivery model of IP can be inadequate even under modest network loads. Although the problem has been alleviated somewhat through making certain applications adaptive to network load where possible, there is still a firm need to provide many applications with additional service classes offering enhanced quality of service (QoS) with regard to bandwidth, packet queuing delay, and loss. These additional enhanced QoS delivery classes would supplement the best-effort delivery service in what could be described as an integrated services Internet [1].

IETF Integrated Services

In response to the growing demand for an integrated services Internet, the Internet Engineering Task Force (IETF) [2] set up an Integrated Services (intserv) Working Group [3], which has since defined several service classes that, if supported by the routers traversed by a data flow,[1] can provide the data flow with certain QoS commitments. In 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 the third section. The requests dictate the level of resources (e.g., bandwidth, buffer space) that must be reserved along with the transmission scheduling behavior 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 asynchronous transfer mode (ATM) or certain types of local area network (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 [4]. 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

[1] *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 is identified by source address and port number, while its protocol type must be the same as for the session.*

 0163-6804/97/$10.00 © 1997 IEEE

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 envelope 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 maximum transmission unit (MTU) of the link, 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 flow, and provided the data flow conforms to the traffic envelope supplied in the request. Service-specific policing and traffic reshaping actions, as described in the next two subsections, will be employed within the network to ensure that nonconforming data flows do not affect the QoS commitments for behaving data flows. The IETF has considered various QoS classes such as [5–8], although to date only two of these, Guaranteed Service [7] and Controlled-Load Service [8], have been formally specified for use with RSVP [9].

GUARANTEED SERVICE

Guaranteed Service [7] 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 use "playback" buffers and are intolerant of any datagram arriving after their playback time.

Each router characterizes 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 [10, 11] 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 bound by b/R provided $R \geq r$. To allow for deviations from this perfect fluid model in the router's approximation,[2] two error terms, C and D, are introduced; 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 packetization effect of the flow needs to be taken into account by considering the maximum packet size, M. These additional factors result in a more precise bound on the end-to-end queuing delay as follows:

$$Q_{\text{delayend2end}} = \frac{(b-M)(p-R)}{R(p-r)} + \frac{(M+C_{\text{tot}})}{R} + D_{\text{tot}} \quad (\text{case } p > R \geq r) \tag{1}$$

$$Q_{\text{delayend2end}} = \frac{(M+C_{\text{tot}})}{R} + D_{\text{tot}} \quad (\text{case } R \geq p \geq r) \tag{2}$$

where 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 order for a router to invoke guaranteed service for a specific data flow, it needs to be informed of the traffic characteristics, Tspec, of the flow along with the reservation characteristics, Rspec. Furthermore, to enable the router to calculate sufficient local resources to guarantee a lossless service 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 reshaping point (see below).

Tspec parameters:

- p = peak rate of flow (bytes/s)
- b = bucket depth (bytes)
- r = token bucket rate (bytes/s)
- m = minimum policed unit (bytes)[3]
- M = maximum datagram size (bytes)

Rspec parameters:

- R = bandwidth, i.e., service rate (bytes/s)
- S = slack term (ms)

Guaranteed service traffic must be policed at the network access points to ensure conformance to the Tspec. The usual enforcement policy is to forward nonconforming packets as best-effort datagrams;[4] if and when a marking facility becomes available, these nonconforming datagrams should be marked to ensure that they are treated as best-effort datagrams at all subsequent routers.

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. Reshaping 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 two cases.

First, at branch points in the distribution tree where the reserved Tspecs of the outgoing branches are not the same, the reserved Tspec of the incoming branch is given by the "maximum"[5] of the reserved Tspecs on each of the outgoing branches. Consequently, some of the outgoing branches will have a reserved Tspec which is less than the reserved Tspec of the incoming branch; so it is possible that, in the absence of reshaping, traffic which 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.

Second, at merge points in the distribution tree for sources sharing the same reservation, 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.

CONTROLLED-LOAD SERVICE

Unlike guaranteed service, controlled-load service [7] provides no firm quantitative guarantees. A Tspec 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

[2] *Among other things, the router's approximation must take account of the medium-dependent behavior of the link layer of the data forwarding path.*

[3] *Policing will treat any IP datagram less than size* m *as being size* m.

[4] *Action with regard to nonconforming datagrams should be configurable to allow for situations such as traffic sharing where the preferred action might be to discard nonconforming datagrams. This configuration requirement also applies to reshaping.*

[5] *Maximum according to rules defined in [12].*

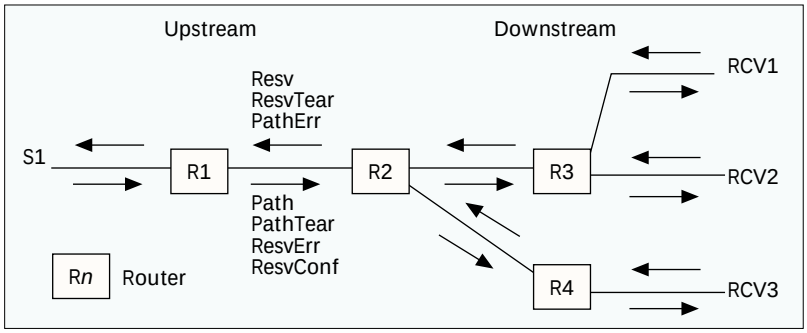

Figure 1. *Direction of RSVP messages.*

include the peak rate parameter. If the flow is accepted for controlled-load service, 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 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. 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.

Routers implementing the controlled-load service must check for conformance of controlled-load data flows to their appropriate reserved Tspecs. Any nonconforming controlled-load data flows must not be allowed to affect the QoS offered to conforming controlled-load 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 nonconforming controlled-load data flow as possible. This might be done by dividing the packets into conforming and nonconforming groups and forwarding the nonconforming group on a best-effort basis. Alternatively, the router may choose to degrade the QoS of all packets of a nonconforming controlled-load data flow equally.

Resource Reservation Protocol

The Resource Reservation Protocol (RSVP) [12] 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 previously. It is worth noting that RSVP is not the only IP reservation protocol that has been designed for this purpose. Others include ST-II [13] and ST-II+ [14], which incidently contain some interesting architectural differences from RSVP, such as the use of hard-state and sender-initiated[6] reservations rather than soft-state[7] and receiver-initiated reservations, as in RSVP. However, for the rest of this tutorial 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 [15].

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; therefore, 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, we concentrate on the more complicated multicast case. Also, we do not discuss the security issues of RSVP or any billing that may be necessary to exert backpressure on the use of reservations.

RSVP is not a routing 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 first to install reverse routing state in each router along the path, and second 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 Fig. 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 input/output (I/O) capability may first encapsulate the RSVP messages within a UDP header.

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 One Pass With Advertising (OPWA) information 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.

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, 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, 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

[6] *ST-II+ permits both sender and receiver-initiated reservations; ST-II permits sender-initiated reservations only.*

[7] *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.*

the tree. The Sender Tspec provides a ceiling to clip any inadvertently overspecified 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, which 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 [16].

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 the next subsection) 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, 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.

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 a Guaranteed Service fragment and 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).

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 to receive the special QoS are never fragmented. This also means that the value of M in the Tspec of a reservation request must never exceed the MTU of any link to which the reservation request applies. A receiver can ensure that this requirement is met by setting the value of M in the Tspec of its reservation request to the minimum of the PathMTU values received in "relevant" Path messages. A Path message is relevant if it originated from a sender that is captured in the intended reservation request in accordance with the reservation styles described later. 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. The minimum value of M from the Tspec of each Resv message[8] 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 [9] 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 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:
- C_{tot} — end-to-end composed value for C
- D_{tot} — end-to-end composed value for D
- C_{Sum} — composed value for C since last reshaping point
- D_{Sum} —composed value for D since last reshaping point (C_{Sum} and D_{Sum} 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 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, 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, B_{gs}, of the outgoing link bandwidth. Consequently, if the Default Path bandwidth value in the Adspec to be sent

[8] *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.*

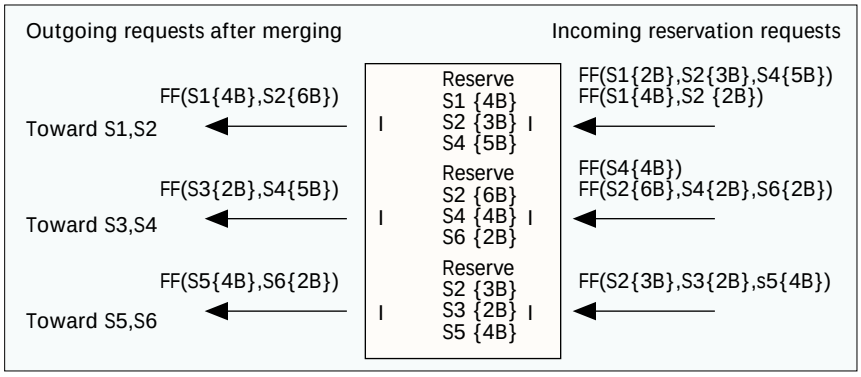

Figure 2. *Fixed filter reservation example.*

out of this interface is greater than B_{gs}, then a Guaranteed Service Specific Path bandwidth header and value equal to B_{gs} 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 support RSVP/IS but do *not* support controlled-load 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.
- 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.

MAKING A RESERVATION USING OPWA

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, 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. 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, C_{tot}, D_{tot}, PathMTU, and path bandwidth.

The required bound on end-to-end queuing delay, Q_{delreq}, 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 Eq. 2 for R equal to peak rate p. If the resultant delay was greater than or equal to Q_{delreq}, Eq. 2 would be used for calculation of the minimum value of R necessary to satisfy Q_{delreq}; otherwise, Eq. 1 would be used for this purpose. This minimum value of R is then obtained by inserting Q_{delreq} into either Eq. 1 or 2 along with M (given by PathMTU), C_{tot}, D_{tot}, r, b, and p, as appropriate. If the obtained value of R exceeds the path bandwidth value as obtained from the Adspec of the received Path message, it must be reduced accordingly. The receiver can now create a reservation specification, Rspec, comprising first the calculated value R of bandwidth to be reserved in each router, and second a slack term that is initialized to zero.[9] 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 the next subsection).
- A filter specification, Filterspec (omitted for the 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 probability[10] 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 message can be merged with other Resv messages arriving on the same interface, according to certain rules as described in the next subsection, to obtain an effective Flowspec and Filterspec. The following actions are 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

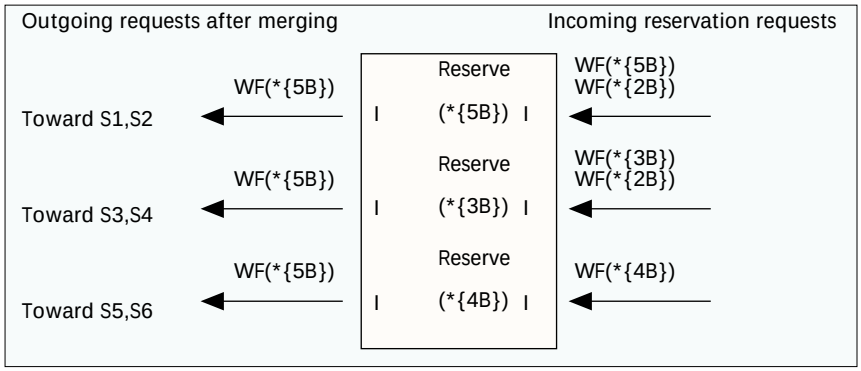

Figure 3. *Wildcard filter reservation example.*

[9] *In 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* Q_{delreq}*, in which case the difference can be represented in a nonzero slack term. In addition, there are other scenarios explained later in which the slack term may not be initialized to zero.*

[10] *In 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.*

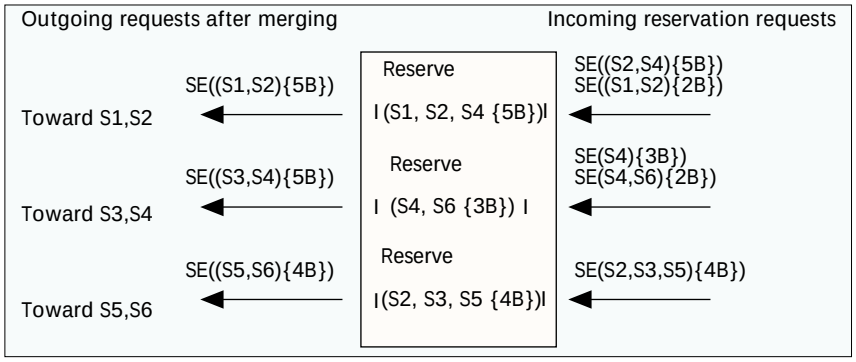

Figure 4. *Shared explicit reservation example.*

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 user's reserved bandwidth even if spare bandwidth exists).

- If the reservation attempt is denied, any existing reservations are left unaltered, and the router must send a `ResvErr` message downstream.
- If the reservation request is accepted, reservation state is set up in accordance with the effective `Flowspec` and `Filterspec` as described in the next subsection. In accepting the request it may be permissible to alter the `Rspec` associated with the reservation from (R_{in}, S_{in}) to (R_{out}, S_{out}) in accordance with the rules described later. The resultant reservation may then be merged with other reservations in accordance with the rules in the next subsection to obtain a new `Resv` message which is sent to the next router upstream, the address of which is obtained from the stored path state.

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 on 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 timeout. 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 three reservation styles are permissible, as described below and illustrated in Figs. 2–4 where the convention style (`Filterspec{Flowspec}`) is used to summarize 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 Figs. 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 received[11] 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.

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 `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 router.[12]

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 `Filterspec`s 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 an 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.

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 overspeaking.

Although RSVP is unaware of to which service (controlled-load or guaranteed) reservations refer, 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 previously. 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.

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 the application, assuming each router along the path reserves *R* bandwidth according to the guaranteed service fluid approximation. Inclusion of a nonzero slack term offers the individual routers

[11] *In cases where the interface connects to a shared-medium LAN,* `Resv` *messages from multiple next hops may be received.*

[12] *Strictly 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 nonshared trees to prevent looping. Further details can be found in [12].*

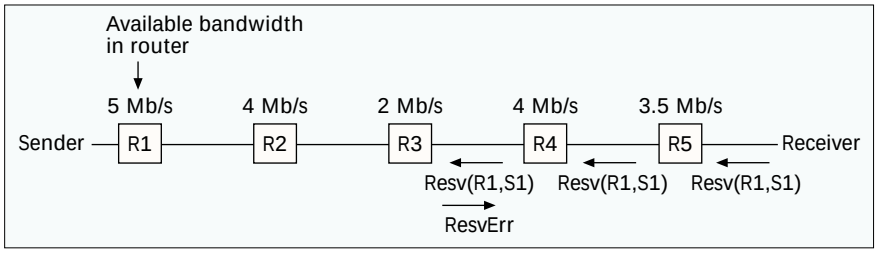

Figure 5. R = *2.5 Mb/s,* S*1 = 0. Reservation request denied.*

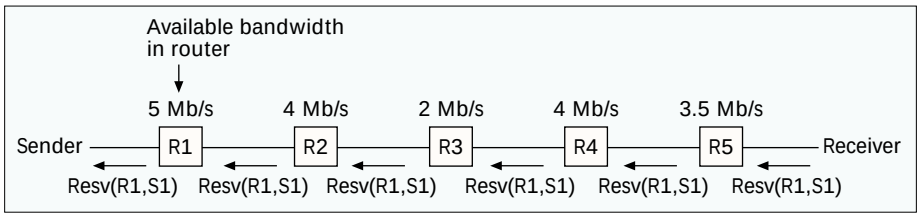

Figure 6. R*1 = 3 Mb/s,* S*1 > 0,* R*2 = 2 Mb/s,* S*2 <* S*1. Reservation accepted.*

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 its deadline requirement for guaranteed service, in which case it might still be able to accept the reservation, provided 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 its reservation must conform to the rules in Eq. 3 to ensure that the end-to-end delay bound remains satisfied.

$$S_{\text{out}} + \frac{b}{R_{\text{out}}} + \frac{C_{\text{tot}\,i}}{R_{\text{out}}} \le S_{\text{in}} + \frac{b}{R_{\text{in}}} + \frac{C_{\text{tot}\,i}}{R_{\text{in}}} \qquad r \le R_{\text{out}} \le R_{\text{in}} \quad (3)$$

where $C_{\text{tot}\,i}$ 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 Figs. 5 and 6. Suppose the token bucket rate of the data to be sent is 1.5 Mb/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 Mb/s, S = 0), which is then requested in Fig. 5. However, because $R3$ only has 2 Mb/s of unused bandwidth and there is no slack available, the reservation is denied. In Fig. 6 the reservation is increased to R = 3 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 3 Mb/s. $R3$ can only reserve a value of 2 Mb/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, $S1$, in the received Resv message, the request can be accepted and a reservation of 2 Mb/s installed in $R3$. $R3$ will then set the Rspec in the Resv message to (R = 2 Mb/s, $S2 = S1 - d_i$) before unicasting it to the next hop upstream, which results in $R2$ and $R1$ also reserving 2 Mb/s.The end-to-end delay bound of the reserved path is now no greater than for a reservation of 2.5 Mb/s in every router if that were possible.

SUMMARY

In this tutorial we have looked at the controlled-load and guaranteed service classes that, if supported by the routers along an end-to-end data path, can 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, the requesting application is able to determine the resultant end-to-end QoS in advance of making the reservation.

REFERENCES

[1] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: An Overview," RFC, July 1994; available at ftp://ds.internic.net/rfc/rfc1633.txt.

[2] IETF home page, http://www.ietf.cnri.reston.va.us.

[3] Integrated Services Charter, http://www.ietf.org/html.charters/intserv-charter.html.

[4] "Integrated Services over Specific Link Layers (issl) Charter," http://www.ietf.org/html.charters/issl-charter.html.

[5] 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.

[6] J. Heinanen, "Protected Best Effort Service," Internet Draft, Feb. 1996, ftp://ds.internic.net/internet-drafts/draft-heinanen-pbe-svc-01.txt.

[7] S. Schenker, C. Partridge, and R. Guerin, "Specification of Guaranteed Quality of Service," Internet Draft, Aug. 1996, ftp://ds.internic.net/internet-drafts/ draft-ietf-intserv-guaranteed-svc-06.txt.

[8] J. Wroclawski, "Specification of the Controlled-Load Network Element Service," Internet Draft, Aug. 1996, ftp://ds.internic.net/internet-drafts/ draft-ietf-intserv-ctrl-load-svc-03.txt.

[9] J. Wroclaski, The Use of RSVP with IETF Integrated Services, Internet Draft, Aug. 1996, ftp://ds.internic.net/internet-drafts/draft-ietf-intserv-rsvp-use-00.txt.

[10] A. Parekh and R. Gallagher, "A Generalized Processor Sharing Approach to Flow Control — The Single Node Case," *IEEE/ACM Trans. Networking*, vol. 1, no. 3, 1993, pp. 366–57.

[11] A. Parekh and R. Gallagher, "A Generalized Processor Sharing Approach to Flow Control — The Multiple Node Case," *IEEE/ACM Trans. Networking*, vol. 2, no. 2, 1996, pp. 137–50.

[12]R. Braden *et al.*, "Resource Reservation Protocol (RSVP) — Version 1 Functional Specification," Aug. 12, 1996. Available via http://www.ietf.org/html.charters/intserv-charter.html.

[13] C. Topolcic, "Experimental Internet Stream Protocol, Version 2 (ST-II)," RFC1190, Oct. 1990, ftp://ds.internic.net/rfc/rfc1190. txt.

[14] L. Delgrossi and L. Berger, "Internet Stream Protocol Version 2 (ST2) Protocol Specification — Version ST2+," RFC1819, Aug. 1995, ftp://ds.internic.net/rfc/rfc1190.txt.

[15] D. Mitzel *et al.*, "An Architectural Comparison of ST-II and RSVP," *Proc. Infocom '94*, http://www.isi.edu/div7/rsvp/pu b.html.

[16] D. Clark, "The Design Philosophy of the DARPA Internet Protocols," *Proc. ACM SIGCOMM '88*, Aug. 1988.

BIOGRAPHY

PAUL P. WHITE was awarded a B.Eng. degree (First Class Honours) in electronic and electrical engineering at the University of Birmingham, England, in 1989, and an M.Sc. degree (with Distinction) in data telecommunications and networks at the University of Salford, England, in 1995. He is a member of the IEE and has over five years of work experience in various fields of hardware/software technology, including telecommunications. He has been working toward a Ph.D. at University College London since 1995 in the field of integrated services in the Internet and is supported by British Telecom Laboratories, Ipswich, England.