Providing IP Quality of Service over ATM

S. Thomson and M. Garrett
Bellcore

December, 1996

1 Introduction

In this document, we describe guidelines for implementing IP realtime services over ATM networks using the emerging IP and ATM standards. The guidelines address two subjects: how to implement the IP resource reservation protocol (RSVP) in terms of the ATM user-network interface (UNI), and how to implement IP Guaranteed and Controlled Load service classes using the ATM Constant Bit-Rate (CBR), Variable Bit-Rate (VBR), Unspecified Bit-Rate (UBR) and Available Bit-Rate (ABR) service categories.

The guidelines assume the classical model of IP in which ATM subnetworks are configured as logical IP subnets[9]. In particular, we do not address the implications of using protocols[10] that allow an ATM connection to be established directly between nodes on different subnets.

The mappings are based on the following IETF and ATM Forum specifications: Guaranteed Service specification[14], Controlled Load Service specification[15]. RSVPv1[16], UNI 3.1[6], UNI 4.0[8] and Traffic Management (TM) 4.0[7]. This year, proprietary signalling standards have been proposed which would remove the need for the UNI signalling in certain topologies[5, 4]. We do not address the use of these protocols in this document. However, to the extent that the service classes implemented in ATM hardware are those described in UNI 3.x and TM 4.0, and to the extent that the nature of virtual circuits remains unchanged, the guidelines recommended in this document and the lessons learned are applicable to the new signalling protocols as well.

Our work to date in addressing the problem of mapping IP services onto ATM has led us to the following key conclusions:

- Guaranteed Service can be implemented in UNI 3.x and UNI 4.0. However, the mapping is not straightforward in some cases and does require special configuration. The level of implementation complexity will depend on switch implementations.
- Controlled Load Service can be supported in UNI 3.x and 4.0, but it is not possible to properly capture the delay and loss behavior of this service in ATM terms. The result is that it is only possible to request a very conservative implementation of Controlled Load Service. The mapping for Controlled Load Service may also not be straightforward depending on switch implementations for the reason given below.
- One of the main problems in supporting IP services is that not all ATM service classes or combinations of traffic parameter values support best-effort service for non-conforming traffic.

Even if the standards do support this behavior, some switches may not implement the desired behavior, because it is not mandatory.

- RSVP can be implemented using UNI 3.x and UNI 4.0. Some implementation complexity is introduced to support receivers that change reservation requests during a session. Multicast sessions are significantly more complicated than unicast sessions, because different receivers can make different reservation requests and some members of a group may make no requests at all. The UNI 4.0 leaf-initiated join feature is not necessary or immediately helpful in supporting RSVP.
- The RSVP specification as written favors point-to-point and broadcast link technologies, both in its recommendations and message process rules. Modifications need to be made to support non-broadcast technologies like ATM.

In the rest of the document, we explain these conclusions and describe in detail how the mappings are made. The document is structured as follows. Section 2 gives an overview of the problem, and defines terms and abbreviations. Section 3 specifies the valid service mappings for the two IP real-time services, while Section 4 describes VC management policies that implement RSVP reservations. The relationship between the VC management policies and the service mappings are discussed in Section 5. Section 6 concludes the document.

2 Problem Description

In IP, RSVP[16] is used to signal the appropriate quality of service. Senders send PATH messages to set up the route for data packets for a session (which may be unicast or multicast), and receivers send RESV messages towards the senders to reserve resources along the (uni-directional) path.

PATH messages carry the sender's description of the data in terms of token bucket parameters (TSpec), as well as information about the services available on the path and resource characteristics of the path (AdSpec). RESV messages carry a TSpec characterizing the data (the receiver may use the sender's TSpec, or may modify it) as well as the requested service and service parameters (RSpec).

In ATM, a sender uses the appropriate UNI signalling messages to set up a virtual circuit (VC) with a specified quality of service. A VC SETUP request establishes a route with the appropriate QoS resources to one or more receivers. Point-to-point VCs may be bi-directional; point-to-multipoint are uni-directional. The SETUP message includes a traffic descriptor containing information similar to the TSpec, and the name of the requested service along with the service parameters in a manner similar to the RSpec.

To illustrate the context for mapping RSVP onto ATM signalling, we consider a simple example of four IP nodes connected to a logical IP subnet that happens to be an ATM network (Figure 1). Assume node A is a sender of data and that nodes B, C and D are receivers of that data. (The network nodes may be routers or hosts. If routers, the sender is a router forwarding packets from a source on an upstream network to the receivers on the ATM subnet. The receivers on the ATM subnet are routers that are forwarding packets to the ultimate destination.)

Before sending data on a QoS session, the sender (node A) sends a PATH message to all session receivers (nodes B, C and D). On receiving a PATH message, a receiver may send a RESV message

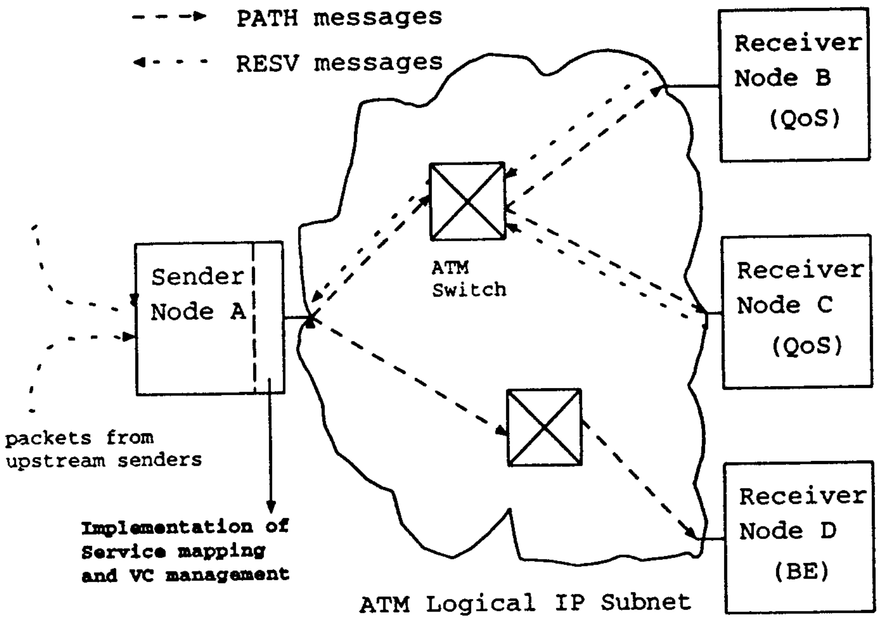

Figure 1: QoS Sender and QoS/BE Receivers in an ATM Logical IP Subnet

to reserve resources along that path. (If a receiver does not send a RESV message, like Node D, the receiver is implicitly asking for best-effort service.) A RESV message is sent hop-by-hop in the reverse direction of the data path. In the above example, a RESV message from a receiver will first be sent to node A. If node A is a router, the message will be merged with any existing reservations for the same session, and forwarded to the source in a hop by hop manner until the merged message reaches the source or until it reaches a place along the path where the resources have already been reserved for that session. This could happen if another receiver has already reserved the resources and shares part of the path. RSVP has several styles allowing receivers to apply a reservation to all or some subset of the senders to the session.

On receiving a RESV message, node A must reserve the necessary resources towards the receiver that sent the RESV message. This is where the IP resource reservation request must be translated into a VC setup request at the ATM layer. Two problems that need to be solved in an IP/ATM topology are:

- **Service mapping policies:** A VC must be used that has a QoS that supports the IP service being requested. There are several ways of doing the mapping for each IP service, and the choice will depend on the switch implementations and on network management policy. We discuss these in Section 3.
- **VC management policies:** There are various ways of mapping resource reservation requests to VCs; the choice of policy needs to be configurable as it will depend on the network environment. It is possible to use a QoS VC per individual reservation request, per session or some number of sessions. We discuss these in Section 4.

Other problems that need to be addressed in an IP/ATM QoS implementation include per interface admission control, scheduling, policing and reshaping, and flow classification. These problems are not peculiar to ATM, though, and need to be addressed when using other link layer technologies as well. We refer to these issues in this document, but do not discuss them here.

2.1 Terminology

A **session** is defined by a destination address, destination port and transport-layer protocol. The address may be unicast or multicast.

A **flow** is a stream of packets that have the same source and destination address and port pairs, and carry the same transport protocol identifier.

(It follows from the above definitions that the number of flows per session depends on the number of senders.)

A **reservation** is a request from a receiver for an allocation of resources (or set of allocations) that will provide a particular set of flows in a given session with the specified QoS (RESV message). Depending on the style, a single reservation may be used for a number of different flows (shared explicit and wildcard filter styles) or a reservation may be made per flow (fixed filter style).

We note that the RSVP specification recommends that individual reservation requests that come in over the same interface be merged into a single reservation for the session, so that there is per session allocation of resources only at the link-layer. This suits broadcast and, trivially, point-to-point links. In point-to-point links, there is only ever one receiver and hence one reservation request for the session. In the case of broadcast links, per session allocation is sufficient because the nature of the medium means that all receivers will receive the same data in any case. In general, this recommendation does not suit ATM since it might not be efficient to provide all receivers on a logical IP subnet with the same QoS. In the section on VC management policies (Section 4), we discuss this issue and policies that support merged as well as individual reservations on a single interface.

2.2 Abbreviations

In this document, we often use the descriptive abbreviation "QoS" to refer to IP receivers that have made a reservation request for non-default quality of service, i.e. sent a RESV message requesting Guaranteed Service or Controlled Load service. Hence, we distinguish **QoS receivers** (receivers that use RSVP to make resource reservations) from **BE receivers** (receivers that do not use RSVP), and we distinguish **QoS VCs** (VCs set up in response to RSVP reservation requests) from **BE VCs** (VCs set up to carry best-effort traffic).

ABR	Available Bit Rate
ATM	Asynchronous Transfer Mode
BCOB	Broadband Connection-Oriented Bearer Capability
BCOB-A,C,X	Bearer Class A, C, or X
BE	Best Effort
BE VC	A VC used for best-effort traffic
BE receiver	A receiver that has made no reservation request
BT	Burst Tolerance
CBR	Constant Bit Rate
CDV	Cell Delay Variation
CDVT	Cell Delay Variation Tolerance
CLP	Cell Loss Priority (bit)
CLR	Cell Loss Ratio
CLS	Controlled Load Service
CTD	Cell Transfer Delay
GS	Guaranteed Service
IP	Internet Protocol
IWF	Interworking Function
MBS	Maximum Burst Size
MCR	Minimum Cell Rate
MPL	Minimum Path Latency
PCR	Peak Cell Rate
QoS VC	A VC implementing an RSVP resource reservation
QoS receiver	A receiver that has made a reservation request
SCR	Sustained Cell Rate
UBR	Unspecified Bit Rate
VBR	Variable Bit Rate
nrtVBR	Non-real-time VBR
rtVBR	Real-time VBR

3 Service Mappings

In this section, we discuss how to request the two IP services, Guaranteed Service and Controlled Load Service in terms of the ATM service categories. The mappings will be used by the VC management policy to set up an appropriate VC, or to determine whether an existing VC is suitable for a resource reservation request.

The mappings must of course ensure that the semantics of the IP services are properly translated for both conforming and non-conforming traffic. Mapping the IP service classes onto the ATM service classes requires mapping the following sets of parameters:

- the IP traffic characterization (TSpec) must be mapped onto the ATM traffic descriptor
- the IP service class (RSpec) parameters must be mapped onto the ATM service category (QoS) parameters

- the IP advertised parameters (AdSpec) must be filled in with appropriate ATM layer values

Since both IP and ATM use the same model for characterizing traffic, the mapping of traffic descriptors is straightforward. However, the mapping of the other parameters and service semantics may not be.

3.1 Problems

There are three problems with mapping IP services onto ATM service categories (bearer classes):

- ATM service categories do not always support best-effort semantics for non-conforming traffic as is required for the IP service classes
- There is no way to indicate the approximate nature of the delay and loss characteristics of Controlled Load Service in ATM terms
- There is no analog in ATM of the IP AdSpec

3.1.1 Dealing with Non-conforming Traffic

Both IP real-time services require that non-conforming traffic, i.e. traffic that exceeds the TSpec of the reservation, must be sent best-effort[1]. That is, non-conforming packets should get exactly the same treatment as best-effort packets in the absence of congestion.

In ATM, there are two possibilities for providing best-effort service to non-conforming packets:

- Set up a QoS VC with a high peak cell rate (relative to the line rate) that carries tagged cells with best-effort service
- Send these packets down a separate VC established to carry best-effort traffic

The former option is significantly better than the latter option since packet ordering is maintained, and there is no need to use a separate VC which may or may not be readily available[2].

However, tagging with best-effort service is not supported by all combinations of traffic-related parameters in ATM. In particular, the PCR value represents an upper bound on the rate at which data can enter the network — cells sent above this rate, whether tagged or not, are policed and dropped. Thus, there is no support for sending traffic above the peak rate rate with best-effort service. It is possible to avoid this problem by setting PCR to the line rate (or some other upper bound that a system administrator has configured that a flow may never exceed e.g. 10% of the line rate). In this case, it does not matter that non-conforming cells are dropped since the network capacity has been exceeded.

[1] Guaranteed Service does not strictly speaking require this, but it is a strong recommendation given the applications being considered in this document.

[2] It is unclear how badly packet misordering affects receivers in practice, but it is generally agreed that misordering should be avoided if possible for interactive realtime applications. Other applications using TCP, for example, may not be as susceptible to misordering.

The above problem may be exacerbated by switch implementations. The ATM standards do not specify that tagged cells must be carried best-effort. A compliant implementation may drop tagged cells irrespective of whether there is congestion. Furthermore, some switches may not allow VCs to be established with peak rates that are high relative to the bandwidth of the link. So it may not be possible to set PCR to the link rate or some significant fraction thereof. Such implementations are unlikely to support IP well. Note that one cannot get around this problem by buffering non-conforming traffic since, while this may decrease loss rate, it increases delays unnecessarily providing a service that is worse than ordinary best-effort service.

We note that tagged cells in a QoS VC should be given the same service as cells in a UBR VC. In particular, tagged cells in a QoS VC should not be given priority over cells in a UBR VC. It is unclear whether scheduling algorithms in ATM switches have this property.

3.1.2 Supporting Controlled Load Service

In Controlled Load Service, the delay and loss requirements are not stated in precise quantitive terms. From the specification:

> The end-to-end behavior provided to an application by a series of network elements providing controlled-load service tightly approximates the behavior visible to applications receiving best-effort service *under unloaded conditions* from the same series of network elements. Assuming the network is functioning correctly, these applications may assume that:
>
> - A very high percentage of transmitted packets will be successfully delivered by the network to the receiving end-nodes. (The percentage of packets not successfully delivered must closely approximate the basic packet error rate of the transmission medium).
>
> - The transit delay experienced by a very high percentage of the delivered packets will not greatly exceed the minimum transmit delay experienced by any successfully delivered packet. (This minimum transit delay includes speed-of-light delay plus the fixed processing time in routers and other communications devices along the path.)

ATM does not have a service that defines loss and delay characteristics in this way. In those ATM service categories where these parameters are specified, the semantics are that these values are upper bounds, and conformant implementations must not exceed these bounds. What Controlled Load Service really needs is the ability to specify delay and loss rates that are the target for a conforming implementation, but which may very occasionally be exceeded.

It is possible to request Controlled Load Service by setting delay and loss parameters to be the minimum transit delay and the packet error rate of the transmission medium respectively. However, this would enforce implementations to be extremely conservative, and not take advantage of the looseness in the definition to achieve statistical multiplexing gain. Note that, while it is possible to set the loss rate and delay to be less stringent than the above, this is unlikely to provide the desired behavior (at least not in the current generation of switch implementations).

A consequence of the above is that it is not possible for switches to distinguish between a request for Guaranteed Service and a request for Controlled Load Service, if both are implemented using the same service category. This may not matter in the short-term because current implementations of Controlled Load Service are likely to be conservative anyway due to the state of the art. However, in the longer term this issue needs to be addressed.

3.1.3 Completing the AdSpec

The IP services require routers to advertise information, such as the services available on a specific interface, and estimates of the link delay and bandwidth. This information is carried in RSVP PATH messages and used by receivers to make reservation decisions.

Some of the information in the AdSpec is service-independent and some is service-dependent. Currently, the service-independent information includes the path hop count, the path bandwidth estimate, the minimum path latency, the path maximum transmission unit, and whether RSVP is supported on each hop of the path. The hop count is incremented by one by each RSVP-capable node. It is expected that the other values in a RSVP-capable node will be configured per interface since there is no practical way of determining them otherwise. The minimum path latency is critical to the implementation of the Guaranteed Service and we will discuss setting this value in more detail in the section on implementing this service.

We discuss the Guaranteed Service AdSpec parameters in Section 3.2. There are no significant service-dependent parameters for Controlled Load service.

3.1.4 Other Issues

When mapping the IP parameters onto ATM parameters, the units need to be converted from bytes to cells. The conversion needs to take into account cell segmentation overhead and the minimum policed unit.

In the constraints that follow, we assume that the necessary conversion has taken place so that the parameters can be meaningfully related.

3.2 Guaranteed Service over ATM

Guaranteed Service is a service that guarantees a maximum bound on the end-to-end delay of a packet and zero loss due to congestion.

Guaranteed Service specifies five parameters in its TSpec: peak rate (p), bucket rate (r), bucket size (b), minimum policed unit (m) and maximum packet size (M). The service is parameterised by two parameters in the RSpec: the requested rate (R) and the slack term (S)[3]. The AdSpec parameters for Guaranteed Service include the cumulative C and D terms advertised by each hop along the path. These terms are implementation-incurred delays that receivers must take into account when determining the end-to-end delay provided by Guaranteed Service. C is a rate-dependent term,

[3] We ignore this term for now.

while D is a rate-independent term.

The delay is determined using parameters in the TSpec, RSpec and AdSpec, according to the following formula[4]:

$$\frac{b}{R} + \frac{C}{R} + D + \mathrm{MPL}$$

R must be at least the token bucket rate r. The reason for the parameter R is to enable a receiver to decrease the delay by requesting a higher value of R.

Since ATM provides no easy way to determine the values of C and D for a network, the guideline is to model ATM as a fixed delay component only, i.e. set C to its minimal value (one) and set D to a value that includes the maximum queueing delay plus any other rate-independent delay. (Note that the propagation delay is carried in the minimum path latency parameter that is part of the general parameter set.)

Below, we discuss how to map Guaranteed Service into ATM classes using UNI 3.x and UNI 4.0 respectively.

3.2.1 UNI 3.x

In the case of UNI 3.x, there is only one possible bearer class to implement Guaranteed Service, and that is CBR. The other bearer classes do not provide any real-time guarantees.

The use of CBR requires the specification of a peak cell rate only. There are two problems with this:

- Non-conforming cannot be supported on the same VC
- The choice of value for PCR may not lead to effective network utilization

PCR must satisfy the following constraints: $R \leq \mathrm{PCR} \leq p^*$, where p^* is the minimum of the line rate and the specified peak rate p. PCR must be set to at least R to satisfy the end-to-end delay constraints promised by Guaranteed Service. The advantage of setting PCR to R is that it consumes fewer network resources. The disadvantage is that the edge device must be capable of buffering bursts (up to size b) above rate R. Unless the value chosen for PCR is the line rate, using the CBR service category does not support the best-effort semantics of non-conforming traffic, and this must be implemented by the edge device as described in Section 3.1.1.

In UNI 3.x, one specifies the service parameters indirectly, using the QoS class information element, which is essentially an index into a table of values that are configured in the network. Since there is a relationship between the D term advertised by the edge device in the AdSpec and the cell transfer delay (CTD) and Cell Delay Variation (CDV) value of a VC, these values must be configured together. The relationship is as follows:

$$\begin{aligned} \mathrm{CTD} &= \mathrm{MPL} + \mathrm{D} + \mathrm{S} \\ \mathrm{CDV} &= \mathrm{CTD} - \mathrm{MPL} \end{aligned}$$

[4] This is a simplified version of the formula, but suffices for our purposes.

The cell loss rate must be that of the packet error rate of the transmission medium. For Guaranteed Service, the QoS class value should be set to one. Note that it is unclear whether any switches make use of this parameter in practice.

3.2.2 UNI/TM 4.0

In UNI/TM 4.0, there is another service class, rtVBR, that may be better suited to supporting Guaranteed Service than CBR depending on the implementation.

When using rtVBR, the PCR and SCR parameters must satisfy the following contraints:

$$R \leq \text{PCR} \leq L$$
$$r \leq \text{SCR} \leq \text{PCR}$$

where L is the line rate, or some configured fraction therof.

The MBS parameter has a value that will depend on the choice of value for PCR and SCR. A good choice of default parameter is the token bucket size b.

Unlike the CBR service category, it may be possible to set PCR greater than R, without lowering network utilization significantly, depending on the implementation. Ideally, it should be possible to set PCR to the higher rate L to take advantage of the tagging mechanism. That is, if the switches support high PCR values efficiently, then such values (like the line rate) may be used independent of the value of p specified in the traffic descriptor. If this feature is not available, then the best-effort semantics of non-conforming traffic must be supported in the edge device.

UNI 4.0 allows service class parameters, which include CLR, CTD, and CDV to be specified explicitly. The CTD, CDV and CLR values for both the CBR and rtVBR services must be set with the same constraint as stated for UNI 3.x above.

3.3 Controlled Load Service over ATM

As discussed in Section 3.1.2, Controlled Load Service is intended to provide a service that has delay and loss characteristics that are similar to best-effort service under unloaded conditions.

Controlled Load Service specifies five parameters in its TSpec: peak rate (p), bucket rate (r), bucket size (b), minimum policed unit (m) and maximum packet size (M). The RSpec and AdSpec are empty.

3.3.1 UNI 3.x

Controlled Load can be implemented using the VBR service class. It is, of course, possible to use the CBR class, but this is likely to be highly inefficient in the absence of a VC management that does flow aggregation (see Section 5).

When using VBR, the PCR, SCR and MBS parameters must satisfy the same contraints defined in the rtVBR mapping for Guaranteed Service above.

In the Controlled Load Service, non-conforming traffic must be given best-effort semantics. If the ATM switches allow PCR to be set to the line rate, and provide best-effort service to tagged cells, then Controlled Load Service is completely supported by the VBR service category. Otherwise, the edge device must compensate.

The QoS class value that is used to index network-specific service parameters values should be set to three. Once again, it is unclear whether this parameter is used by switches for any practical purpose.

3.3.2 UNI/TM 4.0

In UNI/TM4.0, Controlled Load service can be implemented using the nrtVBR class, the rtVBR class, the CBR class and the ABR class.

The use of nrtVBR is similar to using VBR in UNI 3.x. The main difference is that nrtVBR allows a loss rate to be specified. Since Controlled Load does not specify a loss rate, this value is set by the edge device and will typically be manually configured per interface. Since, as described above, the loss rate semantics of Controlled Load cannot be captured well using CLR, it should be set to the packet error rate of the transmission medium. This provides a conformant implementation of Controlled Load, but does not allow an efficient implementation.

Implementing Controlled Load service using rtVBR is possible, but is more than is necessary because rtVBR guarantees a delay bound, which may cause the allocation of more resources than necessary. (This is an analogous problem to setting the loss rate parameter.) If rtVBR is used, PCR, SCR and MBS should be set to that specified for Guaranteed Service (see Section 3.2.2). The value of CTD, CDV and CLR should also be set as for Guaranteed Service.

CBR may be used to implement the Controlled Load Service, but it is likely to lead to low network utilization as resources will be reserved for a flow unnecessarily. The rate of the CBR VC would need to be at least r, and the edge device would need to have a buffering capacity of at most b. Also, the edge device would need to support non-conforming traffic appropriately. The QoS parameters should be set as for rtVBR above.

ABR supports Controlled Load service well in the sense that it guarantees a minimum cell rate and provides best-effort service above that rate. This corresponds to the "best-effort with floor" semantics of Controlled Load. However, switches running ABR use a feedback mechanism to notify senders to slow down under congestion — this is unlikely to be of any use to receivers of Controlled Load service since delayed packets are of no more value than lost packets.

Using ABR, the MCR traffic descriptor parameter must be at least the value of the Controlled Load TSpec parameter r. The Controlled Load parameter b is at most the amount of buffering required at the edge device. The use of tagging and QoS parameters in ABR are for further study.

4 VC Management Policies

A VC management policy determines when to set up and close VCs, and how to map flows to VCs.

VCs need to be set up to carry RSVP control packets as well as data packets. Section 4.1 discusses VC management policies for RSVP control packets, for both unicast and multicast sessions. In Sections 4.2 and 4.3, we discuss VC management policies for data traffic. The first section discusses unicast traffic and the second multicast data traffic.

The policies in the first three sections assume one VC per session for any particular sender, or possibly one VC per session reservation request, if heterogeneous receivers need to be supported. This is suitable for environments where VCs are plentiful relative to the number of sessions, and

VC setup and teardown does not incur a significant cost. However, this is not always the case. In Section 4.4, we explore QoS implications of aggregating sessions or flows into a single VC.

4.1 VC management for Unicast and Multicast RSVP Messages

RSVP control packets need best-effort service, and will likely be sent on the same VCs as other best-effort traffic. We assume in this document that VC management policies for setting up the best-effort path follow the guidelines in RFC 1755, its successor[12] and RFC 2022, unless otherwise noted. It should be ensured that sufficient capacity exists to carry RSVP messages as well as normal best-effort data messages along a best-effort path.

In RSVP, every sender to a session periodically sends PATH messages to establish the route that the data will follow, so that subsequent reservation requests, RESV messages, can follow the reverse of this path and allocate resources along the data path.

PATH messages are sent to the session address, and so may be unicast or multicast. PATH messages should follow that of the best-effort data path, and should not be mixed in with the QoS data path. A typical VC management policy for best-effort data[11, 12, 1] is to set up a VC per different next-hop address, i.e. one per unicast address and one per multicast group address. This works sufficiently well for unicast next-hops, since data forwarded to the same address can be easily aggregated into a single VC. This may not be good enough for multicast next-hops, since this implies a VC per group, even if groups all have the same members. However, aggregation of multiple groups is non-trivial because group membership is dynamic.

RESV messages are always unicast to the previous hop on the data path. RESV messages should follow that of the best-effort unicast data path to the previous hop. A VC management policy may set up a point-to-point VC with bandwidth allocated in one direction only, or the policy may be to always set up bi-directional point-to-point VC with the expectation that unicast best-effort traffic is typically two way.

In the case of a multicast session, the best-effort path for a PATH message and the best-effort path for a RESV message are likely to use different VCs: a point-to-multipoint VC in the forward direction for the PATH message and a point-to-point VC in the backward direction for the RESV message. However, in the case of a unicast session, it is possible to use the same VC for both PATH and RESV messages (and possibly other best-effort data traffic as well). The choice is dependent on VC management policy for best-effort traffic. Typically, a best-effort VC management policy does allocate bandwidth in both directions for point-to-point VCs.

The rest of this section deals with data traffic as opposed to RSVP signalling traffic. In the following two subsections on data traffic, we discuss how to set up a QoS VC on receiving a reservation request for a session, how to modify the QoS of the session dynamically, and also how to deal with different reservation requests in the case of a multicast session.

4.2 VC management for Unicast Data Traffic

VC management for unicast data sessions is straightforward. When a RESV request arrives from a receiver, a point-to-point VC is set up to the receiver with the appropriate QoS (as determined by the service mappings described in Section 3). The VC must stay up for as long as the RSVP reservation is current. That is, a VC management policy must not time out VCs based on the absence of data traffic as it would for a best-effort VC management policy[11]. Rather, the VC should be torn down when the RSVP reservation times out. This is easy to arrange for the sender, since the sender sets up the VC and knows that the VC is set up based on a RSVP reservation. This is not as simple for the receiver of the connection, since the receiver is unaware whether the VC is set up by RSVP, and hence what the VC management policy should be. Note that it cannot be assumed that a QoS VC is necessarily set up by RSVP, even if the protocol being used is IP: different future IP signalling protocols may have different rules for timing out connections. Possible solutions to this problem are:

- Signal to the receiver that only the sender will delete the connection. It is possible to use an existing information element in the UNI specification to do this, but since this is not a mandatory part of the specification, the element may be not be carried end-to-end by the switches.
- Send an empty data packet on the VC periodically when idle to prevent the receiver timing out the VC. This is not an ideal solution, since device drivers, and possibly applications, will be bothered by spurious packets.
- Assume that all QoS VCs are timed out by the sender only (or, alternatively, set the timeout value for incoming QoS VCs on the receiver side to some "infinite" value). This is the easiest solution to implement, and may suffice for the foreseeable future.

RSVP allows a receiver to change the requested level of service dynamically. Unfortunately, ATM signalling does not allow the QoS of an existing VC to be changed. The only solution is (i) to create a new VC with the new QoS, and (ii) once that has been successfully set up, to divert the data traffic from the old VC to the new VC, and then (iii) to close the old VC. If this solution is not feasible for some reason, it is possible to implement a policy that does not support dynamically changing reservation requests without violating the RSVP specification. For example, when a receiver decreases the level of service requested, the policy can be to keep the existing VC in place. When a receiver increases the level of service requested, the policy may be to deny such a request. The feasibility of such policies will depend on user requirements, and also on whether there is a pressing need for dynamic QoS changes. It is still a matter of debate whether Internet applications will require support for dynamic QoS. We discuss this issue further below.

4.3 VC management for Multicast Data Traffic

QoS VC management for multicast sessions is significantly more complicated than a unicast QoS VC management policy for the following reason. Some subset of receivers may make resource reservations and a (disjoint) subset may not make reservations. Those that make reservations must get the QoS requested, and those that do not must get (at least) best-effort service. Furthermore, receivers making reservations may ask for different qualities of service. Since ATM point-to-multipoint VCs do not support leaves with different QoS, it is necessary, in general, to set up more than one VC to carry data traffic.

To make the problem as simple as possible to start with, let us assume as a starting point that all receivers of a multicast session that make a reservation ask for the same quality of service. That is, receivers may ask for a single level of some real-time service, or they may make no reservation at all, in which case they get best-effort service. We then address the problem of heterogeneous reservation requests.

We believe that the above simplifying assumption is very reasonable. To date, there is no support in the Internet for receivers making different reservations for a single session. In the case where one receiver makes a large reservation, and the other makes a small reservation, there is no mechanism defined which indicates how routers should size an incoming data stream into the smaller reservation. Research is underway to solve this problem. The most promising approach in our view is to use layered coding techniques and stripe the different layers across different multicast groups[13]. Receivers will join more or less groups depending on the level of service they are willing or able to support. The implication of this approach is that a single level of service is associated with a group or session, and hence reservation requests will be the same for a single session.

4.3.1 Homogeneous Reservation Requests

The simplest per session VC management policy is to set up a single point-to-multipoint QoS VC to all receivers in the session. All receivers whether these receivers have made reservations or not, will be placed on a QoS VC as soon as one of the receivers makes a reservation. (Prior to this, all receivers will be receiving data over the best-effort data path - this is likely to be the best-effort point-to-multipoint VC that is also carrying RSVP PATH messages for the session. See Section 4.1 above.) This policy is acceptable only as long as all receivers can be added to the QoS VC (this will not be the case if there are insufficient QoS resources), and that receivers do not get worse service from the QoS VC than they would under a best-effort path (this should not happen, but ...)

It is imperative that any VC management policy does not support QoS receivers at the expense of best-effort receivers. Since the above policy does not guarantee service to best-effort receivers under all conditions, it cannot be the only VC management policy implemented. The simple policy above can only be used in conjunction with a policy which explicitly supports best-effort receivers as well as QoS receivers. Such a policy would implement two VCs for session data traffic: one QoS VC and one BE VC. Both VCs would be set up when the first reservation request came in: the QoS VC would be set up to the receiver that has made the reservation request, and the BE VC would be set up to the remainder of receivers in the session. When the next receiver makes a reservation request, the receiver will be added to the QoS VC and removed from the BE VC, and so on.

Note that the QoS and BE VCs used in the above policy are separate from the BE VC used to carry RSVP PATH messages. It is not possible to put PATH and BE data messages on the same VC when a QoS receiver exists because the PATH messages must go to *all* receivers in the session, while the BE data must go to only those receivers that have not made a reservation request. Sending data down the same VC as the PATH messages would mean than QoS receivers get duplicate data packets: one on the PATH VC and one on the QoS VC. It is extremely undesirable for receivers to get duplicate packets. If the receiver is a host, the application is likely not to adapt well to sustained packet duplication. If a receiver is a router, duplicates will likely be forwarded down a QoS path degrading the QoS of the flow on the next link of the path.

The above policy (the **limited heterogeneity** policy), which may be used in conjunction with the simple policy above (the **homogeneous** policy), is the minimal requirement for implementing VC management for RSVP over ATM. Note that the limited heterogeneity policy should not break the RSVP specification if heterogeneous reservations are made or if a receiver changes a reservation request dynamically. It is consistent with the RSVP specification that a new receiver that asks for a reservation that is different from other reservations can be denied service. Any existing receiver that increases a previous reservation request can also be denied service. An exsiting receiver that decreases a reservation request must always be given a successful response (whether the resource allocation is downgraded or not).

4.3.2 Heterogeneous Reservation Requests

If we relax our assumption above, and design policies that explicitly support heterogeneous reservation requests for a single session, then the limited heterogeneity policy can still be used with the proviso that the quality of the QoS VC is the result of merging all the individual reservation requests. Since receivers make QoS requests at different times, and may change these requests during a session, the QoS of the VC will very likely need to change. As discussed in the section on unicast sessions above, a new QoS VC will need to be established to support the session and the old one closed.

An alternate policy is to implement a multicast group using several point-to-point VCs, one per group member, or possibly several point-to-multipoint VCs, one per level of service. The choice would depend on whether requests are "clumped", ie whether a subset of receivers (where the subset is greater than one) ask for the same QoS or whether each receiver is likely to ask for a completely different QoS. Point-to-point VCs have the advantage that aggregation of different sessions is easier. Point-to-multipoint VCs are more efficient to use from a sender's point of view because the sender does not need to duplicate data to all leaves.

A middle road between the above two policies would be to support only a predefined set of QoS levels per session, and map reservation requests into the minimum level of service that satisfies the request. Modifying the QoS of an existing request means moving the receiver to a VC with a higher or lower QoS.

4.4 Session Aggregation

In the above VC management policies, we have assumed that whenever resources need to be allocated to a QoS session, a new VC or set of VCs is established to serve that session. There was no sharing of VCs by flows in more than one session. Such VC management policies do not scale well when sessions are short-lived, and there are many of them.

It is possible to map flows from several sessions into an existing VC provided that the VC is "large" enough to carry the traffic and the QoS of the VC meets the QoS requirements of each flow. This requires more sophisticated admission control schemes in the edge device since a decision must be made whether an existing VC can satisfy a new session request (rather than relying on the ATM network to make this decision when a new VC is set up). The admission control algorithms are likely to be service-dependent. We discuss the relationship between VC management policy, service mappings and admission control in Section 5.

We discuss VC aggregation in the context of VC management policies that use only point-to-point VCs, and then those that use point-to-multipoint VCs.

4.4.1 Point-to-Point VCs

Suppose one of the above VC management policies determines that a point-to-point VC of some size and QoS is needed. Before setting up the required VC, the policy determines whether an existing VC exists which has sufficient idle resources and makes the appropriate QoS guarantees for the new reservation request.

To do aggregation, a "large" QoS VC suitable for supporting several reservation requests must be set up. This may be done at start-up time, if destinations are known a priori, or may be done when the first reservation request for a new session comes in. Sizing the VC may be done dynamically. For example, a predefined size could be defined for initial allocation. Once the VC gets filled up, one could allocate another VC with twice the traffic capacity, and so on, and close down the original, smaller VCs as they become idle.

Note that there may need to be several "large" QoS VCs to a destination, one per QoS. This requirement is service-dependent and is discussed in Section 5.

4.4.2 Point-to-Multipoint VCs

Now, suppose one of the above VC management policies determines that a point-to-multipoint VC of some size and QoS is needed for multicast traffic. In principle, aggregating multiple QoS sessions onto a single point-to-multipoint QoS VC is the same problem as that for a point-to-point QoS VC. The difficulty is that multicast group membership changes dynamically so a point-to-multipoint VC that supports some set of groups at one time may not be able to support that same set at another time. When group membership changes, it may mean changing the endpoints of the VC or allocating the group to a different VC. Note that this problem is independent of IP QoS per se. The same problem arises in a purely best-effort model.

5 Relationship between VC Management Policy and Service Mappings

There are places where the VC management policy and service mappings are related. The choice of VC management policy as well as the type of service requested will determine the values of service parameters to be mapped as well as the choice of mapping.

5.1 Service Parameter Values

In VC management policies that do not do aggregation, a VC is set up whenever a new resource allocation needs to be made. Thus, a service mapping must be done from a resource reservation described in a RSVP RESV request (possibly merged with existing reservations). In VC management policies that do perform aggregation, a VC must be established of some "large" size and some level of QoS so that reservations belonging to many sessions can be satisfied by the VC. In these cases, the TSpec and RSpec to be mapped will have values determined by the VC management policy and the admission control policy, rather than coming from an individual reservation request. The mappings from the policy-determined values will be no different from the service mappings described in Section 3.

As far as sizing the VC goes, a VC needs to be set up with a token bucket descriptor that will hold some reasonable number of individual flows. So to set up a large VC, one sets up a VC with a large peak rate, token bucket rate and token bucket size and some expected minimum policed unit and maximum packet size. When mapping each new resource allocation request to the VC, one must determine if the VC has sufficient unused resources to carry the flow. The simple approach is to sum the token bucket descriptors of each flow in the VC, and determine whether there are sufficient unused resources to carry the token bucket descriptor of the new flow. Token bucket descriptors are summed as defined in each of the IP service definitions. This policy is likely to be very inefficient for variable bit-rate flows. However, such a policy is likely to be necessary for implementing the stringent requirements for Guaranteed Service.

For Controlled Load Service, the above admission control policy should take into account the gains to be had from statistical multiplexing. Admission control algorithms that do this are still a research issue, however. One possible starting point for the meantime is to take some fraction of the traffic descriptors into account when determining VC utilization. For example, one might assume that the statistical multiplexing gain is 1.5:1, and thus take 67% of the sum of the traffic descriptors into account when determining utilization. Such a policy would not work in general, but may have its place in controlled, private environments.

Besides ensuring that the TSpec of a new flow can be mapped into an existing VC, it must also be ensured that the RSpec requested is satisfied by the VC. For example, in Guaranteed Service, receivers can specify different values of R which implies different delay guarantees. Thus, new flows can only be mapped into an existing VC if the VC can accommodate traffic according to the TSpec *and* the VC can provide the necessary delay guarantees implied by the RSpec. However, as we have modelled ATM as a fixed-delay component in Guaranteed Service, one delay class may be all that is necessary.

Since the Controlled Load service does not have an RSpec, it is not necessary to be concerned with meeting different RSpec requirements.

5.2 Choice of service mapping

In the section on service mappings, we presented several choices for each of Guaranteed service and Controlled Load service. The choice of VC management policy is likely to influence the choice of service mapping. VC policies that aggregate flows may find using the CBR service attractive, rather than either of the VBR services. In the extreme, this translates into using the ATM network as a dumb pipe or leased line. If QoS software exists in routers to deal with point-to-point links, then this may be an expedient approach. This is especially true since admission control and scheduling algorithms are often designed with a fixed-rate link in mind, and its unclear what advantages there would be in taking into account the burstiness allowed in VBR VCs. There would be an advantage though, if the VBR services were significantly cheaper than CBR services (as they appear to be today).

ABR is defined only for point-to-point VCs, so this service cannot be used in conjunction with any VC management policy that uses point-to-multipoint VCs.

6 Conclusions

In this report, we have described guidelines for mapping IP QoS classes onto ATM service categories, and RSVP reservations onto ATM VCs, using the currents standards in the IETF and the ATM Forum. This work falls under the charter of the Integrated Services over Specific Link Layers

Working Group (ISSLL WG) in the IETF. This working group was established in mid-1996, and the guidelines for IP QoS over ATM are due to be completed in the second quarter of 1997[3, 2].

It is possible to implement Guaranteed Service and Controlled Load Service over ATM. However, the mappings are not always straightforward, and may require significant configuration. One important area of mismatch is that ATM does not always meet IP requirements for providing best-effort service to non-conforming traffic. Also, it is not currently possible to map the imprecise nature of delay and loss characteristics of Controlled Load Service, which makes it difficult to request a cheaper implementation of this service.

It is also possible to implement RSVP over ATM using either UNI 3.x and UNI 4.0 (without the leaf-initiated join feature). However, without session aggregation, the number of VCs per sender that is needed in multicast sessions, is at least two including the best-effort path for PATH messages, excluding RESV messages, possibly three if best-effort receivers are supported and even more if full heterogeneity is supported (up to one VC per reservation request from receiver). Such VC usage will not scale to a large number of senders and sessions on a single logical IP subnet.

There is much practical experience that needs to be gained with the new IP services and the ATM services. Further research on admission control, and scheduling and policing algorithms needs to be done to implement these services efficiently. Also, the signalling protocols in both IP and ATM may well be too heavyweight for certain network environments. Support for flow aggregation at the IP layer would not only make RSVP (or any other IP signalling protocol) scalable, but would make VC management simpler and more efficient. The recent proposals for IP switching and tag switching may well form the basis for lightweight signalling. Much works needs to be done, though, in fleshing out these protocols, and addressing aggregation, multicast and QoS requirements. It is likely that the lessons learned in mapping IP and ATM using existing standards can be used in integrating QoS into new signalling protocols.

References

[1] G. Armitage. Support for multicast over UNI 3.0/3.1 based ATM networks. Internet RFC 2022, IETF, November 1996.

[2] S. Berson and L. Berger. IP Integrated Services with RSVP over ATM. Internet Draft, draft-ietf-issll-atm-support-02.txt, IETF, November 1996.

[3] M. Borden and M. Garrett. Interoperation of Controlled-Load and Guaranteed-Service with ATM. Internet Draft, draft-ietf-issll-atm-mapping-01.txt, IETF, November 1996.

[4] P. Dooland, B. Davie, D.Katz, Y. Rekhter, and E. Rosen. Tag distribution protocol. Internet Draft, draft-doolan-tdp-spec-00.txt, IETF, September 1996.

[5] P. Newman et al. Ipsilon Internet Flow Management Protocol for IPv4 v1.0. Internet RFC 1953, IETF, May 1996.

[6] The ATM Forum. *ATM User-Network Interface Specification, Version 3.1.* Prentice Hall, Upper Saddle River NJ, 1995. ISBN: 0 13 393828 X.

[7] The ATM Forum. *ATM Traffic Management Specification, Version 4.0.* Prentice Hall, Upper Saddle River NJ, expected 1996.

[8] The ATM Forum. *ATM User-Network Interface (UNI) Signalling Specification, Version 4.0.* Prentice Hall, Upper Saddle River NJ, expected 1996.

[9] M. Laubach. Classical IP and ARP over ATM. Internet RFC 1577, IETF, January 1994.

[10] J. Luciani, D. Katz, D. Piscitello, and B. Cole. NBMA Next Hop Resolution Protocol. Internet Draft, draft-ietf-rolc-nhrp-10.txt, IETF, September 1996.

[11] M. Perez Maher and A. Mankin. ATM signalling support for IP over ATM. Internet RFC 1755, IETF, February 1995.

[12] M. Perez Maher and A. Mankin. ATM signalling support for IP over ATM — UNI 4.0 update. Internet Draft, draft-ietf-ion-sig-uni4.0-01.txt, IETF, November 1996.

[13] S. McCanne, V. Jacobson, and M. Vetterli. Receiver-driven Layered Multicast. *Proceedings of SIGCOMM'96 Conference*, pages 117–130, August 1996.

[14] S. Shenker, C. Partridge, and R. Guerin. Specification of Guaranteed Quality of Service. Internet Draft, draft-ietf-intserv-guaranteed-svc-06.txt, IETF, August 1996.

[15] J. Wroclawski. Specification of the Controlled-Load Network Element Service. Internet Draft, draft-ietf-intserv-ctrl-load-svc-03.txt, IETF, August 1996.

[16] L. Zhang, R. Braden, D. Estrin, S. Herzog, and S. Jamin. Resource Reservation Protocol (RSVP) – Version I Functional Specification. Internet Draft, draft-ietf-rsvp-spec-14.txt, IETF, October 1996.