RC 20250 (10/27/95)
Computer Science

IBM Research Report

Provisioning of RSVP-based Services over a Large ATM Network

A. Birman†, V. Firoiu‡, R. Guérin†, and D. Kandlur†

†IBM Research Division
T.J. Watson Research Center
Yorktown Heights, New York

‡Computer Science Department
University of Massachusetts
Amherst, MA 01003 USA

Provisioning of RSVP-based Services over a Large ATM Network

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, Computer Science Department, Amherst, MA 01003
vfiroiu@cs.umass.edu

November 1, 1995

Abstract

In this work we address the problem of flow establishment with QoS constraints and reservation of resources in an IP network that spans over ATM networks. Specifically, we study the interaction and integration of the RSVP protocol using Integrated Services flow specifications with the ATM signaling. Among the issues raised by such an integration are: the interaction between the flow/call establishment protocols, and the translation between flow/call characteristics. For these issues we sketch several possible solutions rather than focus on a particular design. Finally we discuss the extensions to RSVP and to the ATM signaling required for the implementation of these solutions.

*Part of this work was done while visiting the IBM T.J. Watson Research Center

Contents

1 Introduction

Quality of Service (QoS) guarantees for both unicast and multicast applications are expected to become increasingly important and the corresponding traffic to amount to an increasingly larger fraction of network resources. This trend is reflected in the introduction of a number of proposed protocols for flow establishment and allocation of network resources. Some of the major ones are ST-II [Top90, DB95] and RSVP [ZDE$^+$93, BZE$^+$95] for IP networks, and UNI signaling [For94, Sam95, Sat95] for ATM networks. While these varied proposals reflect the high level of activity in this area, they also introduce additional issues to an already complex problem. Specifically, it now becomes necessary to enable these different protocols to inter-operate, i.e. to provide a homogeneous service in a heterogeneous environment.

1.1 Problem statement

The goals of this document are to introduce some of the issues faced when attempting to provide end-to-end QoS guarantees to IP flows whose path crosses ATM networks, and discuss possible solutions. We consider user applications that originate in the IP domain and that use the RSVP protocol to request QoS guarantees. Our target is the definition of an interface between the RSVP protocol and the ATM signalling, that specifies how control and data flows are to be mapped from the RSVP to the ATM environment, and vice-versa. This mapping should be such that it allows efficient utilization of ATM resources in satisfying the desired end-to-end QoS guarantees, and is transparent to the user applications and to the RSVP service elements not directly connected to the ATM network. Note that an implicit assumption here is that ATM switches are typically not capable of performing any IP processing (in contrast to [OEN95]).

Obviously, the extension of IP flows across ATM networks raises many problems, that need to be addressed irrespective of whether QoS guarantees are requested or not. Resolving such issues is clearly important, and much recent work has been devoted to this topic. As a result, and since our focus is on aspects that are specific to supporting QoS guarantees, we will, whenever possible, simply rely on known solutions to deal with "standard" IP over ATM inter-operability problems. For example, we will assume that mechanisms such as those of [Hei93, Atk94] are used to encapsulate IP packets into ATM AAL5 PDU. Similarly, we will defer the address resolution (IP address to ATM address) problem within a single Logical IP Subnet (LIS) to existing proposals [Lau94, Arm95].

In addressing the problems associated with the extension of IP (RSVP) QoS guarantees across ATM networks, our general goal is to preserve in as much as possible the robustness and scalability of the RSVP protocol, while leveraging ATM's ability to efficiently provide QoS guarantees. For example, our solution should preserve RSVP's ability to recover from failures in either the IP or the ATM portions of a path. It should also be able to handle large networks and large numbers of (RSVP) receivers. Conversely, the solution should be flexible enough to accommodate possible extensions and enhancements to the ATM signalling that are currently under development, e.g., negotiation of individual QoS parameters, Leaf Initiated Join, etc. Furthermore, it should also preserve efficient use of resources and attempt to maximize the use of ATM capabilities and paths whenever possible. In general, this means that we will favor the use of direct ATM connections against paths that involve multiple IP hops. Note that we, however, assume the existence of an IP *overlay* network, that is used to carry IP traffic without QoS guarantees. Similarly, we also defer the problem of routing and distribution of routing information to this overlay IP network. Finally, we make the further restriction that there is a one-to-one correspondence between RSVP flows and ATM connections, and we limit our attention to RSVP flows that require Guaranteed Delay service.

1.2 Related works

As mentioned before, the emergence of ATM as a key networking technology has triggered a large number of studies addressing the interactions between IP and ATM networks. Some comprehensive surveys of such works can be found in [CSV95, All95].

The mapping of IP packets to ATM cells at the data link layer is a relatively well understood issue, and a stable set of solutions has been established. For example, the specification of multi-protocol encapsulation over ATM Adaptation Layer 5 can be found in [Hei93], although special encapsulation provisions for IP multicast routing over ATM might still be needed [GA95]. The issue of packet fragmentation is addressed in [Atk94], where a large IP MTU size (9180 bytes) is proposed for efficient IP packet handling.

On the other hand, when it comes to issues related to the mapping of IP flows/routes to ATM connections at the network layer, a large number of alternatives are still being considered, and this is where the focus of much of the recent work on IP over ATM has been. Many of the problems that have been addressed so far are related to route establishment and address resolution for both unicast and multicast communications. For example, in the simple case of "local" ATM networks (spanning only one IP network address), a now well established solution to the problem of IP routing and address resolution is described in [Lau94, PLM+95] (Classical IP/ATM). It essentially defines how to directly connect local nodes via ATM VCs, and how to connect to remote nodes through routers, that are themselves connected via a chain of VCs. Here, the notion of "local vs remote" is based on the source and destination IP addresses and the subnetwork mask specification. Specifically, within a LIS there is a single ATMARP server which resolves ATM addresses for all nodes in that subnet.

Extensions to the above model were proposed in [RK95], where the establishment of direct ATM VCs was based on QoS requirements, rather than on the IP destination address. Another extension of the Classical model, named Conventional model [OEN95], attempts to eliminate the overhead of hop-by-hop IP processing when going to a remote destination, by enabling the routers to "splice" directly at the ATM level the VCs that are associated with the same IP flow, thus forming a "bypass pipe."

The general issue of a scalable address resolution mechanism for non-broadcast multi-access (NBMA) networks, e.g. ATM, has also been addressed in [KP95, Can95] which describes the Next Hop Resolution Protocol (NHRP). The NHRP protocol describes how queries for the ATM address associated with a given IP resource are to be propagated in an NBMA network, and how and when replies carrying a destination ATM address are to be returned. Similarly, support for IP/ATM multicast and broadcast routing is proposed in [Arm95, SA95], where a set of multicast group membership servers (MARS) provides multicast address resolution within the ATM network. Finally, the problem of IP inter-domain routing over ATM is considered in [RV95].

All of the works mentioned above rely on the model of an "overlay" IP network on top of the ATM network, with some exchange of control information between the two. A different model is used in [CS95, Cal94, PL94, Bro95], which assume an environment where IP routers and ATM switches interact on a peer-to-peer basis (Peer or Integrated model). In this integrated model, interactions between ATM and IP are greatly simplified, but this imposes constraints that may not always be easily satisfied. For example, the Integrated model requires that IP and ATM routing be consistent and, therefore, be based on similar metrics. Similarly, the Peer model requires both IP routers and ATM switches to use a common (interchangeable) address format. This clearly represents a desirable goal, but it is not clear how soon it can be realized. As a result, in this report we do not require that the assumptions of this integrated model be satisfied.

In the area that is the focus of our work, i.e. the establishment of real-time flows with QoS requirements, the main issue is to deal with the heterogeneity of the flow establishment protocols (ST-II and RSVP for IP and ATM signaling for ATM). In early works [GS93, HP93] proposed mechanisms for interactions between ST-II and ATM signalling. [KNE95] outlines a scheme by which the construction of an ATM "bypass-pipe" could be triggered by RSVP messages, while [KN95, Got95] dealt with the issue of how to advertise IP flow identifiers across an ATM network and how to map them onto ATM call identifiers. More recently, [BCDB95] identified a set of problems related to the establishment of real-time IP flows across ATM networks. The most complete work on this issue to date and the one most relevant to our work, is probably [Mil95a, Mil95b], which puts forth a comprehensive proposal on how to support unicast and multicast IP flows, both best effort and real-time, over ATM networks. In this proposal, the best effort traffic is handled through regular IP forwarding, i.e. a router overlay network, whereas the real-time flows are established through a combination of RSVP messages and ATM signalling using ATM VCs across the ATM network.

The rest of this document is structured as follows. In Sections 2 and 3 we describe mechanisms that relate flow establishment in IP and call establishment in ATM, for the unicast and multicast cases, respectively. In Section 4 we address the issue of translating flow/call specifications (traffic parameters and QoS requirements) and the service requirements of the RSVP protocol into corresponding ATM signalling and QoS negotiations. In particular, we propose simple mappings between the IP Integrated-Services service specifications [SP95] and ATM QoS parameters [Sat95]. Section 5 summarizes the proposal puts forth in the paper, and identifies both the necessary modifications to the current RSVP protocol and the extensions and changes to the ATM signalling that are needed to better support RSVP. It also points to a number of open issues.

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 four 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 "RSVP-based ATM shortcut" and the "NHRP-based ATM shortcut" extend the classical RSVP support by enabling ATM shortcuts using the ingress router as the controlling entity for establishing shortcuts. The main difference between these shortcut methods is that the latter attempts to leverage and extend existing NHRP mechanisms to determine how ATM shortcuts should be established. The "Receiver-based ATM shortcut" method exploits the duplex nature of ATM point-to-point VCs to allow the egress router to become the controlling entity for establishing shortcuts. The main benefit of this method is its synergy with the handling of multicast flows when support for Leaf Initiated Join (LIJ) becomes available from the ATM signalling, i.e. in UNI 4.0 (see Section 3.4 for details).

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 has sufficient information to set up a VC from G to B. Similarly, VCs will be set up from F to G, from E to F, and from A to E.

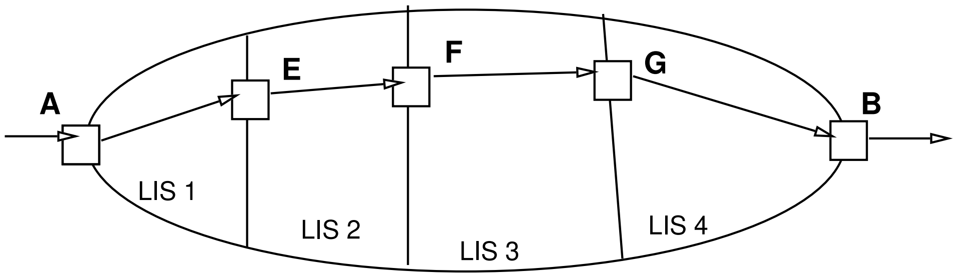

Figure 1: Reservation setup using "classical" RSVP support

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 [Bro95].

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 for RSVP support using ATM shortcuts.

2.2 RSVP-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. When the first *Path* message for a session arrives at A (Figure 2), 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.) 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 4.2 for details).

Since intermediate nodes do not need to process the *Resv* message, an alternative here is to encapsulate the *Resv* message into an IP datagram that is then tunneled from B to A. Tunneling provides the advantage that packet processing is expedited (along the fast-path through the router) since there is no special processing at intermediate nodes. On the other hand, the packet is not treated as a signaling packet and is susceptible to normal loss at intermediate nodes.

After the shortcut VC from A to B is set up, B needs to be able to associate the newly created VC with the RSVP flow. In order to achieve 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

[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.

acknowledgment that the *Resv* message was received at router A. This means that router A also 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. As a result, the egress router B now suppresses the transmission of *Resv* refreshes towards router A, unless they carry a modified service specification.

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 in the edge routers A and B.

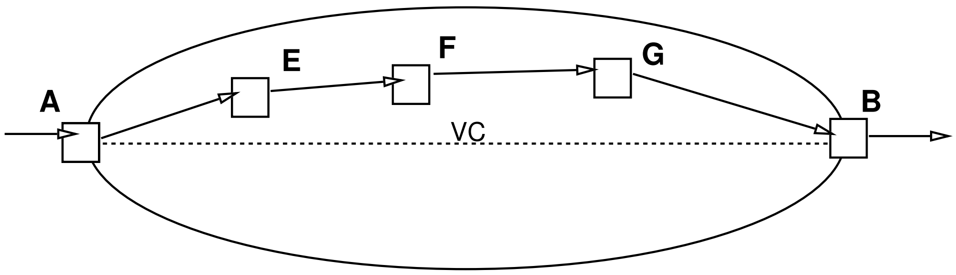

Figure 2: Reservation setup using ATM shortcuts

2.3 NHRP-based ATM shortcuts

An alternative but functionally equivalent method to setup an ATM shortcut route, is to instead rely on an extension of NHRP. In this method the ingress router A creates an NHRP Query and stores the contents of the RSVP *Path* message in the QoS object of the query. The NHRP query travels to the appropriate egress router B which then recreates the *Path* message. B also creates an NHRP response and returns it to A. When a new *Resv* message arrives at B it is as with the previous method forwarded to A, which then sets up the ATM VC to B.

The main advantage of this approach is that it avoids many modifications to the handling of *Path* and *Resv* messages, by instead relying on the availability of NHRP mechanisms. Furthermore, as discussed in Section 2.5 it also makes possible the suppression of *Resv* and *Path* refresh messages once the ATM shortcut has been established. However, because the approach is not readily extendable to the multicast case, the solution of Section 2.2 is probably the preferred one of the two.

2.4 Receiver-based ATM shortcuts

This last shortcut method is identical to that of Section 2.2 in its handling of *Path* messages, but differs from it in that it shifts the responsibility of establishing the ATM shortcut VC from the ingress router A to the egress router B (see Figure 2). This is possible because ATM unicast calls are always duplex, and resources can be reserved in both directions. Specifically, when a *Resv* message arrives at the egress router B, B can generate a SETUP message towards A and specify the resources required in both directions. The SETUP message will specify QoS requirements in the direction A to B to accommodate the service specifications carried in the *Resv* message. Conversely, it will not request any QoS or bandwidth guarantees from B to A since there is no data flow in this direction. While the VC setup is now handled by the egress router, it is still necessary to forward the *Resv* message to the ingress router, so that it can propagate that information upstream (it cannot be accurately inferred for the traffic and QoS parameters carried in the SETUP message). In order to do that, *Resv* messages including refreshes for reliability purposes, will keep on being forwarded onto the IP route. However, as with the method of Section 2.2, they are not acted upon at intermediate routers. Another alternative is to include the *Resv* message as higher layer information in the SETUP message.

The main advantage of this scheme is that it is consistent with the preferred solution for multicast flows when the LIJ capability of UNI 4.0 becomes available (see Section 3.4 for details). As a result, we recommend that it be the solution of choice in the UNI 4.0 environment, while the solution of either Section 2.2 or Section 2.3 should be used with UNI 3.1.

2.5 Handling of failures and route changes

The handling of failures in both the ATM and IP domains is important, as is the ability to react to changes in the IP routing. It is particularly important to avoid the formation of persistent routing loops that may be caused by interactions between the ATM and IP level paths. As a general rule, this is achieved by giving precedence to the IP level mechanisms to decide when to tear-down a connection or establish a new one.

To detect connection failures in the ATM domain, we rely on the ATM mechanisms based on OAM flows [For94] and hard connection states that the ATM network maintains.

Robustness against IP route changes is ensured by preserving the ability to detect such changes and eventually reflect them on any underlying ATM connection. In the context of the solutions of Sections 2.2 and 2.4, this is achieved by having *Path* messages continue to flow along the IP routed path. As a result, whenever the IP routing changes, the *Path* messages will automatically follow the new route. If the routing change does not affect the ingress and egress routers, the RSVP session remains undisturbed. However, if the *Path* message reaches a new egress router the RSVP session must be modified. This modification will be triggered when the ingress router A receives a *Resv* message from a new egress router C. In response, router A sets up a new VC to C, and initiates the tear-down of the VC to the previous egress router B.

Note that because this scheme relies on refresh *Path* messages to detect route changes, it is susceptible to transient loops. However, the duration of the transient loop is limited by the refresh period and the amount of data/packet loss (and the network impact) can be bounded by adjusting the refresh period based on the flow characteristics. This can be accomplished quite easily by indexing the refresh period to the source traffic specification contained in the RSVP messages.

In the case of the solution of Section 2.3, robustness to changes in IP routes relies on NHRP. However, while the current NHRP mechanism works well when A and B belong to the same IP routing domain, its application to a more general environment is complicated. It has been shown that, under certain circumstances, the use of NHRP for the general router-to-router case can lead to the creation of persistent routing loops. Rekhter and Cole [RK95] have proposed three approaches to extend NHRP for this general environment:

1. terminate NHRP at IP routing boundaries;
2. maintain NHRP state at routing boundaries for queries that pass through;
3. detect routing changes using refresh messages between the ingress and egress routers.

Clearly, the first approach is the simplest and requires little state information or refreshes and hence, their recommendation is to use this approach for the general environment. However, this solution requires additional IP hops through the ATM network, which is what we are trying to avoid for RSVP flows with QoS requirements. As a result, we advocate that this first approach, which is simple, be used for NHRP queries without specific QoS requirement. On the other hand, queries which contain QoS information, as is the case in the approach of Section 2.3, should be processed using the third approach to obtain the maximum shortcut. In this case, NHRP refresh queries essentially replace the refresh *Path* messages used in the solutions of Sections 2.2 and 2.4.

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 that the data flow traverses the ATM network in order to reach the receivers of data, which 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 first examine how to resolve IP multicast addresses to ATM addresses. Then, we consider three models of RSVP support over ATM networks. The "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. We describe the two approaches first for the case of single-sender multicasts. We consider next the interplay between RSVP 'soft' state and ATM 'hard' state in a discussion on handling failures and route changes. Finally, the general case of multiple-sender multicasts is covered in a discussion on handling RSVP filters.

3.1 Multicast address resolution

When multicasting over an ATM network a mechanism is needed to resolve IP group addresses to the corresponding ATM addresses. For example, when referring to Figure 3, forwarding the first *Path* message received at A to the next IP hop E, requires that the multicast group address be mapped to the ATM address of E. At times routers are statically configured with PVC connections, and then the routing of RSVP messages can proceed without this address resolution step. In the general case, however, an SVC would have to be set up between two routers or between a router and a host attached to the ATM network, in which case the address resolution step is required. We describe below two methods for resolving IP group address to ATM addresses. The term 'host' is taken here to mean either a host or a router.

In the first method, one starts by mapping the IP group address to the set of IP addresses of member hosts on the LIS. Milliken [Mil95a] uses the fact that the ATM subnetwork is not a broadcast medium and modifies the IGMP mechanism. In his scheme the router elicits IGMP reports from hosts on the LIS. The IGMP reports sent by hosts do not get retransmitted to other hosts and thus hosts are not aware of other hosts' presence and are thus forced to send their own IGMP report. After collecting the IP addresses of hosts in the multicast group the router can map these addresses to ATM addresses by using the ATMARP server on the LIS. A known problem with this method is its specificity to the IP protocol and IGMP.

An alternate method is to use the MARS address resolution mechanism [Arm95], which generalizes the ATMARP server to multicasting. Every host is registered with the MARS (Multicast Address Resolution Server). When an IP multicast address needs to be resolved the host sends a MARS_REQUEST message to the MARS, and gets in return a list of ATM addresses. The rest of this section does not depend on the scheme used for multicast address resolution. Either of these two schemes, or possibly other schemes, could be used.

3.2 "Classical" RSVP support

We extend now 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 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 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 [Mil95a] suggests that *Resv* refreshes are not needed, since the RSVP 'soft' state has been replaced in the ATM environment by a 'hard' state. Following [Mil95a], non-refresh *Resv* messages will be sent only if the QoS parameters of the flow change.

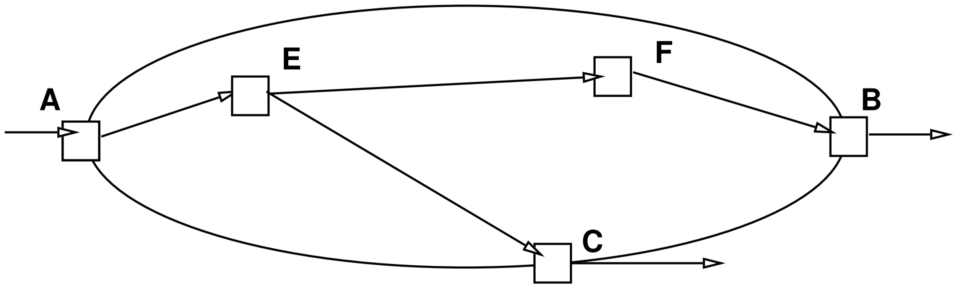

Figure 3: Reservation setup using "classical" RSVP support

3.3 Root-initiated ATM shortcuts

We start by extending the unicast scheme of Section 2.2 to single-sender multicast flows, as illustrated in Figure 4. As mentioned before, this is the approach best 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.

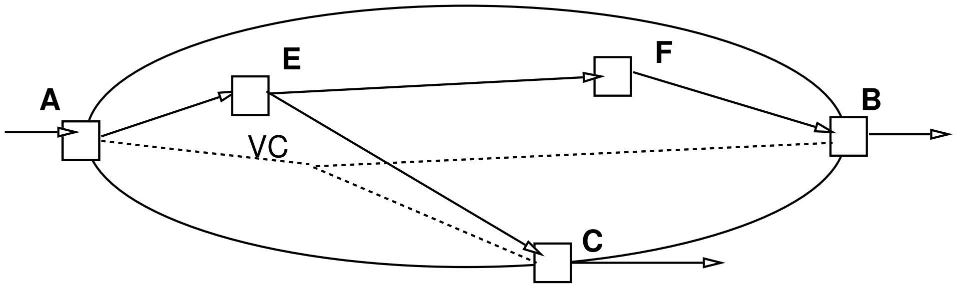

Figure 4: Reservation setup with maximum shortcut

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

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

towards A. Specifically, they do not merge *Resv* messages and do not perform any reservation. As in the unicast case, an alternative is to tunnel the *Resv* message directly to A by encapsulating it into an IP message. 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 signalling 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 between the ingress and egress routers. The use of a maximum shortcut minimizes IP-level processing at intermediate nodes and thus shortens end-to-end packet delays, but the (signalling) load imposed on the ingress router may become a problem when dealing with large multicast groups.

The avoidance of this potential problem was one of the motivations for the following scheme suggested by Milliken [Mil95a]. Its intention is to partly alleviate the problem by distributing the (signalling) processing load among the routers. This load distribution is achieved by allowing some flexibility at each router on deciding whether or not to extend an ATM shortcut. For example, an intermediate router, such as E in Figure 4, may choose between two options when it receives non-refresh *Resv* messages from any one of its descendants in the multicast routing tree (C and F in this example). It can follow the maximum shortcut scheme, and simply forwards the *Resv* messages upstream. However, it can also decide to merge two or more *Resv* messages. When such a merge takes place, the router then sets up a point-to-multipoint VC with itself as the root, and the leaves being the nodes whose ATM addresses were carried by the *Resv* messages. In a way, the intermediate router resets the shortcut processing, so that it now becomes a leaf by including its own ATM address in the merged *Resv* message that it forwards upstream. Note that all *Resv* messages in the above description correspond to a single session.

There are many possible heuristics that a router can follow to determine which of the two above options to select. The approach suggested by Milliken is that an intermediate router forward up to N distinct *Resv* messages upstream, where N is an integer parameter. Subsequent *Resv* messages are not forwarded, but instead are merged as described above. Since the number of *Resv* messages propagated upstream has now been reduced, the processing load on the upstream nodes will be reduced as well. The scheme is suggested as an experimental device in [Mil95a], one whose effectiveness needs to be evaluated after some use.

A more promising and systematic approach to eliminate the possibility of signalling overload at the ingress router, it to use the Leaf-Initiated Join (LIJ) capability of of UNI 4.0. We discuss such a solution in the next section.

3.4 Leaf-initiated ATM shortcuts

Consider the ATM network in Figure 4 and assume the flow of *Path* messages is as described in the previous section. That is, *Path* messages continue to use the default IP routed path, and a mechanism such as MARS is used for local multicast delivery on this path. As before, the *Path* message is not processed at intermediate routers, i.e. no state is kept and the PHOP is not modified, and it is extended at the ingress router A to carry the ATM address of A. In addition, A also chooses a "global connection identifier" 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 for generating an LIJ request to the GCID it 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, but the ingress router A is not notified of this new join. This eliminates the potential processing overload at router A since it is only required to handle a single signalling request, i.e. when the first leaf joins.

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 as in Section 2.4, even refreshes are sent in order to ensure reliable delivery of *Resv* messages to the ingress router.

3.5 Handling of failures and route changes

Consider an RSVP-based multicast data flow which traverses an ATM network, and for which an ATM point-to-multipoint connection has been established. As for unicast connections, the issue is to ensure that the ATM and IP failure and recovery mechanisms interact consistently, so that permanent loops are avoided and robustness against failures is provided whenever feasible. Failures can take place in both the ATM and in the IP domains, and when this happens there will be failure detection and recovery activity at both levels. As before, precedence is given to IP level mechanisms when dealing with error recovery procedures.

At the IP level, the soft-state mechanism of *Path* refreshes is used to recover from route changes and failures. When an egress router stops receiving *Path* refreshes, it can conclude that it is no longer on the path to the destination and remove itself, directly (with LIJ) or indirectly, from the ATM multicast connection. On the other hand, if the ATM connection is broken while the RSVP session is still in place the egress router will attempt to re-establish the ATM connection by sending a *Resv* message to the ingress router and re-initiating the leaf initiated join.

3.6 Handling of RSVP filters

We consider here multiple-sender multicasts and examine the impact of RSVP filters on the reservation setup models. Figure 5 shows a multicast flow with three sources, S1, S2 and S3, traversing an ATM network. The egress router B receives a *Resv* message from R, the receiver of the multicast.

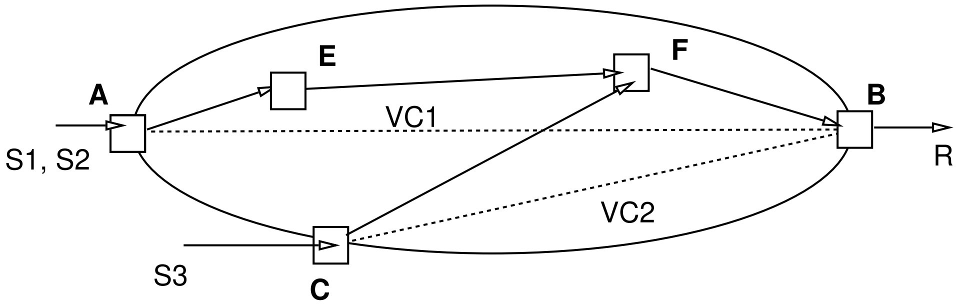

Figure 5: ATM shortcuts for an RSVP filter spec

Assume first that the *Resv* message at B specifies a Fixed Filter (FF) involving the three sources. The egress router B maps this FF request into two separate FF requests, one for $S1, S2$ and the other for $S3$. These FF requests are then forwarded upstream towards the ingress routers. Separate VCs would be setup for the three flows either by the ingress routers or by the egress router (in LIJ).

Assume now that the *Resv* message received at B specifies a wildcard filter (WF) involving the three sources. Wildcard filters allow the sharing of resources across multiple flows as specified in the *Resv* message. This sharing is, however, only possible if the different flows are indeed routed on a common link. This is a valid assumption in an IP-only environment where *Resv* (and *Path*) messages travel on the same route as the data flow since routing decision are made independently of the flow resources and services requirements. This then implies that flows from different senders but destined to the same address (unicast or multicast), are likely to be routed on paths with significant overlap (of IP links).

The situation is, however, quite different when flows are to be routed across ATM networks. This is because the path assigned to a given flow is selected as a function of the flow traffic parameters and service requirements. Furthermore, since paths for different flows are likely to be requested and generated independently of each other, i.e. often triggered by requests from different receivers, the potential for overlap and sharing of resources on ATM links is, therefore, much more limited. In addition, ATM currently does not provide the ability, in terms of signalling, to specify the sharing of resources between multiple ATM connections, that is implied by wildcard filters. Furthermore, even if this ability was provided, it would also require that ATM switches on the path of such connections be capable of enforcing this sharing, i.e. controlling the number of cells sent. It is not clear if and when such capability might become ubiquitous in ATM switches as it requires tight per-connection queueing and scheduling.

As a result, requests that specify wildcard filters can currently only be handled in one of two ways. The wildcard filter can be translated into a set of fixed filters, one for each individual flow, and the creation or joining of an ATM connection is then performed for each of them. This amounts to ignoring the possibility of resources sharing within the ATM network, which may be acceptable when, for example, most of the flows specified in the wildcard filter have already been established across the ATM network as a result of (fixed filter) requests from other receivers. In such configurations, the potential for sharing within the ATM network is relatively small. Note that this also applies to an IP-only environment. The second possible approach for handling wildcard filters, is to simply defer such requests to the IP overlay network. This clearly preserves all the sharing potential, but suffers from the obvious drawbacks of requiring IP level processing at each hop within the ATM network, and of forcing flows to be routed on links where resources might be constrained. Such an approach may, however, be the preferred one for sessions with large numbers of senders and receivers, where typically only one sender is active at the time, e.g., large audio conferences.

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 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 and QoS specifications are not defined in RSVP. They are deferred to the int-serv IETF draft documents. The Guaranteed Delay int-serv draft [SP95] defines the traffic specification (TSpec) as consisting of a token bucket with a given bucket depth b (in bytes) specifying the maximum allowed burst size for the flow, a bucket rate r (in bytes/second) giving the average rate of the flow. The combination of bucket depth and rate defines a maximum traffic envelope for the flow, that is similar to the leaky bucket based ATM GCRA (Generic Cell Rate Algorithm). Policing of the flow can then be performed based on this traffic specification, i.e. the traffic must obey the rule that for any time unit T, the amount of data sent cannot exceed $rT + b$.

In addition to the two bucket parameters, the TSpec also provides a minimum packet size p (in bytes). Currently, no peak rate is specified, and instead a maximum packet rate is defined. It is meant to ensure that the number of packets received per time T, does not exceed $(rT + b)/p$. Note that this does not precludes packets of size smaller than p as such situations can occur because of fragmentation inside the network. However, the primary purpose of specifying a maximum packet rate is to ensure that the packet processing overhead never exceeds its maximum expected value. The specification of a peak rate was proposed in [GGPRS95b], where the TSpec is extended to also include a peak rate P (in bytes/sec) for the flow. This quantity may, therefore, eventually be available from the TSpec.

In ATM, traffic specifications are given by means of a Traffic Descriptor, that includes the following parameters [For94, Sam95]: the Sustainable Cell Rate (SCR) in cells/second gives the average rate of the call; the Peak Cell Rate (PCR) in cells/second determines the minimum time interval between consecutive cells; and the Maximum Burst Size (cells) corresponds to the maximum allowable number of consecutive cells at the peak rate. Policing is also specified in relation to these parameters through the GCRA. The GCRA essentially corresponds to two leaky buckets in series, with the first policing SCR and the second controlling PCR. In controlling PCR, the ATM network allows for some fluctuations in the inter-cell arrival times through the specification of a Cell Delay Variation Tolerance (CDVT) parameter.

A first, simplifying step towards identifying the appropriate relations between RSVP and ATM traffic descriptors, is to assume a perfect fluid model for both the IP flow and the ATM call. If we then ignore the potential impact of the granularity of the ATM cell size and of the segmentation overhead, the following relations can be established:

$$\mathrm{SCR} = \frac{r}{48}\,; \quad \mathrm{MBS} = \frac{b}{48}\,; \quad \mathrm{PCR} = \frac{P}{48}\,, \tag{1}$$

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 actual peak rate of the flow, if this quantity has been specified. Note that if the TSpec does not include a peak rate, selecting the access link speed as a default value for the ATM PCR will usually be pessimistic and lead to inefficiencies and unnecessary rejection of requests. See, however, the discussion of Section 4.3 for possible alternatives that can remedy this problem to some extent.

As mentioned before, the above expressions need to be adjusted to properly reflect the impact of the ATM segmentation in fixed size cells. There are two types of adjustments which are required. The first is to account for the fact that packet sizes need not be integer multiple of cell payloads, i.e. the last cell is typically not full. The second is to include any overhead introduced by the segmentation layer, which depends on the AAL type. If AAL5 is used, the overhead should be minimal (8 bytes in the last cell of a message).

Determining the exact amount by which the above expressions need to be adjusted is, however, not a simple matter as it depends on the entire packet sizes *distribution*. For example, if all packets are 41 bytes, then assuming AAL5, two cells are needed to transmit each of them (the 8-byte trailer and the 41-byte payload do not fit in a single cell). In this case, equation (1) is off by more than a factor 2. On the other hand, if all packets are 280 bytes long, then six cells (288 bytes) suffice to transmit them and equation (1) is very accurate. There are two possible approaches to resolve this problem. The first is to be conservative, while the second is to rely on some approximation.

A conservative estimate can be obtained based on the minimum packet size information provided in the RSVP TSpec. The basic idea is that while the error in equation (1) is not a continuous function of packet size (one cell for 40-byte packets and two cells for 41-byte packets), it is nevertheless a "regularly" decreasing function of packet size. Specifically, it has a saw-tooth behavior with jumps for every packet size of the form $n \times 48 - 7$ bytes, i.e. the number of cells needed goes from n to $n+1$ at these values. Based on this, conservative estimates for SCR, MBS, and PCR can be obtained by assuming that all packets are of minimum size and always require the maximum possible number of cells, i.e. are just after a jump. The corresponding relations are then of the form:

$$\mathrm{SCR} = \alpha r \ ; \quad \mathrm{MBS} = \alpha b \ ; \quad \mathrm{PCR} = \alpha P \ , \tag{2}$$

where $\alpha = (1 + \lceil p/48 \rceil)/p$ represents the worse case overhead due to the ATM segmentation with AAL5 and a minimum packet size of p (in bytes). Note that if some traffic shaping is performed at the IP/ATM boundary as it probably should, the value of PCR could actually be lowered (more on this in Section 4.3).

Equation (2) is conservative mostly for small values of p. In such cases, it may be preferable to use a simple approximation based on measuring two basic parameters, the number of bytes and packets received in a given time interval T. Specifically, let B be the number of bytes received in T and N the corresponding number of packets. SCR and the number $K(T)$ of ATM cells needed to transmit the B bytes and N packets can be approximated by

$$\mathrm{SCR} = \frac{K(T)}{T} \approx \frac{B + 8N}{48T} + \frac{aN}{T} \ , \tag{3}$$

where a is a parameter which can be tuned to adjust the approximation. Specifically, a value of $a = 1$ always yields an upper bound for $K(T)$, while a value of $a = 0$ gives a lower bound. The above expression can be further simplified, if B/T is replaced by r and N/T by r/p which in both cases also provide upper bounds. The above approximation is typically quite accurate, and a value of $a = 0.5$ was found to provide reasonably accurate results for a number of LAN traffic traces. A similar approach can be used for MBS and PCR, although as mentioned earlier PCR can probably be easily adjusted through reshaping.

4.2 Mapping of QoS guarantees and service specifications

In this section we address the problem of translating RSVP service specifications into corresponding ATM QoS guarantees. In addition, we also articulate how this mapping is performed and its relation to the standard RSVP process based on the combination of the *Path* and *Resv* messages. Note that as for traffic specifications, service specifications are not defined in RSVP. They are again deferred to the int-serv IETF Guaranteed Delay draft [SP95]. The document defines the service specification (RSpec), and how it is determined as a function of the delay requirements of the flow and the capabilities/characteristics of the routers (service elements) on its path. An important issue to be addressed at the boundary between IP and ATM networks, is then to define how the ATM network is to participate in this process. The problem is best understood by first going through an example illustrating the steps followed during the establishment of an RSVP flow.

The end-to-end delay d and the associated service specifications for the flow are not quantities that are initially provided explicitly. Rather, they are determined at the receiver upon receipt of the *Path* message carrying the values of the "error terms" $\mathcal{C}_{tot} = \sum_S^D \mathcal{C}_i$ and $\mathcal{D}_{tot} = \sum_S^D \mathcal{D}_i$, which have been accumulated on the connection's path. The term $\mathcal{C}_i$ and $\mathcal{D}_i$ correspond to the error contributed by router i when compared to a perfect fluid service model. For example, $\mathcal{C}_i$ is the MTU size for that flow if Weighted Fair Queuing scheduling is used, and $\mathcal{D}_i$ accounts for local, propagation, and transmission delays. Given the quantities $\mathcal{C}_{tot}$ and $\mathcal{D}_{tot}$ and a desired upper bound d on the end-to-end delay, the service specification provided by the receiver in the *Resv* message, consists essentially of a clearing rate $R \geq r$. The clearing rate R is to be allocated at each router on the path, so that the guaranteed end-to-end delay $\tilde{d}$ for the flow verifies:

$$\tilde{d} \leq \frac{b + \mathcal{C}_{tot}}{R} + \mathcal{D}_{tot} \leq d \tag{4}$$

As a result, a buffer clearing rate R is to be reserved at each router on the path in order to guarantee the required delay d.

Although not directly stated in [ZDE+93] or [SP95], these documents suggest that the resource reservation for a flow from S to D with guaranteed delay requirement is performed in the following way. The source S generates *Path* messages that contain the traffic characterization (TSpec)of the flow. The *Path* message, therefore, includes the parameters b,

r, p, and two fields $\mathcal{C}_{tot}$ and $\mathcal{D}_{tot}$. which are both initialized to 0. At router i, these fields are incremented using the local values $\mathcal{C}_i$ and $\mathcal{D}_i$:

$$\mathcal{C}_{tot} \leftarrow \mathcal{C}_{tot} + \mathcal{C}_i; \quad \mathcal{D}_{tot} \leftarrow \mathcal{D}_{tot} + \mathcal{D}_i \quad .$$

At the receiver D, a desired end-to-end delay d is selected, and the required clearing rate R is computed from equation (4) as:

$$R = \max(r, \frac{b + \mathcal{C}_{tot}}{d - \mathcal{D}_{tot}}) \quad .$$

The clearing rate R is then loaded in the RSpec included in the *Resv* message sent towards S.

Note that while the above approach does yield a value for the clearing rate R such that the desired end-to-end delay guarantee is met, this could have typically been achieved with a lower value. This is because the above approach amounts to assuming an infinite peak rate for the flow. If as proposed in [GGPRS95b], the TSpec is expanded to also include the peak rate P of the flow, a lower value can then be determined for R. Hence, the inclusion of a peak rate term in the TSpec can not only facilitate interactions with ATM (as discussed in Section 4.1), but also improve overall efficiency.

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 $\mathcal{C}_{\mathrm{ATM}}$ and $\mathcal{D}_{\mathrm{ATM}}$, when updating the $\mathcal{C}_{tot}$ and $\mathcal{D}_{tot}$ fields in the *Path* message. This is difficult for two reasons. First, the correct value is a function of the path through the ATM network, and this is not known at the time the *Path* messages 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 [SP95], 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 signalling. 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 $\mathcal{D}_{\mathrm{ATM}}$, and further to comply with this advertised value when an ATM connection actually needs to be setup upon receipt of a *Resv* message. There are a number of possible alternatives to address this problem, that involve different trade-offs in terms or efficiency versus complexity. In the rest of this section, we review them for both unicast and multicast flows and articulate the pros and cons of each approach.

4.2.1 Unicast flows

The case of a unicast flow is illustrated in Figure 2. From our earlier discussion, we know that after the *Path* message starts from S it accumulates the quantities $\sum \mathcal{C}_i$ and $\sum \mathcal{D}_i$ from each router it traverses, so that upon arriving at router 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 as described in Section 2, the *Path* message 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 to be resolved is to identify the router which is responsible for determining the value $\mathcal{D}_{A,B}$, and updating the *Path* message accordingly. In the case of a unicast flow, 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 as is described in Section 2.4.

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 simply generates an estimate for the delay $\mathcal{D}_{\mathrm{ATM}}$ from router A to itself across the ATM network. This estimate can be a pre-configured value or could possibly be inferred from information made available by the ATM network.

For example, if router B has a PNNI interface to the ATM network, it will then have access to the PNNI topology database [For95]. This database contains link metrics from which it can obtain a reasonable delay estimate for a connection between A and itself, with traffic parameters as specified in the TSpec.

The advantage of this approach is its simplicity and the fact that it avoids the exchange of signalling messages with the ATM network. Its main disadvantage is its potential inaccuracy and lack of flexibility, especially in the case where a PNNI interface is not available and the advertised quantity is pre-configured.

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. (Note that B can initiate such a connection.) This interaction clearly involves additional overhead, but this overhead can be minimized by caching returned CMCTD values. The main issue is, however, the selection of the delay requirement ¸ to be specified in the connection request.

Based on the current signalling specs [Spi95a], the user can specify in the SETUP message a Desired Maximum Cell Transfer Delay (DMCTD) for both the forward and backward directions of a connection. As the SETUP is routed through the ATM network and passes through different switches, each switch determines the maximum local delay it can guarantee in each direction and allocates resources accordingly. A Cumulative Maximum Cell Transfer Delay (CMCTD) field for each direction is carried in the SETUP message, and incremented by the local values at each switch. This provides a mechanism to track the evolution of the cell transfer delay. Once the SETUP message reaches its destination, i.e. the ingress router A, the value carried in the CMCTD field for the backward direction identifies the delay guarantee that the network is actually able to provide to the connection. This value is then returned to the calling party, the egress router B, in the CONNECT message. This returned value can then be used by router B as its estimate for $\mathcal{D}_{A,B}$. Note that since in this case the value is also known to router A (it was included in the SETUP message), it may not be necessary to include it as well in the first *Resv* message sent by router

B. However, it is recommended to continue doing so as depending on the difference between DMCTD and CMCTD arriving at A, the ATM network may adjust, i.e. increase, the CMCTD value carried in the CONNECT message. This means that the final value communicated to router B could differ from the one initially received by router A.

The main issue in the above procedure is for router B to determine which value of DMCTD to specify in the connection request. A default value could be used, but the benefit of querying the network would then be minimal. Preferably, the request should be for the "best" possible delay. Such a capability is not readily supported by the ATM signalling, but may be available in future extensions. For example, if it is possible to indicate that DMCTD is a "soft" value, i.e. the connection should not be rejected even if the specified value cannot be guaranteed. Router B could then safely choose an aggressive value, e.g., DMCTD = 0. The returned CMCTD value, typically larger than the required DMCTD, would then give the best feasible delay guarantees between A and B. A similar result would also be achieved by allowing the specification of both a Desired and and an Acceptable MCTD. This was indeed included in the first version [Spi95b] of the ATM signalling specifications. Setting the desired field to 0 and the acceptable field to ∞, again results in the network returning the best possible delay value for the connection.

After router B has obtained a value for $\mathcal{D}_{A,B}$ and used it to update the $\mathcal{D}_{tot}$ field in the *Path* message, it remains to decide what to do with the ATM connection which was setup in order to obtain that information. The simplest solution is to disconnect it. This minimizes the amount of resources wasted (and paid for ...), but introduces a dependency on the state of the ATM network at the time the request was made. For example, if the network was unusually idle, the value returned for CMCTD would be much better than a typical one, and the network may not be able to match this guarantee later when the *Resv* arrives and the actual connection needs to be established. This may, however, be acceptable since even without crossing ATM networks, RSVP connections can be rejected for lack of available resources. Another solution is to keep the connection alive. This ensures that the advertised guaranteed delay can be provided when the *Resv* message eventually arrives. The cost is a potentially significant wastage of resources. As a result, disconnecting the connection may be the preferred approach.

One feature common to the two above solutions 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 not included in the current specifications [SP95], but can be provided through a simple extension as described in [GGPRS95c]

[GGPRS95c] proposes to add a slack field $\mathcal{S}$ to the RSpec in the *Resv* message. 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.

4.2.2 Multicast flows

Multicast flows share many of the same problems as unicast flows when it comes to mapping RSVP QoS guarantees onto corresponding ATM quantities. For example, they also face the problem of determining which delay guarantees to advertise for the ATM network in the *Path* messages. Similarly, they too can benefit from the availability of a slack term in *Resv* messages, to deal with potential inaccuracies in selecting an appropriate delay estimate for the ATM network. Because of similarity in the issues and arguments, we do not embark on any further discussion on these items in this section. Instead, we focus on aspects for which significant differences exist between the multicast and unicast cases.

A first major difference between the unicast and multicast cases is in terms of where an estimate for $\mathcal{D}_{\mathrm{ATM}}$ can actually be obtained. In the unicast case, this could be performed at either the ingress or the egress routers. This is not true for multicast flows as even if the ingress router was able to identify all the ATM addresses associated with the multicast address, this would still not be sufficient. Specifically, it is likely that different ATM addresses would yield different values for $\mathcal{D}_{\mathrm{ATM}}$, and those could not be differentiated through a single *Path* message, unless only the largest one was specified. This could be very inefficient as it might force some receivers to request a much greater clearing rate than

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

necessary. As a result, the determination of an estimate for $\mathcal{D}_{\mathrm{ATM}}$ must be performed at the egress routers for multicast flows. This also offers the advantage of better distributing this function. Note that each egress router determines an estimate for $\mathcal{D}_{\mathrm{ATM}}$ 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 $\mathcal{D}_{\mathrm{ATM}}$. It should, however, only be of limited significance.

A second difference between unicast and multicast flows is in terms of how the information is to be 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 signalling available to establish point-to-multipoint connections.

Root-initiated point-to-multipoint ATM connection 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 $\mathcal{D}_{\mathrm{ATM}}$ 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. Note that this works well if the root of the point-to-multipoint connection is the actual ingress router for the ATM network, i.e. a maximum ATM shortcut is used. If as discussed at the end of Section 3.6, "intermediate" roots are created to improve scalability, additional difficulties are introduced.

In this case, each intermediate roots has to determine which of the portion[5] of the advertised delay budget $\mathcal{D}_{\mathrm{ATM}}$ it should use, and more important which single value to include in the merged *Resv* message forwarded to the original ingress router. Basically this value should be the *smallest* of all the residual advertised delay budgets. This is the delay that has to be guaranteed between the original ingress router and the intermediate root, in order to ensure that the total delays to all the egress routers are consistent with the values they advertised. All this introduces significant additional complexity and inefficiencies. Therefore, despite the potential scalability problem of having the ingress router handle all call setup requests (ADD-PARTY), our recom-

[5]The delay value $\mathcal{D}_{\mathrm{ATM}}$ which was computed by each egress router, assumed the original ingress router as the root of the ATM connection.

mendation is to not merge the *Resv* messages and to always use a maximum ATM shortcut.

Leaf initiated join (LIJ) to a point-to-multipoint ATM connection This capability is to be available in UNI 4.0, and allows a leaf, an egress router, to directly request to join a point-to-multipoint without notifying or involving the root of the connection. This eliminates the previous scalability problem. Egress routers join the point-to-multipoint connection by specifying its global identifier (GCID) as well as the desired delay guarantee $\mathcal{D}_{\text{ATM}}$ they determined and advertised in the *Path* messages. This is clearly the recommended solution, which should be used as soon as the LIJ capability is available. However, note that while UNI 4.0 defines the LIJ capability, it does not yet specify if different guarantees can be provided to different leaves. The unavailability of such a feature would clearly be a major problem in supporting multicast RSVP flows.

4.3 Handling of changing flow specifications

In this section, we briefly address the handling across ATM networks of the flow reservation changes that RSVP allows. Specifically, RSVP allows receivers to change at any time the service specification carried in the *Resv* messages for a given flow. Request for such changes should essentially be transparent to the ATM network since it does not fully participate in the resource (clearing rate) allocation process triggered by a reservation request. Specifically, recall that the ATM network is handled in the *Path* advertising phase as a *fixed* delay component. Hence, it is unaffected by any change in the requested clearing rate R. This certainly simplifies the handling of RSVP reservation changes in ATM networks, but also points to a limitation in terms of the flexibility with which ATM networks can support RSVP services.

It is possible to remedy this problem to some extent using an approach based on the method of [GGPRS95a]. The basic idea is that the peak rate of an RSVP flow can actually be limited at the entrance of an ATM network to the clearing R, without impacting the end-to-end delay guarantees. The ATM network could, therefore, perform reshaping of the flow to a peak of R as per the clearing rate value specified in the received *Resv* message. Since lower peak rate connections are easier to deal with and typically require the allocation of fewer resources for a given delay guarantee, the ATM network could take advantage of this. In particular, it could keep the delay guaranteed to a given connection constant, i.e. equal to the value advertised in the *Path* message, but adjust its allocated resources as a function of the specified RSVP clearing rate. This requires reshaping of the traffic to ensure that its "peak rate" indeed complies with the clearing rate, as well as the ability to signal such changes to the ATM network.

While the above approach lets ATM networks take advantage of changes in RSVP reservations, the ability to do so may not always be available from the ATM signalling. In particular, UNI 3.1 does not allow the characteristics of a VC to be changed once it is established. A possible alternative may then be to teardown the existing VC and setup a new VC in the ATM segment of the flow, but this can be expensive if user data interruption is to be eliminated (buffering and/or two coexisting calls will be needed). UNI 4.0 should allow for dynamic changes of call characteristics, and the above approach may then be readily supported.

Finally, note that using the requested clearing rate R as the peak rate for the connection, can also help improve the likelihood that the ATM network can guarantee the advertised delay $\mathcal{D}_{\mathrm{ATM}}$ when the connection request is eventually generated. This is because when $\mathcal{D}_{\mathrm{ATM}}$ was determined the requested clearing rate was not yet known and, therefore, $\mathcal{D}_{\mathrm{ATM}}$ was obtained assuming a higher peak rate, i.e. the access link speed or the peak rate value specified in the TSpec, if available. The final decision on which peak rate to specify to the ATM network when the connection request is generated depends on a number of other factors, such as the reshaping and buffering abilities at the access point, i.e. the ingress router, and is likely to vary for each implementation.

5 Summary and future work

In this report, we have identified issues in extending RSVP flows requiring Guaranteed Delay service across ATM networks, and proposed approaches to achieve that. We have considered the cases of both unicast and multicast RSVP flows, and also the impact of the different versions of the ATM signalling specifications.

The approaches we proposed involve a number of modifications to the RSVP protocol and in some instances extensions to the ATM signalling. In as much as possible, we have tried to provide solutions that are common for unicast and multicast flows as well as for the different versions of ATM signalling. However, because of the significant differences that exist between the ATM signalling specifications of UNI 3.1 and UNI 4.0, different approaches had to be used in some instances.

5.1 RSVP modifications for UNI 3.1 environment

In this environment, the general approach we take can be characterized as *root oriented.* This means that most of the interactions with the ATM signalling needed to extend RSVP flows across ATM networks, originate in the ingress router. Such extensions require a number of modifications to the processing of *Path* and *Resv* messages.

The first step at the ingress router is to identify that the flow is to cross an ATM network and should, therefore, be handled differently. Once this has been determined, subsequent modifications to the *Path* message handling vary somewhat as a function of the approach used. Typically, the *Path* message will be forwarded on the normal IP path, and extended to carry the ATM address of the ingress router. *Path* processing is also different at intermediate (non-egress) routers which do not update the PHOP field, so that it still points to the ingress router, and do not maintain state information. This helps lower the processing overhead for such messages. In addition, the $\mathcal{D}_{tot}$ field (and $\mathcal{C}_{tot}$) is not updated until the *Path* message reaches the egress router(s), where it is incremented by an estimate of the maximum delay the ATM network would contribute. *Path* messages continue flowing on the IP route even after an ATM VC shortcut has been established for the flow.

In the unicast case, we also outline a possible alternative to the forwarding of *Path* messages, that relies on NHRP mechanisms to forward the *Path* information to the appropriate egress router. This eliminates the need to modify *Path* processing at intermediate routers, and allows leveraging of mechanisms that may be available. However, this approach cannot be extended to multicast flows.

The processing of *Resv* messages is also affected when crossing ATM networks. They are used to trigger the establishment of an ATM shortcut when received at an egress router(s). The connection request originates from the ingress router (ADD-PARTY for multicast flows, or SETUP for unicast flows) upon receipt of a new *Resv* message from an egress router. This *Resv* message carries the standard RSVP information, i.e. filter and service specifications, that are needed by the ingress router to forward *Resv* messages to its upstream neighbor. The *Resv* message also contains the ATM address of the egress router as well as the delay guarantees needed for the connection across the ATM network. Note that the receipt of the SETUP (or ADD-PARTY for multicast flows) at an egress router provides an implicit acknowledgment that the ingress router has received the *Resv* message and that the ATM reservation has been successful. Finally, refresh *Resv* messages are suppressed, i.e. not forwarded on the IP path, and connection liveness is guaranteed by ATM mechanisms.

5.2 RSVP modifications for UNI 4.0 environment

The major enhancement in UNI 4.0, from the point-of-view of RSVP support, is the LIJ ability in point-to-multipoint connections. This allows us to use a *leaf oriented* approach to support RSVP flows (both unicast and multicast) which ensures better scalability.

The handling of *Path* messages remain essentially as for the UNI 3.1 case, in that they are forwarded on the normal IP path but not processed at intermediate routers, i.e. PHOP field and OPWA objects are not modified and no state is created. In addition to carrying the ATM address of the ingress router, the *Path* message also carries a global ATM call identifier (GCID) in the case of multicast flows. This GCID is then specified in the LIJ message generated by egress routers upon receipt of a new *Resv* message, when they want to join an existing point-to-multipoint connection associated with a given multicast flow. In the case of a unicast flow, the egress router simply initiates a SETUP to the ATM address of the ingress router.

Because in the *leaf oriented* approach the egress routers are responsible for the establishment of ATM connections, it is not necessary to forward *Resv* messages to the ingress router for that purpose. However, it is still necessary to transmit the RSVP information contained in the *Resv* message (filter and service specifications) to the ingress router, so that it can propagate it upstream. This is achieved by forwarding all *Resv* messages (including refreshes for reliability) on the IP route to the ingress router. Note that although *Resv* messages are processed at intermediate routers they are not acted upon, i.e. merging of *Resv* messages will take place when required but no reservations will be triggered and no state is maintained.

5.3 ATM extensions and modifications

As stated above, it is clear that many of the extensions to be included in UNI 4.0 are key to an efficient support of RSVP flows across ATM networks. Foremost among them is the LIJ capability, which is critical to handle large multicast connections. This capability should, however, be such that different leaves are allowed to specify different service requirements. Another desirable extension is the ability to renegotiate the characteristics of an established connection. However, there other desirable extensions which may not be provided in UNI 4.0. For example, it would be helpful for an RSVP router to query the ATM network to find the *best* delay that can be guaranteed to a given destination. This can be achieved either by allowing "soft" requests, or by supporting both "desired" and "acceptable" QoS parameters. Similarly, the ability to let the root of a point-to-multipoint call assign a GCID even before any leaf has requested to join, could simplify some of the processing when establishing such calls.

5.4 Future works

The main purpose of this document has been to identify issues and outline potential solutions rather than finalize a particular design. Therefore, many of the items discussed here can be considered as areas for future work. However, there are clearly a number of topics for which much more remains to be done. For example, there is room for improvement in being able to better account for ATM networks 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 the ATM signalling and service specifications to better emulate the int-serv [SP95] model. Similarly, while the availability of a LIJ capability in the ATM UNI 4.0 clearly improves scalability, it does so by shifting the processing burden from the ingress router to the ATM network. It is, therefore, important to ensure that a scalable solution for managing large multicast groups be introduced in ATM, and only limited work has been done on this topic so far. Finally, as Section 3.6 clearly pointed out, the coupling in ATM between routing and service requests, makes it difficult to efficiently handle RSVP wildcard filters. It may be possible to improve this if ATM signalling introduces the notion of call correlation, and ATM switches are designed so as to be capable of taking advantage of this knowledge. However, much work remains to be done, in order to ensure a truly efficient handling of such cases.

References

[All95] A. Alles. ATM Internetworking. Cisco Systems, Inc., May 1995.

[Arm95] G. Armitage. Support for Multicast over UNI 3.1 based ATM Networks. Internet draft-ietf-ipatm-ipmc-06.txt, August 1995.

[Atk94] R. Atkinson. Default IP MTU for use over ATM AAL5. Internet RFC1626, May 1994.

[BCDB95] M. Borden, E. Crawley, B. Davie, and S. Batsell. Integration of Real-time Services in an IP-ATM Network Architecture . Internet RFC1821, August 1995.

[Bro95] C. Brown. Baseline Text for MPOA. ATM Forum 95-0824R1, July 1995.

[BZE+95] R. Braden, L. Zhang, D. Estrin, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification. Internet draft-ietf-rsvp-spec-06.ps, June 1995.

[Cal94] R. Callon. Integrated PNNI for Multi-Protocol Routing. ATM Forum 94-0789, September 1994.

[Can95] D. Canserver. NHRP Protocol Applicability Statement. Internet draft-ietf-rolc-nhrp-appl-01.txt, March 1995.

[CS95] R. Callon and B. Salkewicz. An Outline of Integrated PNNI for IP Routing. ATM Forum 95-0649, August 1995.

[CSV95] R.G. Cole, D.H. Shur, and C. Villamizar. IP over ATM: A Framework Document. Internet draft-ietf-ipatm-framework-doc-04.ps, July 1995.

[DB95] L. Delgrossi and L. Berger. Internet STream Protocol Version 2+. Internet draft-ietf-st2-spec-02.txt, March 1995.

[For94] ATM Forum. ATM User-Network Interface Specification. Version 3.1, September 1994.

[For95] ATM Forum. ATM PNNI Draft Specification. ATM Forum 94-0471 R10, July 1995.

[GA95] B. Gleeson and G. Armitage. Issues sorrounding a new encapsulation for IP over ATM. Internet draft-armitage-ipatm-encaps-02.txt, June 1995.

[GGPRS95a] L. Georgiadis, R. Guérin, V. Peris, R. Rajan, and S. Shenker. Issues in the Provision of Guaranteed Delay Service and the Use of Rate-Controlled Service Disciplines. Private communication, July 1995.

[GGPRS95b] L. Georgiadis, R. Guérin, V. Peris, R. Rajan, and S. Shenker. Proposal for the introduction of peak rate. Contribution to the int-serv mail list, August 1995.

[GGPRS95c] L. Georgiadis, R. Guérin, V. Peris, R. Rajan, and S. Shenker. Proposal for the introduction of delay slack. Contribution to the int-serv mail list, August 1995.

[Got95] Y. Goto. Session Identity Notification Protocol. Internet draft-goto-sinp-00.txt, July 1995.

[GS93] M. Graf and H.J. Stüttgen. An Internetworking Architecture for Multimedia Communication over Heterogeneous Networks. In *Dagstuhl Workshop on Communication*, 1993.

[Hei93] J. Heinanen. Multiprotocol Encapsulation over ATM Adaptation Layer 5. Internet RFC1483, July 1993.

[HP93] O. Hagsand and S. Pink. ATM as a Link in an ST-2 Internet. In *4th International Workshop, NOSSDAV*, 1993.

[KN95] Y. Katsube and K. Nagami. Mapping of IP Flow to Datalink Layer Connection. Internet draft-katsube-flow-mapping-dl-conn-00.txt, May 1995.

[KNE95] Y. Katsube, K. Nagami, and H. Esaki. Router Architecture Extensions for ATM: Overview. Internet draft-katsube-router-atm-overview-00.txt, March 1995.

[KP95] D. Katz and D. Piscitello. NBMA Next Hop Resolution Protocol (NHRP). Internet draft-ietf-rolc-nhrp-04.txt, May 1995.

[Lau94] M. Laubach. Classical IP and ARP over ATM. Internet RFC1577, January 1994.

[Mil95a] W. Milliken. Integrated Services IP Multicasting over ATM. Internet draft-milliken-ipatm-services-00.txt, July 1995.

[Mil95b] W. Milliken. IP Multicasting over ATM: System Architecture Issues. Internet draft-ietf-ipatm-arch-00.txt, July 1995.

[OEN95] M. Ohta, H. Esaki, and K. Nagami. Conventional IP over ATM. Internet draft-ohta-ip-over-atm-02.txt, March 1995.

[PL94] D. Perkins and F. Liaw. Beyond Classical IP - Integrated IP and ATM Architecture Overview. ATM Forum 94-0935, September 1994.

[PLM+95] M. Perez, F. Liaw, A. Mankin, E. Hoffman, D. Grossman, and A. Malis. ATM Signaling Support for IP over ATM. Internet RFC1755, February 1995.

[RK95] Y. Rekhter and B. Cole. Alternatives for router-to-router NHRP. Submission to the rolc mailing list, July 1995.

[RK95] Y. Rekhter and D. Kandlur. IP Architecture Extensions for ATM. Internet draft-rekhter-ip-atm-architecture-01.txt, July 1995.

[RV95] Y. Rekhter and C. Villamizar. Inter-Domain Routing over ATM Networks. Internet draft-rekhter- idr-over-atm-00.txt, February 1995.

[SA95] T. Smith and G. Armitage. IP Broadcast over ATM Networks. Internet draft-smith-ipatm-bcast- 01.txt, July 1995.

[Sam95] Pradeep Samudra. Draft of UNI Signaling 4.0. ATM Forum 94-1018 R4, July 1995.

[Sat95] S. Sathaye. Traffic Management Specification 4.0. ATM Forum 95-0013R7, July 1995.

[SP95] S. Shenker and C. Partridge. Specification of Guaranteed Quality of Service. Internet draft-ietf-intserv-guaranteed-svc-01b.txt, August 1995.

[Spi95a] E. M. Spiegel. Signalling of Individual QoS Parameters. ATM Forum 95-1009R1, October 1995.

[Spi95b] E. M. Spiegel. Signalling of Individual QoS Parameters. ATM Forum 95-1009, August 1995.

[Top90] C. Topolcic. *Experimental Internet Stream Protocol: Version 2 (ST-II).* Internet RFC 1190, October 1990.

[ZDE+93] Lixia Zhang, Stephen E. Deering, Deborah Estrin, Scott Shenker, and Daniel Zappala. RSVP: A New Resource ReSerVation Protocol. *IEEE Network*, 7(5):8–18, September 1993.