Support for RSVP-based Services over ATM networks

Alex Birman†, Victor Firoiu‡*, Roch Guérin†, Dilip Kandlur†

†IBM T.J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598
{birman,guerin,kandlur}@watson.ibm.com
‡University of Massachusetts, Department of Computer Science, Amherst, MA 01003
vfiroiu@cs.umass.edu

Abstract

In this paper we focus on RSVP-based resource reservations in a heterogeneous environment which includes ATM networks. We describe a method for establishing shortcuts for the data flow through an ATM network which avoids the performance penalty associated with layer 3 processing in the classical IP over ATM approach. For the guaranteed and controlled-load types of services we show how to map the RSVP flow characteristics to ATM call parameters, and thus enable end-to-end quality of service. Finally, we discuss some concerns which have been raised regarding the use of RSVP in establishing shortcuts.

1 Introduction

We consider a heterogeneous environment in which legacy networks coexist with ATM networks. For applications that require performance guarantees the reservation of network resources is carried out using RSVP [5] as the reservation setup protocol. The operation of RSVP over an ATM network is the focus of our paper.

Our starting point is the classical IP over ATM model [10] in which an ATMARP server is used for address resolution within a Logical IP Subnetwork (LIS), while the inter-LIS traffic is routed through IP routers. For an application with QoS requirements the classical IP over ATM architecture does allow for QoS support over the VCs between the routers. Alternatively, the resource reservation can be done at the ATM level, by establishing a direct ATM connection ("shortcut") through an ATM network. This shortcut architecture, compared with the classical one, would benefit from a better usage of network resources, assuming that the ATM network is topologically richer than the overlay IP network. The benefits are further increased if the layer 2 forwarding and QoS enforcement are done more efficiently at the ATM layer. We consider that the former aspect is significant whereas the latter depends on the capabilities of routers and switches in the network. We describe below a method to establish such shortcuts over ATM networks, first for unicast and then multicast application flows. In order to enable end-to-end performance guarantees we study the interplay between establishing RSVP flows and setting up ATM calls. For the Guaranteed and Controlled-Load Service specifications [16, 17], of the Integrated Services framework, we show how to map RSVP flow characteristics to ATM call parameters. Additional details can be found in [2].

The area of ATM technology and its role in the Integrated Services framework of the IETF has been the focus of numerous recent contributions. References [1, 3, 4, 7, 9, 8, 13] in particular are related to the topic of this paper. A different method for establishing shortcuts through an ATM network for multicast sessions, one which is based on changes to the routing protocol, is described in [14].

* Part of this work was done while visiting the IBM T.J. Watson Research Center. Part of this work was supported by the National Science Foundation under Grant NCR-95-08274. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

2 Reservation setup for unicast flows

This section focuses on the RSVP-based reservation setup for *unicast* flows in a heterogeneous environment which includes ATM networks. It is assumed that the data flow traverses an ATM network, and that the source and the destination of the flow could be located on or off this network.

The schemes under consideration aim at setting up QoS VCs through the ATM network. The parameters necessary for setting up these VCs are obtained through the RSVP mechanism involving the flow of $Path$ messages downstream and the flow of $Resv$ messages upstream. We describe two models of RSVP support over ATM networks:

- The "classical" RSVP support preserves the IP routing but adds QoS connections (VCs) between the routers, and between the routers and the hosts (source and destination).
- The "sender-based ATM shortcut" extends the classical RSVP support by enabling ATM shortcuts using the ingress router as the controlling entity for establishing shortcuts.

two other models are described in [2].

2.1 "Classical" RSVP support

Figure 1 shows an ATM network consisting of four LISs. A is the ingress router to the ATM network, B is the egress router. RSVP messages follow the IP route $AEFGB$. Thus, a $Path$ message will travel downstream from A to B, while the corresponding $Resv$ message will travel upstream from B to A. When the $Resv$ message arrives at G the router sets up a VC from G to B (see Section 4 for details). Similarly, VCs will be set up from F to G, from E to F, and from A to E.

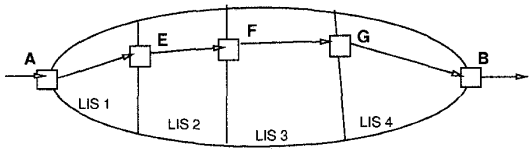

Figure 1: Reservation setup using "classical" RSVP support

In particular, if the ATM network consists of a single LIS then the route from A to B has only one hop, although there could be multiple hops at the ATM level. This would also be the case if all hosts were served by a single Route Server in the Multiprotocol over ATM (MPOA) model [6].

For the multi-hop case, while RSVP messages travel over best-effort VCs, data packets flow over QoS VCs and enjoy QoS

support in the routers. Traversing the routers, however, entails IP-level processing and thus is less desirable than a shortcut VC from A to B. In the rest of this section we discuss several schemes to avoid this overhead by using ATM shortcuts.

2.2 Sender-based ATM shortcut

In this scheme we modify the RSVP operation in order to identify the appropriate egress router for the purpose of establishing a shortcut route through the ATM network (Figure 2). When the

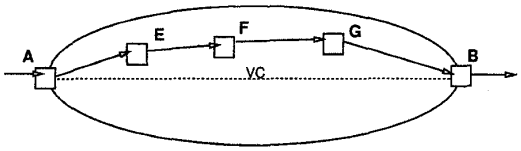

Figure 2: Reservation setup using ATM shortcuts

first $Path$ message for a session arrives at A, the node determines that the message will be forwarded over an ATM link and thus node A is the ingress node into the ATM network. The $Path$ message is routed along the overlay IP route, and is modified to carry both the ATM address and the IP address of A (the IP address of A is the 'previous hop' or PHOP). At each node along the route an ATM connectivity check is performed to determine whether the current node is the egress point from the ATM network. This decision would be based on the ATM connectivity between the current router, the upstream router, and the downstream router as determined by the logical ATM network in which they reside (the concept of the logical ATM network is similar to the one described in the NHRP document [11].) If the current router is not an egress router, it forwards the $Path$ message to the downstream router *without updating the PHOP address field.* This router does not create any $Path$ state for the session. If the current router is an egress router (e.g. B) it processes the $Path$ message in the default manner, creates $Path$ state for the session and stores, among other things, the IP address and the ATM address of A.

When a *new*[1] *Resv* message arrives at B, B inserts its own ATM address as an object into this message, and forwards the message along the default routed path to A. Intermediate routers recognize the *Resv* message but do not create any session or reservation and simply forward the message upstream. When this *Resv* message arrives at A it carries in addition to the regular RSVP information, both the ATM address of the egress router B and QoS information necessary to determine the type of ATM VC that needs to be setup (see Section 4.3 for details). A will then initiate the VC setup.

After the shortcut VC from A to B is set up, it is advantageous to allow the egress router B to suppress the transmission of *Resv* refreshes towards router A, unless they carry a modified service specification. To achieve this, B needs to be able to associate the newly created VC with the RSVP flow. In order to accomplish this, the flow identifier consisting of the tuple `(source address, destination address, transport layer)` is carried in the SETUP message in the Broadband High Layer Information (B-HLI) element[2]. The source and destination addresses themselves further consist of pairs of the form `(IP address, port number)`. Note also that the receipt of the SETUP message provides an implicit acknowledgment that the *Resv* message was received at router A. This means that router A has received all the information necessary to forward *Resv* messages upstream, i.e., the RSVP filter and service specifications that are not directly available from the ATM connection characteristics.

Figure 2 shows a shortcut VC from A to B which bypasses nodes E, F and G. The shortcut VC is used for the RSVP data traffic, but $Path$ messages continue to flow along the default routed path. It is noted that this scheme for creating shortcut routes is independent of the underlying routing mechanism and is oblivious to any IP routing domain boundaries. Moreover, RSVP state is required only at the edge routers A and B.

3 Reservation setup for multicast flows

This section focuses on the RSVP-based reservation setup for *multicast* flows in a heterogeneous environment which includes ATM networks. We consider the general case in which the source of the data flow may reside outside the ATM network, and where the data flow traverses an ATM network in order to reach the receivers of data. These could be located on or off the ATM network.

The IP multicast model is a receiver initiated model and permits many-to-many communication within a multicast group. Receivers wishing to subscribe to a multicast group, which is an IP address in Class D, use the IGMP protocol to inform their local router. Routers use multicast routing protocols such as DVMRP, MOSPF, or PIM to disseminate membership information. A sender wishing to send data to a multicast group simply sends IP packets to the IP address of the multicast group.

In this section we consider three models of RSVP support over ATM networks, and we focus on the case of multicast flows with fixed filter reservations. The first approach, or "classical" RSVP support, preserves the IP routing for the data flow but adds QoS support through ATM VCs between the multicast routers, and between the routers and the participating hosts. The "root-initiated ATM shortcut" model, and the "leaf-initiated ATM shortcut" model, extend the "classical" RSVP support by enabling ATM shortcuts. The main motivation for presenting both approaches is that the former is better suited to the present UNI 3.1 environment, while the latter is the preferred model when the LIJ capability of UNI 4.0 becomes available. Such a distinction is not warranted in the case of unicast flows as both UNI 3.1 and UNI 4.0 yield essentially identical solutions. We then consider the interplay between RSVP 'soft' state and ATM 'hard' state in a discussion on handling failures and route changes. We refer the reader to [2] for a treatment of the general case of multiple-sender multicasts in a discussion on handling RSVP filters.

3.1 "Classical" RSVP support

We extend the "classical" RSVP support of Section 2.1 to single-sender multicast flows. The $Path$ messages traveling downstream are routed by the multicast-capable routers towards the members of the multicast group. The example in Figure 3

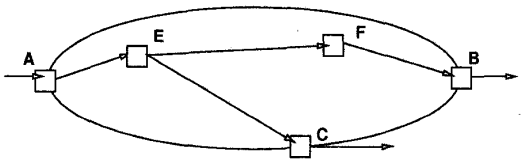

Figure 3: Reservation setup using "classical" RSVP support

shows the route followed by these messages through an ATM network. A is the ingress router to the ATM network, while B and C are the egress routers. Consider now the $Resv$ messages from the receivers of the multicast which follow the reverse path upstream. When the first $Resv$ message from B arrives at F, the router at F sets up a point-to-multipoint VC from F to B. $Resv$ messages from F and C travel independently towards E. The arrival at E of these messages will eventually result in a point-to-multipoint

[1] By new we mean both reservation requests for new flows and requests to modify the reservation of existing flows.

[2] The length of this field would have to be extended from its current size of 8 bytes. The source and destination IP addresses cannot be inferred from the ATM addresses in the router–router case.

VC being set up, having root E and leaves F and C. Another VC will be set up later from A to E.

The previous description concerns initial $Path$ and $Resv$ messages which trigger the reservation setup. $Path$ refresh messages are also forwarded along the IP route, in order to track route changes. Milliken [12] suggests that $Resv$ refreshes are not needed, since the RSVP 'soft' state has been replaced in the ATM environment by a 'hard' state. Following [12], non-refresh $Resv$ messages will be sent only if the QoS parameters of the flow change.

3.2 Root-initiated ATM shortcuts

We extend the unicast scheme of Section 2.2 to single-sender multicast flows, as illustrated in Figure 4. In this method, the

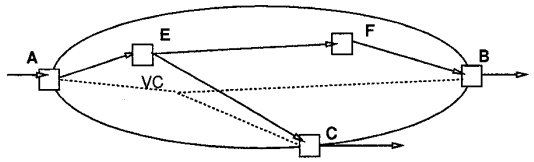

Figure 4: Reservation setup with maximum shortcut

establishment of multicast VC is initiated at the ingress router A, the root of the VC, this being suited to a UNI 3.1 environment. The determination of the ATM shortcut follows the same steps as in Section 2.2. When a $Path$ message for a session arrives at node A, the node determines[3] that the message will be forwarded over an ATM link and thus node A is the ingress node into the ATM network. The ATM address of A is inserted as an object into the $Path$ message, which is routed over the IP route. At each node along the route an ATM connectivity check is performed to determine whether the current node is an egress point from the logical ATM network. If the current node, such as F in Figure 4, is not an egress point then the $Path$ message is forwarded to the downstream nodes without updating the PHOP (previous hop) address field. As in the unicast case, F does not create and maintain a $Path$ state for this flow. Note that this means that we are in fact using automatic tunnelling back to the ingress router A. However, this also implies that the $Resv$ messages will be handled as default IP traffic and not as control messages. This lack of preferential treatment for $Resv$ messages is the price paid for avoiding states at intermediate routers.

When the first $Resv$ message arrives at an egress point, say B, B forwards the message along the reverse path to A. The ATM address of B is carried as an object in the $Resv$ message. Intermediate routers, F and E in this case, simply forward the message upstream towards A. Specifically, they do not merge $Resv$ messages and do not perform any reservation. When the first $Resv$ message arrives at A, say from B, A has all the information necessary to create a shortcut point-to-multipoint VC with root A and leaf B. In order for B to associate the newly created VC with the RSVP flow, the flow identifier consisting of the pair `(source IP address, destination IP address)` is carried in the SETUP message in the Broadband High Layer Information (B-HLI) element. Later, when the $Resv$ message from C arrives at A, A adds C to the point-to-multipoint VC with an ADD PARTY signaling message. The ADD PARTY message will also carry the flow identifier in the B-HLI element.

In order to track route changes and changes in group membership, $Path$ refresh messages keep flowing normally over the IP route. However, $Resv$ refreshes from each router are suppressed as soon as the egress router receives the ATM setup message (ADD PARTY or SETUP for the first leaf). This is because the setup message indicates that the initial $Resv$ message has been received by the ingress router, and that the reservation through the ATM network has been successfully performed. This suppression prevents the steady state implosion of refresh $Resv$ messages at the ingress router. However, the ingress router is still required to perform as many ATM connection SETUPs as there are leaves in the ATM network for the multicast address. This is because, the scheme always results in the use of a "maximum" ATM shortcut that directly connects the ingress and egress routers. A more promising and systematic approach to eliminate the possibility of signaling overload at the ingress router, it the use of Leaf-Initiated Join (LIJ) in UNI 4.0, which is discussed next.

3.3 Leaf-initiated ATM shortcuts

Consider the ATM network in Figure 4 and assume that the flow of $Path$ messages is as described in the previous section. That is, $Path$ messages continue to use the default IP routed path. As before, the $Path$ messages are not processed at intermediate routers and the PHOP is not modified. $Path$ messages are also extended at the ingress router A to carry the ATM address of A. In addition, A also chooses an ATM "global connection identifier" (GCID), and inserts it into the $Path$ message. This global connection identifier consists of a call identifier uniquely chosen by the root, which is paired with the root's ATM address for LIJ setup. For a given RSVP session, there may be multiple flows transiting through A and, for each flow, A would choose a distinct global connection identifier. This connection identifier will be used by egress routers when generating an ATM LIJ request to join the point-to-multipoint connection associated with the IP multicast address.

When the first $Resv$ message reaches an egress router, say B, the router has all the information needed to generate a Leaf Initiated Join (LIJ) request to the connection identified by the GCID received. The ATM point-to-multipoint connection is then created at this time, with the ingress router A as its root and B as the first leaf. As other egress routers, such as C in Figure 4, also receive their first $Resv$ message, they signal their intention to join the connection in exactly the same manner, i.e. through a LIJ request to the specified GCID. They are then added as new leaves to the existing point-to-multipoint connection, and the ingress router A is not notified of this new join, which eliminates the potential processing overload at router A.

However, note that as a result of not notifying the ingress router of new leaves joining, the information carried in the $Resv$ messages arriving at the associated egress routers is not forwarded to the ingress router during the ATM setup process. This information is, however, necessary for the ingress router to further propagate $Resv$ messages upstream, i.e. it needs information elements such as the RSVP service and filter specifications, which as mentioned before cannot always be directly inferred from the ATM traffic and QoS parameters. In order to achieve this, $Resv$ messages, including refreshes, will continue to be propagated and merged on the IP path, but no reservation will be triggered at intermediate routers. The merging on the IP path ensures that the ingress router is not overwhelmed by the volume of refresh $Resv$ messages it receives, while providing it with all the information it needs to forward $Resv$ messages to its upstream neighbor. Note that $Resv$ refreshes are not suppressed in order to ensure reliable delivery of $Resv$ messages to the ingress router.

4 Issues Related to Flow/Call Characteristics

The previous sections have dealt with many of the issues related to the mapping between RSVP and ATM control flows. In this section, we focus on similar problems but at the level of the data flows. Specifically, we consider issues related to the mapping of traffic parameters and QoS guarantees as well as function placement. Some of these mappings consist of relating ATM cell-based measures to the corresponding packet/byte level quantities used in RSVP. Others are caused by differences in service

[3] This step only needs to be performed upon receipt of the first *Path* message.

specifications and capabilities, or simply needed to identify where and how each step in the establishment of a connection is to be performed.

4.1 Traffic parameters mapping

Traffic is characterized in both IP and ATM using leaky bucket models, with peak rate, mean rate and maximum burst size ((p, m, b) in IP and (PCR,SCR, MBS) in ATM). If we assume a perfect fluid model for both the IP flow and the ATM call and ignore the potential impact of the granularity of the ATM cell size and of the segmentation overhead, the following relations can be established [16]:

$$\text{SCR} = \frac{r}{48}\ ; \quad \text{MBS} = \frac{b}{48}\ ; \quad \text{PCR} = \frac{P}{48}\ , \qquad (1)$$

where 48 is the number of user data bytes per ATM cell, and P corresponds to the minimum of the speed of the incoming link and the peak rate p of the flow. Note that for a Controlled Load flow, P is always set to the speed of the incoming link since the Controlled Load TSpec does not include a peak rate term. It should be pointed out that the above expressions need to be adjusted to reflect the impact of the ATM segmentation in fixed size cells. We analyze these adjustments in [2].

4.2 Mapping of Controlled Load Service Specifications

The Controlled Service can be mapped to the Available Bit Rate service of ATM [15] in the case of unicast flows. In this case, the token bucket rate r of the Controlled Load TSpec is mapped to a corresponding value of the ABR Minimum Cell Rate (MCR). This value guarantees a floor rate to the flow, which is in keeping with the spirit of the Controlled Load specifications. The ABR service also allows transmission at a higher rate when resources are available. This is also in keeping with the spirit of the Controlled Load specifications, that allows a Controlled Load flow to exceed its TSpec but without any real guarantee for that traffic. In that respect, the ABR service will actually provide a better level of service through the ATM network since it determines, through feedback messages, the maximum rate at which the flow can transmit without risking excessive losses. However, note that this ability of the ABR service to identify the "bottleneck" link for the flow, comes at a price, i.e., the overhead of generating and processing RM cells.

While ABR provides a suitable service mapping for unicast Controlled Load flows, it cannot be used for multicast flows. This is because the ABR specifications have currently not been defined to cover the case of point-to-multipoint connections. Instead, Controlled Load multicast flows will have to be mapped onto Variable Bit Rate Non Real Time (VBR-NRT) ATM connections. Unfortunately, this mapping does not readily allow for Controlled Load flows to exceed their specified rate, unless the rate parameter of the corresponding VBR connection has been set to a higher value than that corresponding to the token bucket rate r of the Controlled Load flow. This could be wasteful of resources in the ATM network. Another, possibly preferable, alternative is to rely on the marking feature of ATM networks, where excess traffic from the Controlled Load flow would be sent as CLP=1 cells through the ATM network. Note that such a mapping could also be used for unicast flows if desired.

4.3 Mapping of Guaranteed Service Specifications

In the IP Guaranteed Service model [16], a maximum delay $\tilde{d}$ is guaranteed to a flow by reserving a minimum buffer clearing rate R such that

$$\tilde{d} = \frac{M + C_{tot}}{R} + D_{tot} + \begin{cases} \frac{b-M}{R}\frac{p-R}{p-r} & r \le R < p \\ 0 & p \le R \end{cases} \qquad (2)$$

where $\mathcal{C}_{tot} = \sum_S^D \mathcal{C}_i$ and $\mathcal{D}_{tot} = \sum_S^D \mathcal{D}_i$ are values accumulated by $Path$ on its way from source S to receiver D. R is computed at the flow's receiver such that $\tilde{d} \le d$, the end-to-end delay requirement. R and $\mathcal{S} = d - \tilde{d}$ are included in RSpec and sent toward S in the $Resv$ message.

A key aspect of the above approach, that complicates the interactions with ATM, is the decoupling between the advertising (accumulation of $\mathcal{C}_{tot}$ and $\mathcal{D}_{tot}$ as the *Path* message progresses) and the reservation phases (request for allocation of the clearing rate R). The main issue at the boundary of an ATM network is to determine which values to select for the terms $\mathcal{C}_{\text{ATM}}$ and $\mathcal{D}_{\text{ATM}}$ (that characterize the future ATM connection), when updating the $\mathcal{C}_{tot}$ and $\mathcal{D}_{tot}$ fields in the *Path* message. This is difficult for two reasons. First, the correct values are function of the path through the ATM network, and this is not known at the time the *Path* message reaches the ingress (or egress) router of the ATM network (it will only be nailed down upon receipt of a *Resv* message at the egress router of the ATM network). Second, the form of the delay guarantees specified in [16], i.e., based on the specification of a clearing rate, will typically not be supported by ATM switches, and furthermore cannot be readily expressed through the ATM signaling. This means that the ATM network has to be accounted for as a fixed delay component on the path. Hence the need to determine a value to advertise for $\mathcal{D}_{\text{ATM}}$, and further to comply with this advertised value when an ATM connection actually needs to be setup upon receipt of a *Resv* message. In the rest of this section, we review alternatives to address this problem. We study the problem of resource reallocation following a change in flow or service specification in [2].

4.3.1 Unicast flows

The case of a unicast flow is illustrated in Figure 2. When $Path$ arrives at A, the fields $\mathcal{C}_{tot}$ and $\mathcal{D}_{tot}$ contain the values $\sum_S^A \mathcal{C}_i$ and $\sum_S^A \mathcal{D}_i$. Then *Path* stops accumulating $\mathcal{C}_i$ and $\mathcal{D}_i$ for the duration of its journey through the ATM network, i.e., until it reaches router B. The issue is then to determine an estimate of the end-to-end delay guarantee $\mathcal{D}_{A,B}$, that given the traffic parameters provided in the TSpec of the *Path* message, can be provided between A and B by the ATM network. We assume here, that the mapping of the TSpec onto ATM traffic parameters is done following one of the methods of Section 4.1.

The first issue is to identify the router which is responsible for determining the value $\mathcal{D}_{A,B}$, and updating the *Path* message accordingly. There are two choices, the ingress or egress routers, i.e., router A or B. Both are equally capable of obtaining an estimate for $\mathcal{D}_{A,B}$, provided they know each other's ATM address. Access to this knowledge is dependent on the approach used to forward RSVP control information across the ATM network. From the discussion in Section 2, we know that using any of the recommended solutions to forward *Path* messages across the ATM network, the ATM address of the ingress router A is delivered to the egress router B together with the first *Path* message. This means that the $\mathcal{C}_{tot}$ and $\mathcal{D}_{tot}$ fields contained in this first *Path* message cannot have been updated by the ingress router A to advertise an estimate of the delay guarantee $\mathcal{D}_{A,B}$ across the ATM network. As a result, it is simpler to leave the responsibility of determining an appropriate value for $\mathcal{D}_{A,B}$ to the egress router B. In addition, as we shall see in the next section, this is also consistent with the approach that has to be used in the multicast case. However, note that this now requires that the selected value for $\mathcal{D}_{A,B}$ be communicated back to router A, so that it can specify the correct value in those cases where it is responsible for initiating the ATM call setup associated with the RSVP flow. This is done by including this information, together with the ATM address of router B, in the first *Resv* message that router B forwards to router A. Note that this problem does not arise if the receiver initiates the connection SETUP.

Once we have identified the router responsible to carry out the determination of $\mathcal{D}_{A,B}$, it remains to specify how this is done. There are two generic approaches to obtain an estimate of $\mathcal{D}_{A,B}$.

Local Determination of Delay Estimate This solution is the simplest in that it involves minimal interaction with the ATM network. Router B generates an estimate for the delay $\mathcal{D}_{\text{ATM}}$ from router A to itself across the ATM network. This estimate can be a pre-configured value or could be inferred from information made available by the ATM network.

Query ATM Network for Delay Estimate This solution attempts to improve the accuracy of the delay estimate by actually querying the ATM network. This query takes the form of an actual connection establishment request to the ATM network, to setup a connection between A and B with specific delay guarantees. The details of obtaining information through ATM connection negotiation is dependent on the ATM Forum UNI and PNNI specifications under development. We present a possible approach in [2].

One feature common to the solutions above is that the advertised value is likely to be rather inaccurate, which can greatly increase the rejection rate of connections having to traverse ATM networks. It is, however, possible to greatly reduce the impact of this inaccuracy, by allowing some flexibility in the delay guarantee that is eventually required from the ATM network, i.e., provide a safety margin around the advertised value. Such a capability is included in [16] as the delay slack $\mathcal{S}$. The slack $\mathcal{S}$ corresponds to what remains of the end-to-end delay budget after the receiver has chosen a value for R. A receiver could purposely[4] select an R value so as to create some slack. The slack can then be used to provide some flexibility in the required delay guarantees through ATM networks. Specifically, each router on the path from S to D can take some of the slack if necessary, provided it properly updates the slack field to reflect the adjusted amount. This means that if router i consumes an amount $\mathcal{S}_i$ of the slack, it updates the slack field as follows: $\mathcal{S} \leftarrow \mathcal{S} - \mathcal{S}_i$, before forwarding the *Resv* message to its upstream neighbor. In the context of a connection through an ATM network, the slack (if present) can be used to compensate for differences between the value currently feasible, and the quantity $\mathcal{D}_{A,B}$ that was initially advertised. This can improve the chances of success of the connection.

4.3.2 Multicast flows

Multicast flows share the problems of unicast flows when mapping IntServ delay guarantees to corresponding ATM quantities. In the following we focus on aspects for which significant differences exist between the multicast and unicast cases.

A first major difference is in the location where an estimate for $\mathcal{D}_{\text{ATM}}$ can be obtained. In the unicast case, this could be performed at either the ingress or the egress routers. In the multicast case, the determination of an estimate for $\mathcal{D}_{\text{ATM}}$ must be performed at the egress routers for multicast flows because it is likely that different ATM addresses would yield different values for $\mathcal{D}_{\text{ATM}}$, and those could not be differentiated through a single *Path* message at the ingress router. Note that each egress router determines an estimate for $\mathcal{D}_{\text{ATM}}$ between itself and the ingress router whose ATM address was carried in the *Path* message. This corresponds to a direct ATM connection between the ingress and egress routers, which is unlikely to be the case as connectivity to (all) the egress routers will typically be provided by a single point-to-multipoint connection. This is yet another source of inaccuracy in the determination of $\mathcal{D}_{\text{ATM}}$. It should be, however, of limited significance.

A second difference between unicast and multicast flows is in the way the information is provided to and used by the router responsible for setting up the ATM connection. In the multicast case, we need to distinguish two cases depending on the type of signaling available to establish point-to-multipoint connections.

[4] For example, if it knew it had to cross some ATM networks.

Root-initiated point-to-multipoint ATM conn. This is the only approach available in UNI 3.1. The point-to-multipoint connection is root initiated, i.e., it relies on ADD-PARTY messages that all originate from the root. It is then imperative that the root be provided with both the ATM address of the egress router and the value of $\mathcal{D}_{\text{ATM}}$ to be used in each ADD-PARTY. These must, therefore, be included in the *Resv* generated from all the egress points. As discussed in Section 3, the *Resv* messages should not be merged as information on each individual "leaf" is needed at the root to setup the point-to-multipoint ATM connection.

Leaf initiated join (LIJ) to a point-to-multipoint ATM connecti
In this case, the egress routers directly join the point-to-multipoint connection while specifying the desired delay guarantee $\mathcal{D}_{\text{ATM}}$ they determined and advertised in the *Path* messages. Note that while UNI 4.0 defines the LIJ capability, it does not yet allow specification of different guarantees to different leaves. The unavailability of such a feature is clearly a major problem in supporting multicast RSVP flows.

5 Discussion

In this paper we focused on the establishment of QoS connections in a heterogeneous environment which includes ATM networks. For sessions whose path extends across ATM networks we investigated the interplay between RSVP flows at the network layer and ATM calls at the subnetwork layer.

We have used an approach, classical RSVP over ATM with shortcuts, which is intended to leverage the strengths of the ATM technology in support of applications with QoS requirements. Establishing shortcuts through an ATM network avoids the performance penalty associated with layer 3 processing in a classical IP over ATM approach.

We provided solutions applicable to both unicast and multicast flows. While attempting to provide solutions which apply equally to signaling in the UNI 3.1 environment, as well as the UNI 4.0 environment, we found that different solutions are required in order to account for significant differences between the two signaling protocols. Additional details may be found in [2].

A number of issues have been raised regarding the approach used here [14]. It has been pointed out that establishing shortcuts via RSVP, and using these shortcuts for multicast traffic, may result in receivers receiving duplicate packets. It has also been pointed out that changes to RSVP are required for implementing the approach described here. Finally, it was suggested that performing shortcuts via RSVP means that RSVP would perform a routing function. This is contrary to a stated RSVP design principle, which requires that the routing function be separate from RSVP, and interoperable with it.

The problem of duplicate packets at some receivers arises sometimes because we want to ensure that we reach all receivers, including those that do not make any reservation. In order to achieve this, data packets should be sent on both the default VC, and the shortcut VC. Depending on how packets are delivered to those receivers which did not make reservations, some receivers with active reservations may receive packets on both the shortcut path, and the default path.

This duplication of packets does not seem to be a serious problem. In normal cases, a simple filtering solution can essentially eliminate the problem altogether. For example, a receiver connected to a shortcut VC stops listening for packets of that flow on the default path. This type of filtering, whose overhead is typically negligible for the end-station, is common in multicast routing.

In contrast to this duplication of packets at the receivers, the requirement that packets be duplicated by the sender, and then sent on a single ATM interface, is potentially more troublesome. Both

problems, however, could be avoided altogether if, and when, 'variegated' ATM VCs become available, i.e. point-to-multipoint VCs which provide different QoS to different receivers, as needed. Until this type of connections is available, one could avoid the packet duplication at the receivers by extending the shortcut VC to include all receivers, including those without a reservation. This solution, which gives receivers without reservation a "free ride", is our preferred solution, and it is similar to the case of a broadcast medium in which some receivers obtain enhanced service based on other receivers' reservations. Our conclusion, therefore, is that packet duplication is not a major problem, and can be handled in a number of ways. If variegated VCs become a reality, it will be eliminated altogether without any sacrifice in efficiency.

The next issue concerns changes to RSVP, and the implication here is that such changes are bad. A closer look at the goals RSVP was designed for, and where RSVP is today on the way to achieve those goals, would be of help to evaluate if changing RSVP is appropriate. This, it turns out, is related to the last issue, whether RSVP is performing a routing function when the shortcut is carried out.

Current routing protocols do not take into account QoS. Likewise, there is no RSVP mechanism to instruct routing on QoS attributes. While RSVP designers stated a worthy goal, that is the goal of separating the routing function from the resource reservation function, it is still an open question how QoS is to be attained. Meanwhile, it is likely that RSVP, and routings protocol as well, will change if the promise of QoS is to become reality.

There are additional considerations, concerning the actual changes to RSVP which are needed in order to implement our approach. The signaling mechanism implemented by *Path* and *Resv* messages coincide with a second signaling requirement, for carrying addressing information needed to establish a layer 2 shortcut. Thus, RSVP messages serve as a vehicle for the shortcut information, but this artifact does not turn RSVP into a routing protocol. While RSVP messages always follow the IP route, once the shortcut is put in place the data flows over it, and not on the IP route. Having the data traffic follow a different path from the path of the signaling messages is needed in order to exploit the ATM attributes, and we see nothing wrong with it. While QoS information carried in the RSVP messages is used by the shortcut managing entity, RSVP has no say in establishing the shortcut.

One could decouple RSVP from the shortcut mechanism by using NHRP instead. If so, NHRP state must be maintained at the source and destination endpoints for each QoS session, which is essentially a duplication of part of RSVP state. Moreover, a temporary connection, along the same path as the shortcut is required to carry RSVP messages prior to creation of the QoS connection. This alternate solution suffers from duplication of function, and thus is not efficient.

The approach described here has, without doubt, shortcomings. Some of these can be traced to the differences between RSVP and ATM signaling and their, sometimes conflicting, design principles. This approach is offered as a possible first step in supporting QoS flows in a heterogeneous environment with ATM networks. If adopted and carried through to implementation, the experience thus gathered may be beneficial in the design of a better next scheme.

In this paper we have touched on a number of topics for which much work remains to be done. Here are two specific items. First, a method is needed to better account for an ATM network's contribution during the advertising phase carried out through RSVP *Path* messages. This can mean better estimates for the delay guarantees that an ATM network can provide, or extensions to ATM signaling and service specifications to better emulate the Integrated Services model [16]. Second, while the LIJ of the UNI 4.0 is intended to improve scalability by removing the bottleneck associated with the ingress router, it does so by shifting the processing burden from the ingress router to the ATM network. It is, therefore, important to ensure that the the server infrastructure which handles ATM signaling is designed in a scalable way, and is able to support large multicast groups.

References

[1] F. Baker, A. Lin, and Y. Rekhter. Support for RSVP and IP Integrated Services over ATM. ATM Forum 96-0028, January 1996.

[2] A. Birman, V. Firoiu, R. Guerin, and D. Kandlur. Provisioning of RSVP-based Services over a Large ATM Network. Technical Report RC20250, IBM T.J.Watson Research Center, October 1995.

[3] M. Borden, E. Crawley, J. Krawczyk, F. Baker, and S. Berson. Issues for RSVP and Integrated Services over ATM. Internet draft-crawley-rsvp-over-atm-00.txt, work in progress, February 1996.

[4] M. Borden, K. Faulkner, and E. Stern. Support for RSVP in ATM Networks. ATM Forum 96-0039, February 1996.

[5] R. Braden, L. Zhang, D. Estrin, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification. Internet draft-ietf-rsvp-spec-13.ps, work in progress, July 1996.

[6] C. Brown. Baseline Text for MPOA. ATM Forum 95-0824R1, July 1995.

[7] J. Crowcroft. RSVP and Q.2931. unpublished manuscript, February 1996.

[8] A. Demirtjis, S. Berson, B. Edwards, M. Maher, B. Braden, and A. Mankin. RSVP and ATM Signaling. ATM Forum 96-0258, January 1996.

[9] R. Guerin and D. Kandlur. Issues in Extending Unicast and Multicast RSVP Flows across ATM Networks. ATM Forum 96-0094, February 1996.

[10] M. Laubach. Classical IP and ARP over ATM. Internet RFC1577, January 1994.

[11] J. Luciani, D. Katz, D. Piscitello, and B. Cole. NBMA Next Hop Resolution Protocol (NHRP). Internet draft-ietf-rolc-nhrp-09.txt, work in progress, July 1996.

[12] W. Milliken. Integrated Services IP Multicasting over ATM. Internet draft-milliken-ipatm-services-00.txt, work in progress, July 1995.

[13] R. Onvural and V. Srinivasan. A Framework for Supporting RSVP Flows Over ATM Networks. Internet draft-onvural-srinivasan-rsvp-atm-00.txt, work in progress, February 1996.

[14] Y. Rekhter and D. Farinacci. Support for Sparse Mode PIM over ATM. Internet draft-rekhter-pim-atm-00.txt, work in progress, February 1996.

[15] S. Sathaye. Traffic Management Specification 4.0. ATM Forum 95-0013, February 1996.

[16] S. Shenker, C. Partridge, and R. Guerin. Specification of Guaranteed Quality of Service. Internet draft-ietf-intserv-guaranteed-svc-06.txt, work in progress, August 1996.

[17] J. Wroclawski. Specification of the Controlled-Load Network Element Service. Internet draft-ietf-intserv-ctrl-load-svc-03.txt, work in progress, August 1996.