Internet Engineering Task Force
INTERNET-DRAFT
draft-berger-rsvp-over-atm-00.ps
Expires December 11, 1996

L. Berger
FORE Systems

RSVP Over ATM:

Framework and UNI 3.0/3.1 Method

June 11, 1996

Status of this Memo

Abstract

This note presents a method for running RSVP over ATM switched virtual circuits (SVCs). It presents an overall approach to the problem, as well as a specific method for running over today's ATM networks. The note includes a review of major related issues, an overview of the proposed approach, and a specific method usable with UNI 3.0 and 3.1. This note is intended to be used as a basis for discussion in the IETF ISSLL working group.

Author's Note

This note is an initial (rough) draft and is expected to be changed and extended based on discussions in the ISSLL working group. All comments and contributions are welcomed.

The postscript version of this document contains figures that are not included in the text version, so it is best to use the postscript version. Figures will be converted to ASCII in the next version.

1. Introduction

This note discusses running IP over ATM in an environment where SVCs are used to support QoS flows and RSVP is used as the internet level QoS signaling protocol. It begins with a general discussion on issues that are key to running RSVP over ATM. Some of these issues have been discussed in related material, others have not been widely discussed. All of the discussed issues must be addressed by any RSVP over ATM solution.

In Section 4, this note presents a specific RSVP over ATM solution. It specifies a method for running RSVP over ATM UNI 3.x (3.0 and 3.1). This note covers key aspects to running RSVP over UNI 3.x and includes an explicit set of processing rules and required RSVP related changes. (Changes are in the message processing and RSVP interface areas.) The defined method conforms to a framework that permits interoperability between simple, or baseline, implementations and more sophisticate, full-featured, implementations. The framework is described in Section 3 of this note.

The described framework is a general approach to running RSVP over ATM. The framework is intended to provide a structure to multiple specific RSVP over ATM specifications. Each specification will match technology or standardization evolution, e.g UNI 3.x and 4.x. The framework is also intended to allow interoperability between implementations that use different approaches in mapping RSVP to ATM. This allows for evolution without requiring uniform deployment. It also permits the specification of basic functionality, while allowing for more sophisticated options.

2. Issues

The general issues related to running RSVP [1] over ATM have been covered in numerous papers including [2], [3], and [4]. This section will review, and introduce two new key issues that must be addressed by any RSVP over ATM solution. The issues that will be covered are:

- VC Usage
- Heterogeneity
- Routing
- QoS Renegotiation
- Encapsulation
- Best-Effort Receivers

Additionally, there are several longer-term issues related to emerging technologies and scaling that will be important to future RSVP over ATM solutions. Lastly, the mapping of INT-SERV QoS classes and parameters is considered to be outside the scope of issues related to running RSVP, the protocol, over ATM and is therefore not addressed in this section or this note. It is expected that this last area will be covered in other drafts.

2.1 Issue: VC Usage

There is a basic need to map from IP and RSVP to ATM virtual circuits (VCs). In the permanent virtual circuit (PVC) environment, this is really a non-issue since PVCs are typically used as point-to-point link replacements. LAN Emulation [5], Classical IP [6] and, more recently, NHRP [7] discuss mapping IP traffic onto ATM SVCs, but they only cover a single QoS class, i.e., best effort traffic. When QoS is introduced, VC mapping must be revisited. For RSVP controlled QoS flows, there are two main issues: VCs to be used for RSVP (control) traffic, and VCs to use for QoS data flows.

RSVP control traffic is used to install state and request allocation of resources, it is not used to deliver any user data. As a control protocol, it is important for RSVP traffic to be delivered with some assurance of reliability. To that end, RSVP includes a soft-state, retry mechanism. This mechanism allows state to be maintained even when some RSVP messages are not delivered. When running RSVP over ATM it is critical for RSVP traffic to receive adequate service so that QoS state is installed and maintained.

The second VC usage issue is data flow to VCs mapping. In the Classic IP over ATM and current NHRP models a single VC is used for all traffic between two ATM attached hosts (routers and end-stations). It is likely that such a single VC will not be adequate or optimal when supporting data flows with multiple QoS types. RSVP's basic purpose is to install support for flows with multiple QoS types, so it is essential for any RSVP over ATM solution to address VC usage for QoS data flows. RSVP reservation styles will also need to be taken into account in any VC usage strategy.

There is also the lesser VC usage issue of reuse of VC reverse paths. ATM point-to-point VCs allow for bi-directional data flow. The reverse path could be used for best effort or QoS traffic. Bi-directional VCs could even be used to provide true receiver initiated reservations (reverse path of VC would be forward path for IP data.) Use of VC reverse paths will need to be addressed as part of a VC usage strategy.

2.2 Issue: Heterogeneity

There are several types of related heterogeneity. One type is that multiple receivers may request different Rspecs within a single session. This means that the amount of requested resources may differ on a per next hop basis, see figure 1. Optimally, only the resources requested will be allocated. This is not possible with current ATM UNI since both UNI 3.x and UNI 4.0 require the same QoS allocations on all leafs of a point-to-mulitpoint VC. It is possible that a future version of UNI will support multiple QoS parameters within a single VC, but this is not yet the case. A RSVP over ATM solution will need to support heterogeneous receivers within a single service class even though ATM does not currently provide such support directly.

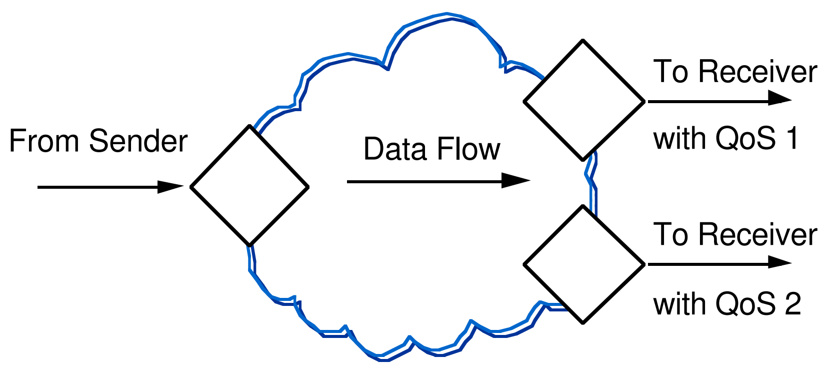

Figure 1: Heterogeneous RSVP Receivers

There is also the possibility that there will be RSVP requests for different service classes within a single RSVP session. This case is not permitted in RSVP version one, so this issue does not need to be addressed by RSVP over ATM solutions. Similarly, there is the possibility that there will be requests for multiple reservation styles within a single session. This case is also not allowed by RSVP version one. While RSVP based requests for multiple service classes are not allowed, there remains a requirement to be able to support a single RSVP requested QoS class along with non-QoS traffic. This means that any RSVP over ATM solution must always support the default, best-effort, service class along with a requested QoS service class.

2.3 Issue: Routing

This section will discuss two routing related issues. The first routing related issue is dealing with ATM short-cuts. As shown in figure 2, short-cuts [7] allow ATM attached routers and hosts to directly establish VCs across LIS boundaries, i.e., the VC end-points are on different IP sub-nets. This model of VC establishment poses several issues when running with RSVP. The key issues are dealing with established best-effort short-cuts, when to establish short-cuts, and QoS only short-cuts. These issues will need to be addressed by all RSVP implementations, furthermore it may be desirable to permit different interoperable strategies for dealing with short-cuts.

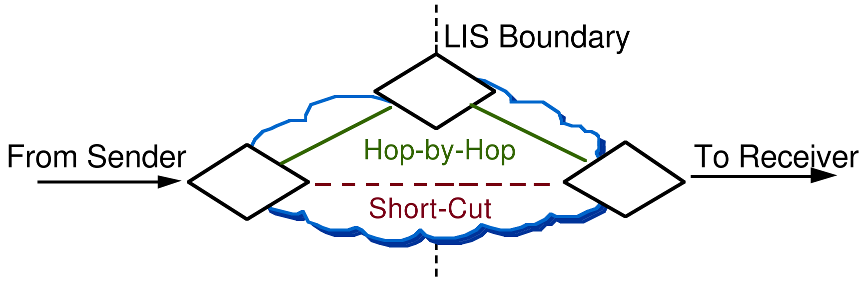

Figure 2: ATM Short-Cuts

The second routing related issue is dealing with multicast servers. Two models are planned for IP multicast data distribution over ATM, see [8]. In one model data, senders establish point-to-multipoint VCs to all ATM attached destinations, and data is then sent over these VCs. This model is often called "multicast mesh" mode distribution. In the second model, senders send data over point-to-point VCs to a central point and the central point relays the data onto point-to-mulitpoint VCs that have been established to all receivers of the IP multicast group. This model is often refereed to as "multicast server" mode distribution. Figure 3 shows data flow for both modes of IP multicast data distribution. Both modes of data distribution will need to be addressed for a complete RSVP over ATM solution.

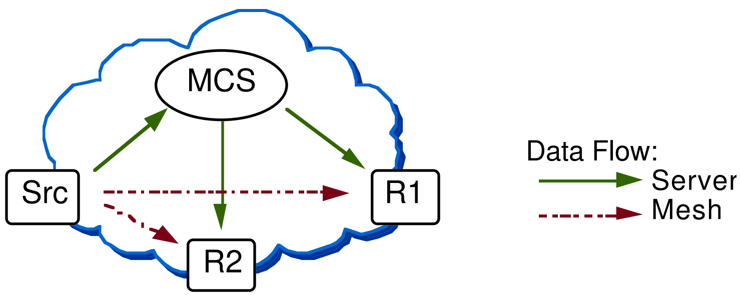

Figure 3: IP Multicast Data Distribution Over ATM

2.4 Issue: QoS Renegotiation

QoS renegotiation when using ATM UNI 3.x and UNI 4.0 is problematic. The basic issue is that RSVP allows for changes in a flow's QoS parameters and QoS is fixed at VC setup time for both ATM UNI 3.x and UNI 4.0. There are several options for dealing with this mismatch in service. A specific approach will need to be specified by any RSVP over ATM solution.

2.5 Issue: Encapsulation

One issue that has not been generally discussed is encapsulation. There are two encapsulation options for running IP over ATM, and it may even be desirable to support RSVP over both options. The first option is described in RFC 1483 [9] and is the option that has been generally assumed to be the one that will be used with RSVP.

The second option is LAN Emulation, as described in [5]. While RSVP over LANE has not been generally discussed, LANE is commonly used in the ATM LAN environment. It is likely that initial RSVP over ATM solutions will only support RFC 1483 encapsulation. While this is the case, it may be desirable to also define RSVP over ATM over LANE. LANE encapsulation has the added issue that LANE does not include a QoS signaling interface. If LANE encapsulation is needed, LANE QoS signaling would first need to be defined.

2.6 Issue: Best-Effort Receivers

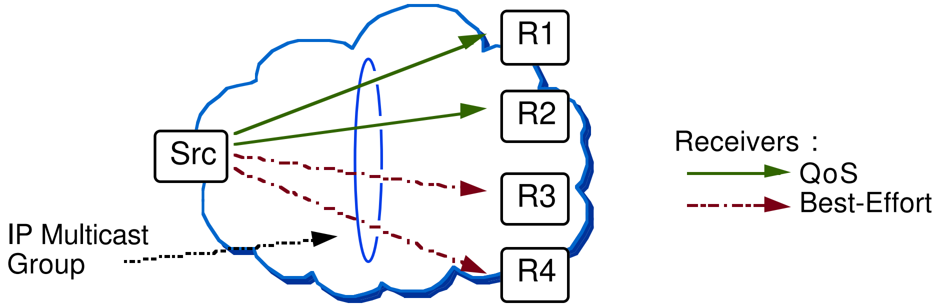

Figure 4: Types of Multicast Receivers

Another issue that has not been generally discussed is dealing with best-effort receivers. In any IP multicast group, it is possible that some receivers will request QoS (via RSVP) and some receivers will not, see figure 4. In shared media, like Ethernet, receivers that have not requested resources can be given a free ride without almost any effort or complications. This is not the case with ATM. In ATM networks, the end-points of each VC must be explicitly added. There may be costs associated with adding the best-effort receiver, and there might not be adequate resources to add the non-QoS receiver. While this issue is not strictly an RSVP issue, it is an issue that must be addressed by any RSVP over ATM solution.

2.7 Future Issues

There are several issues that will become increasingly important to solve as time passes. These issues are not likely to be critical for initial RSVP over ATM solutions. The first issue is dealing with QoS based routing protocols. Standard versions of QoS routing protocols do not yet exist, but there have been private versions as well as discussions related to standardization efforts. So, in the not too distant future it will be likely that RSVP over ATM solutions will need to operate in an enhanced routing environment. There may or may not be additional issues related to running in such an environment. Identification of specific issues will need to wait until such protocols are brought into the IETF.

Another issue will be running RSVP over ATM UNI 4.0. Initial RSVP over ATM solutions are likely to be aimed at UNI 3.x. This will be sufficient for today's ATM networks, but support of UNI 4.0 will also be needed. The key change that may impact an RSVP over ATM solution is the introduction of multipoint-to-multipoint VCs. UNI 4.0 will also allow different service class mappings, but this is more of an INT-SERV issue.

The last issue that will need to be solved is scaling. There are two dimensions of scaling. The first dimension is number of receivers in a single session. Scaling number of receivers may trigger RSVP message implosion as well as ATM signaling limitations (e.g., limits on rate of add parties). The second dimension of scaling is number of sessions. The key ATM related implication of large number of sessions is the number of VCs and associated (buffer and queue) memory. The numbers of flows and sessions supported by an RSVP over ATM solutions is likely to change over time, so a solution that is adequate for initial RSVP over ATM may not be adequate in the longer term. It will be increasingly important to address both of these dimensions of scaling as use of deployment and use of RSVP over ATM increases.

2.8 Non-Issue: Receiver Vs Sender Initiation

There is an apparent mismatch between RSVP and ATM. Specifically, RSVP control is receiver oriented and ATM control is sender oriented. (This is critical in ATM point-to-multipoint VCs, less so for bi-directional point-to-point VCs.) This initially may seem like a major issue, but slightly deeper analyses shows this to be a non-issue. While RSVP reservation (RESV) requests are generated at the receiver, actual allocation of resources takes place at the sub-net sender. This is true for all sub-net technologies. As is shown in figure 5, VCs are always initiated by the ATM "sender". Note that the ATM sender may be a router or an end-station.

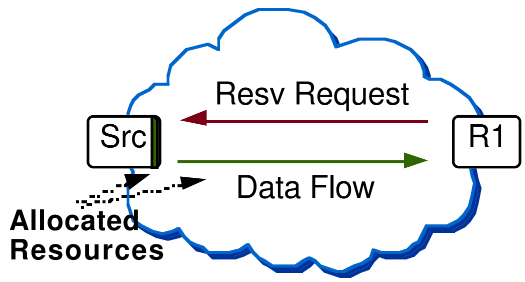

Figure 5: RSVP Resource Allocation

3. Solution Overview

This section describes a framework that allows for evolution of RSVP over ATM solutions as well as variability in implementation. The framework supports multiple specific solutions that may be used to integrate with evolving IETF and ATM forum standards. It also minimizes interoperability interdependencies and allows for interoperability between somewhat different approaches to running RSVP over ATM. This framework addresses VC usage and routing interactions

The rest of this section describes the framework and also covers possible issues that need to be addressed in the short, medium, and long terms. Section 4 provides a framework compliant method for running RSVP over ATM UNI3.x.

3.1 Framework

3.1.1 VC Usage

The most significant aspect of the solution framework is VC usage by RSVP and associated (QoS) data flows. Figure 6 represents the proposed model for data flow VC initiation. For data flows, sub-net senders establish all QoS VCs and that the sub-net receiver always accept incoming VCs. This means that receivers will never initiate reservations via the reverse path mechanism provided by point-to-point VCs. This model is consistent with RSVP version one processing rules and allows senders to use different flow to VC mappings and even QoS renegotiation techniques without necessitating any change in receivers. Additionally, receivers are restricted from using the reverse path provided by point-to-point VCs, since this is a special case that cannot be used to support multicast flows.

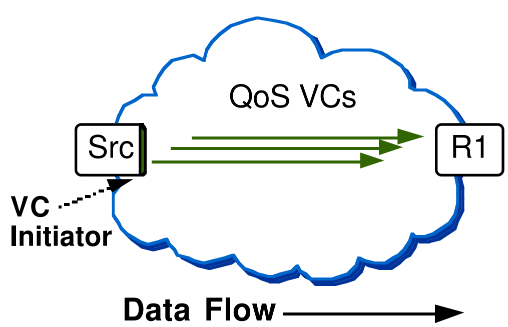

Figure 6: Data Flow VC Initiation

Figure 7 shows the proposed model for VC usage by RSVP control. RSVP control (messages) will always be sent over the best effort data path. This approach is the most efficient and satisfies RSVP reliability requirements. This approach minimizes VC requirements since the best effort data path will need to exist in order for RSVP sessions to be established and in order for RSVP reservations to be initiated. Best effort paths will also be used for data distribution while RSVP reservations are not in place. The reliability of the best effort data path will also be adequate since RSVP allows for a certain amount of packet loss without any loss of state synchronization.

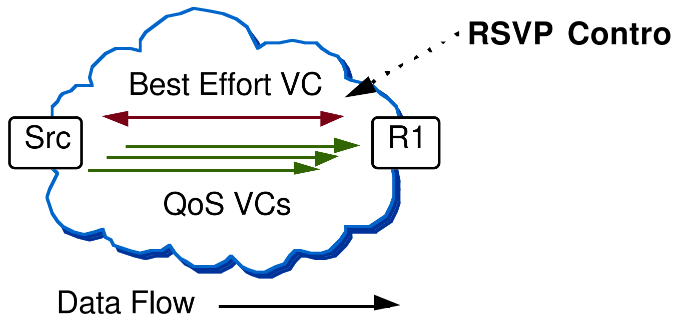

Figure 7: RSVP Control Message VC Usage

3.1.2 Routing

The second critical aspect of the solution framework is dealing with IP routing issues. The proposed solution is for both RSVP control and QoS data flows to just follow IP routing. This means that there is no route pinning (per [1].) It also means that short-cut paths are supported. Initial RSVP over ATM solutions may only follow existing short-cuts, but the framework supports QoS triggered short-cuts or even per flow short-cuts.

The handling of short cuts has been raised as an issue. The area of concern is that a down-stream short-cut may not have a matching up-stream short-cut, and that this would result in PATH and RESV messages following different paths. This asymmetric RSVP control scenario is shown in figure 8. More detailed examination shows this to be a non-issue. RSVP includes mechanisms to detect and handle RESV messages arriving at the wrong router and the wrong interface. For messages arriving at the wrong router, [1] states: "... IP destination address does not match any of the interfaces of the local interfaces, then forward the message to IP destination address ..." For RESV messages arriving at the wrong interface, [1] states: "The logical outgoing interface OI is taken from the LIH in the NHOP object .... [or] it can be learned from the interface matching the IP destination address." Since incoming interface is not used in RESV processing there is no issue. So, both cases are covered by RSVP processing rules and there is no RSVP issue with supporting ATM short-cuts.

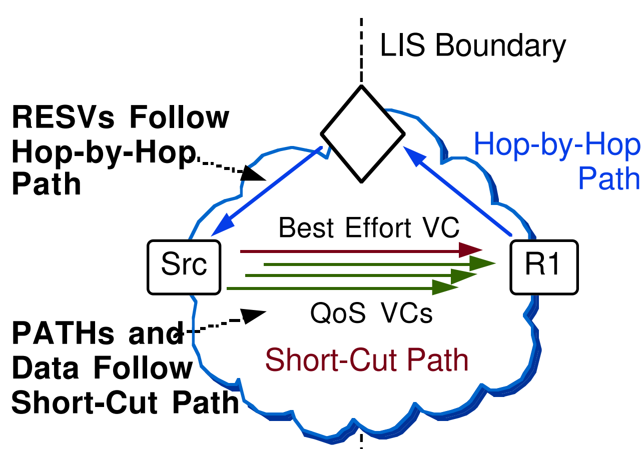

Figure 8: Asymmetric RSVP Message Forwarding With ATM Short-Cuts

3.2 Non-RSVP Protocol Issues

As IETF and ATM Forum protocols evolve it will be necessary to update specific approaches taken to support RSVP over ATM. The framework described above permits an interoperable evolution of RSVP over ATM specifications and implementations. This section presents a view of how some protocol aspects will change and how those changes may impact RSVP over ATM solutions. The relevant IP over ATM and ATM protocol aspects that are highlighted are encapsulation, multicast, QoS renegotiation, heterogeneous flows/VCs, and short-cuts. Protocol evolution is broken down into short-term (now), medium-term, and long-term. The short term is focused on current and soon to be issued standards and will be the basis for the RSVP over ATM solution specified in section 4. The medium term is focused on emerging standards such as UNI4.x. The long term is focused on new QoS related requirements for standards.

Issue	Short-Term	Medium-Term
Encapsulation	RFC 1483	
Multicast		
Signaling	UNI 3.x	UNI 4.0
Data Distribution	Mesh	
QoS Renegotiation		
Heterogeneous VCs		
Short-cut VCs	NHRP	QoS NHRP

Table 1: Possible Related Protocol Developments

Table 1 presents today's RSVP over ATM relevant protocols and some related future protocol developments. For the foreseeable future encapsulation will be based on RFC 1483. This is really the only option since LANE does not support any QoS signaling or control. For multicast, there are two relevant issues: signaling and distribution mode. Today's ATM networks support UNI 3.0 and 3.1 signaling, UNI 4.0 signaling will soon be complete and it's support will be required. For multicast data distribution, only point-to-multipoint VCs can be used to allocate resources for RSVP controlled QoS flows until the time when MARS is extended to provide some form of QoS control for multicast servers (MCS). Neither QoS renegotiation nor heterogeneous point-to-multipoint VCs are supported by either UNI 3.x or 4.0. Lastly, for short-cuts, until use of the NHRP QoS options is specified it is likely that all short-cuts will be done solely on a per destination basis.

3.2.1 Long-Term Issues

The long-term will be driven by future non-RSVP protocol developments. These developments are likely to be somewhat driven by internet QoS signaling (i.e., RSVP) over ATM requirements. This section discusses areas in today's IP over ATM and ATM Forum protocols that if extended would benefit RSVP over ATM. Issues related to NHRP, MARS, UNI, and LANE are covered. The discussion only covers the (RSVP) protocol related issues. INT-SERV related issues are outside the scope of this document.

3.2.1.1 NHRP

The Next-Hop Resolution Protocol, or NHRP, is being developed to support short-cut VC establishment. When using RSVP it may be desirable to establish multiple short-cut VCs, to use these VCs for specific QoS flows, and to use the hop-by-hop path for other QoS and non-QoS flows. The current NHRP specification [7] does not preclude such an approach, but nor does it explicitly support it. We believe that explicit support of flow based short-cuts would improve RSVP over ATM solutions. We also believe that such support may require the ability to include flow information in the NHRP request, but this is an area for further study.

3.2.1.2 MARS

In the MARS model, data distribution may be handled either by point-to-multipoint VCs or via a multicast server (MCS). When using RSVP, it is desirable to establish VCs with specific QoS. Establishing such VCs is readily done when using point-to-multipoint VCs, but is more complicated when using multicast servers. When using a multicast server, the sub-network sender could establish a point-to-point VC with a specific QoS to the server, but there is not current mechanism for the MCS to establish anything other than an UBR VC. In order to support RSVP over a MARS-MCS it will be necessary to provide QoS extensions to MARS.

3.2.1.3 ATM UNI

As discussed in [3], there are two issues related to current ATM UNI specifications. Neither UNI3.x nor UNI 4.0 support QoS renegotiation or heterogeneous point-to-multipoint VCs. Support of either of these features would benefit running RSVP over ATM. QoS renegotiation would be particularly beneficial since the only option available today for changing VC QoS parameters is replacing the VC.

3.2.1.4 LANE

Although running RSVP over LANE is not considered to be critical at this time, it may become desirable at some point in the future. If LANE was used to support RSVP controlled flows, it would be desirable for VCs to be establish with specific QoS for those flows. LANE currently does not include any ability to signal QoS requirements or to control QoS used for established VCs. So, if RSVP was to be run over LANE, LANE would need to be extend to provide a QoS signaling interface. Such extensions would also need to ensure proper QoS allocation for multicast traffic.

4. RSVP Over ATM UNI 3.x

This section describes a method for running RSVP over ATM UNI 3.x. The presented method is consistent with the framework described in the previous section. This method is targeted at today's ATM technology and networks. The key areas of the approach are encapsulation, VC management, and multicasting. The approach also requires changes to RSVP's interfaces and processing rules.

4.1 Encapsulation

Since RSVP is a signaling protocol used to control flows of IP data packets, encapsulation for both RSVP packets and associated IP data packets must be defined. While it is possible to use different encapsulations for each, this does not seem to make sense. So, the same encapsulation will be used for RSVP and the flows of IP data packets controlled by RSVP. As previously mentioned, RFC1483 and LANE are two defined encapsulation options for IP over ATM. Currently LANE doesn't have a QoS control interface. So, since QoS control is needed to make RSVP over ATM useful, RFC 1483 encapsulation must be used by RSVP over ATM. This is the same encapsulation used by Classical IP and NHRP.

4.2 VC Management

This section specifies a baseline solution for management of VCs associated with QoS data flows. The described solution will interoperate with other framework compliant solutions. The general approach taken in developing this method was to use mechanisms that matched today's standards and technology. As described in Section 3, VCs will always be initiated and controlled by the sub-net sender. When establishing and maintaining VCs, the sub-net sender will need to deal with several complicating factors including multiple QoS flows, requests for QoS changes, ATM short-cuts, and several multicast issues.

4.2.1 Flow to VC Mapping

There are multiple options for mapping flows onto VCs, see [2] for a more general discussion. The key issue to be addressed is providing requested QoS downstream. While it is possible to send multiple flows and multiple distinct reservations (FF) over single VCs, implementation of such approaches is still a matter for research. So, baseline RSVP over ATM implementations must allow for the use of a single VC to support each RSVP reservation. By using independent VCs per reservation, delivery of requested resources to the associated QoS data flow can be assured. This approach does not preclude, but does not specify, support for multiple flows per VC.

An RSVP reservation is characterized in [1] by the Traffic Control State Block, or TCSB. In the approach where each reservation maps to its own VC, the RSVP traffic control function "TC_AddFlowspec" will translate into an ATM call setup. The "TC_DelFlowspec" control function will translate into an ATM call release. The traffic control interface will need to be extended to include end-point identification.

4.2.2 QoS Renegotiation

UNI 3.x does not support any modifications to QoS parameters after VC setup. During normal RSVP processing it is possible for the requested resources to be increased or decreased. The RSVP traffic control function "TC_ModFlowspec" is used to request a change in allocated resources. Since UNI 3.x does not support such changes directly, RSVP nodes must compensate.

The proposed approach for supporting changes in RSVP reservations is to attempt to replace an existing VC with a new appropriately sized VC. During setup of the replacement VC, the old VC is left in place and unmodified. The old VC is left unmodified to minimize interruption of QoS data delivery. Once the replacement VC is established, data transmission is shifted to the new VC, and the old VC is then closed.

If setup of the replacement VC fails, then the old QoS VC should continue to be used. When the new reservation is larger than the old reservation, the reservation request should be answered with an error. When the new reservation is smaller than the old reservation, the request should treated as if the modification was successful. While leaving the larger allocation in place is suboptimal, it maximizes delivery of service to the user. Implementations should retry replacing the too large VC after some appropriate elapsed time.

One additional issue is that only one QoS change can be processed at one time per reservation. If the (RSVP) requested QoS is changed while the first replacement VC is still being setup, then the replacement VC is closed and the whole VC replacement process is restarted.

4.2.3 Short-Cuts

As previously discussed ATM short-cuts do not pose any problem for RSVP. The only issue that needs to be addressed by the baseline RSVP over ATM solution is when to establish a short-cut for a QoS data flow. The proposed approach is to simply follow best-effort traffic. When a short-cut has been established for best-effort traffic to a destination or next-hop, that same end-point should be used when setting up RSVP triggered VCs for QoS traffic to the same destination or next-hop. This will happen naturally when PATH messages are forwarded over the best-effort short-cut.

4.3 Multicast

There are several aspects to running RSVP over ATM that are particular to multicast sessions. These issues result from the nature of ATM point-to-multipoint connections. The baseline RSVP over ATM solution addresses next-hops requesting different QoS values, and handling of best-effort receivers. The last issue is not strictly RSVP issues, but needs to be addressed in a complete RSVP over ATM solution.

4.3.1 Heterogeneous Flows

As discussed in Section 2.2, multiple next-hops (or receivers) may request different resources within a single session. The current RSVP specification does address this case, but not within an ATM specific context. The current processing rules and traffic control interface describe a model where the largest requested reservation is used in resource allocation, and traffic is delivered at the higher rate to all next-hops. The simplest approach for RSVP over ATM will be to emulate this approach even though this approach may be undesirable in certain circumstances. So, baseline RSVP over ATM implementations must be able to support heterogeneity in QoS requests by providing the largest requested QoS to all next hops. Baseline implementations, may also support heterogeneity through some other mechanism, e.g., using multiple appropriately sized VCs.

There is an interesting interaction between heterogeneous flows and QoS renegotiation that is worth mentioning. In the case where a RESV message is received from a new next-hop and the requested resources are larger than any existing reservation, both QoS renegotiation and heterogeneity will need to be addressed. The question is whether to first add the new next-hop or to change to the new QoS. This is a fairly straight forward special case. Since the older, smaller reservation does not support the new next-hop, the QoS renegotiation process should be initiated. Since the new QoS is only needed by the new next-hop, it should be the first end-point of the new VC.

4.3.2 Best-Effort Receivers

In any IP multicast group, some end-points will issues RSVP reservation requests and some will not. Those who don't, are best-effort receivers and there is no requirement to provide them improved service delivery. In ATM, providing them better than best-effort service may actually be the opposite of what the user desires since in providing ATM QoS, there may be charges incurred or resources that are wrongfully allocated.

Two possible approaches to this problem are using a single QoS VC or using multiple VCs. In the single VC approach even the best-effort receivers use the RSVP triggered QoS VC. This case optimizes host and network resources. While this approach is simple to implement, it may incur extra charges or may cause the best-effort receiver to not receiver some traffic because the QoS VC setup failed. The multiple VC approach ensures proper QoS delivery of data. It uses a QoS VC and a best-effort VC. The QoS VC is used solely for RSVP controlled end-points. The best-effort VC is used for all other end-points. The best-effort VC does not include the RSVP controlled end-points in order to avoid data duplication. The sender must duplicate packets destined to the IP multicast group, placing one copy on each VC. This approach uses more VCs, and sender processing and link resources, but it assures proper handling of the traffic on the ATM network.

Unfortunately, neither of these approaches is the right answer for all cases. For some networks, e.g. LANs, it is likely that the single VC approach will be desired. In other networks, e.g. public WANs, it is likely that the multiple approach will be desired. The framework discussed in section 3 permits each sub-network sender (router, or host) to choose how traffic is mapped onto VCs. For this reason, baseline RSVP over UNI3.x implementations must support best-effort multicast receivers either using the single QoS VC or the multiple VC approach. Implementations should support both approaches and provide the ability to select which method is actually used, but are not required to do so.

4.4 RSVP Changes

To support ATM there are several changes required to the RSVP specification, [1]. These changes do not alter any protocol formats or behavior. The same RSVP messages are sent and processed over ATM as with any other sub-network technology. The changes required to run over ATM are only internal and are required to ensure proper ATM VC management. The traffic control and routing interfaces need to be extended to include end-point identification, and message processing rules need to be changed to trigger VC initiation.

4.4.1 Traffic Control Interface

The RSVP specification contains a description of a generic traffic control interface. The specified interface is adequate when running over point-to-point (e.g., PPP) or broadcast (e.g., Ethernet) sub-networks, but not when running over ATM. Specifically, the interface does not include the ability to identify next-hops when establishing or modifying reservations. Two calls need to be extended to include next-hop identification. Since described extensions can be used with point-to-point and broadcast network technologies, a single traffic control interface can be used for all network types.

- Make a Reservation

```
Call: TC_AddFlowspec( Interface, TC_Flowspec,
                 TC_Tspec, Police_Flags, Next_Hop_List )
                -> RHandle [, Fwd_Flowspec]
```

The new parameter, Next_Hop_List, contains one or more IP addresses each of which are be associated with the requested reservation. The traffic control module should establish a new reservation with the specified flowspec to all listed next-hop addresses. Typically all next-hops will be on the same sub-net as the sender. The one exception to this case is when (NHRP) short-cuts are being used.

When Interface is an ATM interface, each TC_AddFlowspec call will trigger setup of a new VC. The end-points of the VC will be identified by Next_Hop_List. When there are more than one next-hops listed, next-hops not include in initial VC setup must be added via the “add party” request. The QoS service class and traffic parameters will be set according to (the to be defined) INT-SERV to ATM mappings. All return (reverse path) traffic parameters will be set to zero (0), even for point-to-point calls.

- Modify Reservation

```
Call: TC_ModFlowspec( Interface, RHandle, TC_Flowspec,
                      Sender_Tspec, Police_flags,
                      Next_Hop_List )
                          [ -> Fwd_Flowspec ]
```

Next_Hop_List contains the full list of next-hop addresses. Omission of a previously listed addresses indicates that those next-hops should be dropped from the reservation. Addition addresses should be added to the reservation. It is valid for both TC_FlowSpec and Next_Hop_List to indicate reservation modifications.

When Interface is an ATM interface, the TC_ModFlowspec call will need to check for two types of change, change in next-hops and change in QoS. To check for change in next-hops, the list of existing VC end-points should be compared with Next_Hop_List. Any end-points not listed in Next_Hop_List should be dropped from the VC via the “drop party” request. Any next-hops that are not already end-points of the VC should be add via the “add party” request.

When a QoS change is requested, a new VC should be established to all listed entries in Next_Hop_List. Once all next-hops have been added to the new VC, data forwarding should be set to use the new VC, and the old VC should be released (closed). If all next-hops cannot be added, then the new VC should be closed, and failure should be returned.

It is valid for both QoS and Next_Hop_List to be changed in a single call. In this case, any end-points of the existing VC not listed in Next_Hop_List should be dropped from the old VC before starting setup of the new VC. This ensures proper next-hop handling even when setup of the new VC fails.

4.4.2 Routing Interface

As with Traffic Control, the routing interface needs to be extended to include next-hop identification. Again the described extensions can be used with point-to-point and broadcast network technologies.

- Route Query

```
Ucast_Route_Query( [ SrcAddress, ] DestAddress,
                     Notify_flag )
                 -> OutInterface, Next_Hop


Mcast_Route_Query( [ SrcAddress, ] DestAddress,
                     Notify_flag )
```

```
                                  -> [ IncInterface, ] OutInterface_list,
                                      Next_Hop_List
```

The Next_Hop return value contains the IP address of the next-hop to be used on OutInterface. Next_Hop_List contains a per OutInterface list of next-hop IP addresses. In the case of ATM, it is expected that unicast information will be obtained from standard routing or NHRP [7], and that multicast information will be obtained either via MARS [8] or directly from a multicast routing protocol, e.g., MOSPF.

- Route Change Notification

```
Ucast_Route_Change( ) -> [ SrcAddress, ] DestAddress,
                               OutInterface,
                               Next_Hop


Mcast_Route_Change( ) -> [ SrcAddress, ] DestAddress,
              [ IncInterface, ] OutInterface_list,
              Next_Hop_List
```

The new parameters are used to indicated changes in next-hop information.

4.4.3 Processing Rules

This section lists changes to the processing rules contained in [1]. The changes are limited to traffic control processing, and are fairly minor. While the processing rules are being changed to support RSVP over ATM, the same rules can be used to run RSVP over any network technology.

4.4.3.1 TCSB - Traffic Control State Block

There is only one parameter that needs to be added to the TCSB and this mirrors the changes made to the traffic control interface. The addition to the TCSB is:

- Next_Hop_List, the list of next hops associated with the current reservation.

4.4.3.2 Update Traffic Control Processing

The traffic control processing needs to be extended to support changes in next-hops. This is a fairly minor change. The following is the complete traffic control processing, most of which is directly copied from [1]. All changes/additions are noted with a horizontal bar (|) in the right column.

UPDATE TRAFFIC CONTROL

The sequence is invoked by many of the message arrival sequences to set or adjust the local traffic control state in accordance with the current reservation and path state. An implicit parameter of this sequence is the `active' RSB.

If the result is to modify the traffic control state, this sequence notifies any matching local applications with a RESV_EVENT upcall. If the state change is such that it should trigger immediate Resv refresh messages, it also turns on the Resv_Refresh_Needed flag.

Initially the Next_Hop_List is empty. |

- Compute the traffic control parameters using the following steps.
  1. Consider the set of RSB's matching SESSION, Filter_spec_list, and OI from the active RSB. Initially the local flag Is_Biggest is off.
     - Compute the effective kernel flowspec, TC_Flowspec, as the LUB of the FLOWSPEC values in these RSB's.

- Compute the effective traffic control filter spec (list) TC_Filter_Spec* as the union of the Filter_spec_lists from these RSB's.
- If the active RSB has a FLOWSPEC larger than all the others, turn on the Is_Biggest flag.
- Add each RSB's next hop IP address to Next_Hop_List.

2. Scan all RSB's matching session and Filtss, for all OI. Set TC_B_Police_flag on if TC_Flowspec is smaller than, or incomparable to, any FLOWSPEC in those RSB's.
3. Locate the set of PSBs (senders) whose SENDER_TEMPLATEs match Filter_spec_list in the active RSB and whose OutInterface_list includes OI.
4. Set TC_E_Police_flag on if any of these PSBs have their E_Police flag on. Set TC_M_Police_flag on if it is a shared style and there is more than one PSB in the set.
5. Compute Path_Te as the sum of the SENDER_TSPEC objects in this set of PSBs.

- Search for a TCSB matching SESSION and OI; for distinct style (FF), it must also match Filter_spec_list.
  If none is found, create a new TCSB.
- If TCSB is new:
  1. Store TC_Flowspec, TC_Filter_Spec*, Path_Te, Next_Hop_List, and the police flags into TCSB.
  2. Turn the Resv_Refresh_Needed flag on and make the traffic control call:

     TC_AddFlowspec(OI, TC_Flowspec,
     Path_Te, police_flags, Next_Hop_List)
     -> Rhandle, Fwd_Flowspec
  3. If this call fails, build and send a ResvErr message specifying "Admission control failed" and with the InPlace flag off. Delete the TCSB, delete any RESV_CONFIRM object from the active RSB, and return.
  4. Otherwise (call succeeds), record Rhandle and Fwd_Flowspec in the TCSB. For each filter_spec F in TC_Filter_Spec*, call:

     TC_AddFilter(OI, Rhandle, Session, F)
     -> Fhandle

     and record the returned Fhandle in the TCSB.

- Otherwise, if TCSB is not new but no effective kernel flowspec TC_Flowspec was computed earlier, then:
  1. Turn on the Resv_Refresh_Needed flag.
  2. Call traffic control to delete the reservation:

     TC_DelFlowspec(OI, Rhandle)
  3. Delete the TCSB and return.

- Otherwise, if TCSB is not new but the TC_Flowspec, Path_Te, police flags, and/or Next_Hop_List just computed differ from corresponding values in the TCSB, then:

1. If the TC_Flowspec and/or Path_Te values differ, turn the Resv_Refresh_Needed flag on.
2. Call traffic control to modify the reservation:

   TC_ModFlowspec(OI, Rhandle, TC_Flowspec,
   Path_Te, police_flags, Next_Hop_List)
   -> Fwd_Flowspec

3. If this call fails, build and send a ResvErr message specifying "Admission control failed" and with the InPlace bit on. Delete any RESV_CONFIRM object from the active RSB and return.
4. Otherwise (the call succeeds), update the TCSB with the new values and save Fwd_Flowspec in the TCSB.

- If the TCSB is not new but the TC_Filter_Spec* just computed differs from the FILTER_SPEC* in the TCSB, then:
  1. Make an appropriate set of TC_DelFilter and TC_AddFilter calls to transform the Filter_spec_list in the TCSB into the new TC_Filter_Spec*.

- If the active RSB contains a RESV_CONFIRM object, then:
  1. If the Is_Biggest flag is on, move the RESV_CONFIRM object into the TCSB and turn on the Resv_Refresh_Needed flag. (This will later cause the Resv REFRESH sequence to be invoked, which will either forward or return the RESV_CONFIRM object, deleting it from the TCSB in either case).
  2. Otherwise, create and send a ResvConf message to the address in the RESV_CONFIRM object. Include the RESV_CONFIRM object in the ResvConf message. The ResvConf message should also include an ERROR_SPEC object whose Error_Node parameter is IP address of OI from the TCSB and that specifies "No Error".

- If the Resv_Refresh_Needed flag is on and the RSB is not from the API, make a RESV_EVENT upcall to any matching application:

  Call: <Upcall_Proc>(session-id, RESV_EVENT,
  style, Flowspec, Filter_spec_list
  [, POLICY_DATA])

  where Flowspec and Filter_spec_list come from the TCSB and the style comes from the active RSB.

- Return to the event sequence that invoked this one.

5. Security Considerations

Security issues are not addressed in this memo.

6. References

[1] Braden, R., Zhang, L., Estrin, D., Herzog, S., and S. Jamin, “Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification”, Internet Draft.

[2] Berson, S., “Classical RSVP and IP over ATM”, Proceeedings INET96.

[3] Borden, M., Crawley , E., Krawczyk, J, Baker, F., and Berson, S., “Issues for RSVP and Integrated Services over ATM”, Internet Draft.

[4] Onvural, R., Srinivasan, V., “A Framework for Supporting RSVP Flows Over ATM Networks”, Internet Draft.

[5] The ATM Forum, “LAN Emulation Over ATM Specification”, Version 1.0

[6] Laubach, M., "Classical IP and ARP over ATM", RFC 1577, December, 1993.

[7] Katz, D., Piscitello, D., Cole, B., Luciani, J., “NBMA Next Hop Resolution Protocol (NHRP)”, Internet Draft.

[8] G. Armitage, “Support for Multicast over UNI 3.0/3.1 based ATM Networks”, Internet Draft.

[9] Heinanen, J., “Multiprotocol Encapsulation over ATM Adaptation Layer 5”, RFC 1483, July 1993.

7. Author’s Address

Lou Berger
FORE Systems
6905 Rockledge Drive
Suite 800
Bethesda, MD 20817

Phone: 301-571-2534
EMail: lberger@fore.com