VC Management for Heterogeneous QoS Multicast Transmissions

Jens Schmitt[1], Lars Wolf[1], Martin Karsten[1], and Ralf Steinmetz[1,2] *

1
Industr. Process and System Comm.
Dept. of Electr. Eng. & Inf. Technology
Darmstadt University of Technology
Merckstr. 25, D-64283 Darmstadt,
Germany

2
GMD IPSI
German National Research Center
for Information Technology
Dolivostr. 15, D-64293 Darmstadt,
Germany

Email:{Jens.Schmitt,Lars.Wolf,Martin.Karsten,Ralf.Steinmetz}@KOM.tu-darmstadt.de

Abstract

The ATM as well as the Internet community developed Quality of Service (QoS) architectures to provide integrated services. It can be assumed that both will play a certain role in the future and will coexist for quite some time. Presumably, ATM will be in the backbone, while IP will probably keep its dominance on the desktop. Hence, the use of an overlay model for the interaction between the two QoS architectures is very likely. A crucial component of such a model is the efficient mapping of RSVP (Resource reSerVation Protocol) as the Internet's signalling protocol onto the according ATM mechanisms. Unfortunately, due to the very different paradigms of the signalling protocols, this mapping is rather difficult.

While other components of the QoS architectures like the QoS models, the management frameworks, the charging mechanisms, etc. also need to be dealt with for a complete solution to the problem of overlaying the Internet's QoS architecture onto that of ATM, we concentrate on the support for RSVP by an ATM subnetwork. In particular, this paper focuses on one of the most contrary characteristics of RSVP and ATM. This is the support for heterogeneous reservations by RSVP over the ATM subnetwork, taking into account that ATM only allows for a homogeneous QoS within a single Virtual Circuit (VC). We present previous approaches to the solution of this problem and argue for more sophisticated and efficient approaches to manage ATM VCs taking into consideration ATM tariffs and resource consumption. Furthermore, we discuss how RSVP should be extended to provide a framework to enable these more flexible VC management strategies for supporting heterogeneity over an ATM subnetwork.

Keywords: IP/ATM Networks, RSVP, Multicast, Heterogeneous Reservations, VC Management, Resource Management, Cost Management.

*. This work is sponsored in part by Volkswagen-Stiftung, Hannover, Germany and by Deutsche Telekom AG, Darmstadt, Germany.

1 Introduction

The integration of the rising Internet QoS architecture with the QoS architecture of ATM is an important issue, not only to accelerate the growing usage of ATM as a backbone technology, but also to enable a future integrated services Internet, which is in need of a flexible and high-bandwidth backbone technology with an orderly traffic management.

RSVP/IntServ, which has been proposed by the IETF (mainly in [7],[18],[19]) as the Internet's QoS architecture, is at the moment under heavy discussion mainly due to scalability concerns, i.e., whether it is possible to support a sufficiently large number of concurrent flows. However, we believe that eventually in order to provide integrated services a scheme like RSVP/IntServ is necessary. We do not believe that an architecture like Differentiated Services [4] as it is discussed in the IETF at the moment will be a long-term solution for all QoS aspects, but rather a quick approach to satisfy short-term business needs. Furthermore, new research suggests that it will be technically possible to support many flows in routers in near future [14]. Therefore we assume RSVP/IntServ as the QoS architecture of the Internet and claim that many of the problems when overlaying it to ATM networks will arise for any fine-grained QoS architecture.

Since nowadays the Internet is a multi-provider network even if only its backbone is regarded, it is crucial for a mapping to take economic factors into account. This is of particular interest if the mapping process takes place at the edge between two providers or between a customer and its provider.

One of the most important points of the integration of the two QoS architectures is the mapping of the Internet's signalling protocol RSVP onto corresponding ATM mechanisms. Most problems in this area arise for the multicasting of data. The anticipated new services of a future Internet will be among others multimedia services like video-and audio-conferences, video-on-demand, interactive games, etc. All of them have in common that multicasting is necessary and thus we cannot circumvent the difficulties arising from that case.

One particular difference that only exists for multicast transmissions is RSVP's support of heterogeneous reservations, while ATM only allows for a homogeneous QoS within a single VC. The focus of this paper is on how this difference can be bridged to allow for efficient support of RSVP over ATM. The approaches suggested so far in the literature are either quite limiting or lead potentially to large resource consumption. We describe VC management techniques which support heterogeneous subnet-receivers by merging them into groups. Any such merging method should base its decisions on quantitative criteria. We study two cases, (1) cost-oriented and (2) resource-oriented techniques;

their application depends on the administrative location of the edge devices used for the mapping of RSVP/IntServ onto ATM.

In the next section, we briefly describe the differences between RSVP/IntServ and ATM and discuss whether heterogeneous QoS is possible and useful. In Section 3, VC management strategies are discussed – we review related work, and present our own schemes. As argued in Section 4, the currently defined RSVP traffic control interface is not capable to support NBMA (Non-Broadcast Multiple Access) networks and VC management strategies in particular. In Section 5 we conclude our investigations.

2 Issues in Mapping RSVP/IntServ onto ATM Networks

Before going into the details of heterogeneity support over ATM networks we want to reconsider which are the most important issues in mapping the Internet QoS architecture, RSVP/IntServ, onto ATM. There are two main problem areas: QoS models and QoS procedures. Therefore, the usual approach is to treat them separately, although there are some decisions which need an integrated view.

2.1 QoS Models

QoS models are the declarative component of QoS architectures, consisting of service classes and their traffic specifications and performance parameters. The most salient differences between the QoS models, i.e. the ATM TM 4.0 [1] and the IntServ specifications ([18], [19]), are:

- packet-based vs. cell-based traffic parameters and performance specifications,
- the handling of excess traffic (policing): degradation to best-effort vs. tagging or dropping,
- and, of course, different service classes and corresponding traffic and service parameters.

These differences have to be overcome when mapping IntServ onto ATM without losing the semantics of the IntServ specifications. The IETF has proposed some guidelines for the mapping of the QoS models in [10], but these have been shown to be arguable in [9].

2.2 QoS Procedures

While it is not easy to map the QoS models of the Internet and ATM, it is even more difficult to map their QoS procedures onto each other. This is due to the fact that they are built upon very different paradigms. While the signalling protocols of ATM are still based on the call paradigm used for telephony, the IETF viewed the support of a flexible and possibly large-scale multicast facility as a fundamental requirement [5]. The most prominent differences between RSVP and ITU-T's Q.2931 [12], on which all ATM signalling protocols are based, are:

Dynamic vs. Static QoS. RSVP supports a dynamic QoS, i.e. the possibility to change a reservation during its lifetime. ATM's signalling protocols however are providing only static QoS so far.

Receiver- vs. Sender-Orientation. The different design with regard to the initiation of a QoS reservation reflects the different attitudes regarding centralized vs. distributed management, and also that the RSVP/IntServ architecture had large group communication in mind while the ATM model rather catered for individual and smaller group communication.

Transmission of Control Messages. While in ATM separate control channels are used for the transmission of control messages of the signalling protocols, RSVP uses best-effort IP to send its messages.

Hard State vs. Soft-State. The discrepancies between the ATM QoS architecture and the IntServ architecture in how the state in intermediate systems is realized is another impediment to the interworking of both worlds since it leads to very different characteristics of the two QoS architectures.

Resource Reservation Independent or Integrated with Setup/Routing. The separation of RSVP from routing leads to an asynchronous relation of reservation and flow setup, and further enables an independent evolution of routing and resource reservation mechanisms. However, a possibly major disadvantage may be that QoS routing is much more difficult to achieve than with ATM's integrated connection setup/resource reservation mechanism (P-NNI [2] already supports a form of QoS routing).

Multicast Model. A further issue is the mapping of the IP multicast model on the signalling facilities in ATM for multi-party calls. While IP multicast allows for multipoint-to-multipoint communication, ATM only offers point-to-multipoint VCs to emulate IP multicast by either meshed VCs or a multicast server.

Heterogeneous vs. Homogeneous QoS. While ATM only allows for homogeneous reservations, RSVP allows heterogeneity firstly for different QoS levels of receivers and secondly for simultaneous support of QoS and best-effort receivers. This mismatch in the semantics of RSVP and Q.2931 is a major obstacle to simple solutions for the mapping of the two. This issue of heterogeneous vs. homogeneous QoS is the focus of our paper.

2.3 Heterogeneous vs. Homogeneous QoS

RSVP's heterogeneous reservations concept can, combined with heterogeneous transmission facilities, be very useful to give various receivers (e.g. in multimedia application scenarios) exactly the presentation quality they desire, and which they and the network resources towards the sender are able to handle. Such transmissions demand that the data to be forwarded can be somehow distinguished so that, e.g., the base information of a hierarchically coded video is forwarded to all receivers while enhancement layers are only forwarded selectively. This can be achieved by offering heterogeneity within one (network layer) session or by splitting the video above that layer into distinct streams and using multiple network layer sessions with homogeneous QoS. The latter approach has been studied by several authors, and found especially in form of RLM [15] wide-spread interest. Yet, if used widely and potentially even combined with object-oriented [11] or thin-layered coding schemes (e.g., [20]), this will lead to large numbers of multicast sessions, thus limiting its scalability.

Heterogeneity within one network layer session requires filtering mechanisms within intermediate systems. Such mechanisms are currently often considered as costly in terms of performance. However, we believe that with the evolution of ever faster routers, filtering will be possible at least outside the core area of networks and to do it at the network layer will be attractive for reasons such as scalability in terms of number of sessions and also simplification of applications.

The principle choices for an integration of the RSVP and ATM models with respect to heterogeneous reservations are:

- Ignore the problem and use just one QoS within the ATM subnetwork. As we will show, this is far from optimal with respect to resource consumption respectively costs if outside of the ATM cloud heterogeneous transmissions will exist.
- Change ATM to offer so-called "variegated VCs" where a different amount of data is forwarded to distinct multicast receivers. This requires the ability in switches to distinguish among information units (e.g., video frames). We do not believe that this will be possible on a cell basis in an efficient and useful way.
- Construct heterogeneous multicast trees from multiple homogeneous point-to-multipoint VCs. Here, for a certain receiver requesting a specific QoS it must be decided, e.g., whether one of the existing VCs can be used for it or whether a new one must be established. Hence, VC management mechanisms are needed.

We argue for the last alternative to be the most realistic and efficient one.

3 VC Management Strategies in Support of Heterogeneity

The main assumptions of the VC management approach for supporting heterogeneous RSVP reservations over ATM are:

- existence of mechanisms, e.g. filtering, to support heterogeneous multicast transmissions, and
- unavailability of variegated VCs in ATM devices.

The problem is to find a collection of point-to-multipoint VCs from which the heterogeneous RSVP multicast tree (the part which falls in the ATM network) is being constructed. The QoS of a particular point-to-multipoint VC must be allocated as the maximum of the RSVP requests (transformed into ATM terms) of the subnet-receivers of this point-to-multipoint VC, otherwise the traffic contract would be violated.

This problem is not specific to an RSVP over ATM environment, this is only the most prominent case. It exists in any scenario where a heterogeneous multicast QoS model is layered above a NBMA homogeneous multicast QoS model.

Before proposing new VC management strategies to support heterogeneity, we first discuss existing approaches to this problem.

3.1 Existing Approaches

The IETF working group ISSLL (Integrated Services over Specific Link Layers) is among other topics concerned with the mapping of RSVP/IntServ onto ATM networks, and particularly proposed in [3] the following models to support heterogeneous reservations over an ATM subnetwork:

Full Heterogeneity Model. In the full heterogeneity model (see Figure 1), point-to-multipoint VCs are provided for all requested QoS levels plus an additional point-to-multipoint VC for best effort receivers. This leads to a complete preservation of the heterogeneity semantics of RSVP but can become very expensive in terms of resource usage since a lot of data duplication takes place.

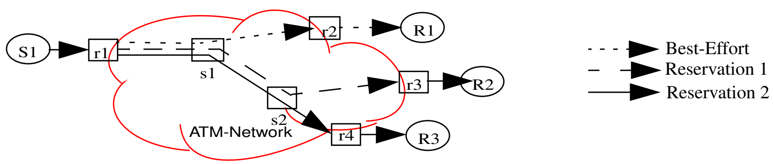

Figure 1: The Full Heterogeneity Model.

Limited Heterogeneity Model. In the limited heterogeneity model (see Figure 2), one point-to-multipoint VC is provided for QoS receivers while another point-to-multipoint VC is provided for best-effort receivers. Here, we assumed that reservation 2 is larger than reservation 1.

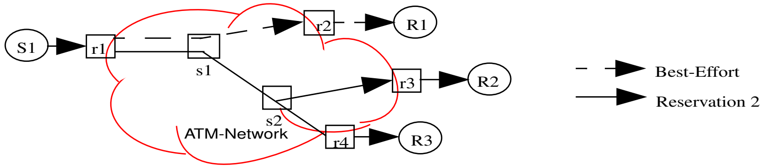

Figure 2: The Limited Heterogeneity Model.

A design question of this model is whether the best-effort VC is provided for all sessions together or one per session. The limited heterogeneity model strongly restricts RSVP's heterogeneity model to simply the differentiation of QoS and best-effort receivers. A further problem is that a single high QoS request can avoid the setup of a QoS VC.

Homogeneous Model. In the homogeneous model solely one point-to-multipoint QoS VC is provided for all receivers including the best-effort receivers. The QoS VC is dimensioned with the maximum QoS being requested. This model is very simple to implement and saves VC space in comparison to the full heterogeneity model, but may waste a lot of bandwidth if the resource requests are very different. A further problem is that a best-effort receiver may be denied service due to a large RSVP request that prevents the setup of a branch from the existing point-to-multipoint VC to that receiver. This is unacceptable to IntServ's philosophy of always supporting best-effort receivers. The modified homogeneous model takes that into account.

Modified Homogeneous Model. The modified homogeneous model behaves like the homogeneous model, but if best-effort receivers exist and if these cannot be added to the QoS VC, a special handling takes place to setup a best-effort VC to serve these. Thus it is very similar to the limited heterogeneity model. However, since the best-effort VC is only setup as a special case it is a little bit more efficient than the limited heterogeneity model with regard to VC consumption. On the other hand, it may be argued that best-effort VCs will be needed all the time, at least in the backbone, and thus it might be cheaper to leave the best-effort VCs open all the time, i.e., to use the limited heterogeneity model.

Another, quite different architecture for mapping RSVP/IntServ over ATM is proposed in [16]. With respect to heterogeneity support the authors introduce the:

Quantized Heterogeneity Model: This model represents a compromise between the full heterogeneity model and the limited heterogeneity model, by supporting a limited number of QoS levels, including the best-effort class, for each RSVP multicast session. Each QoS level maps into one point-to-multipoint VC.

While this proposal is an improvement over the very rigid models proposed by ISSLL, it says nothing about how to allocate the supported QoS levels for a RSVP multicast session. That means the concrete VC management decisions are left open to the implementor of an edge device (or rather the so-called Multicast Integration Server (MIS) in this architecture, for details see [8]). How to make these decisions in an efficient manner is exactly what we will deal with in the rest of this section.

3.2 Administrative Location of the Edge Device

In Figure 3 the basic network configuration when overlaying RSVP/IntServ over an ATM subnetwork is illustrated. Here, different administrative locations of the so-called edge devices (also called subnet-sender/receiver, virtual source/destination) are distinguished.

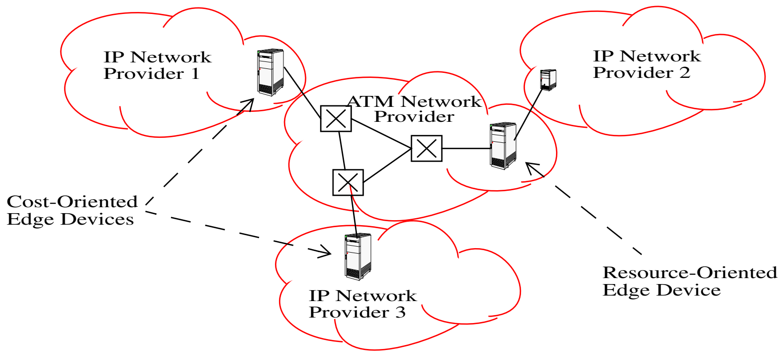

Figure 3: Different Types of Edge Devices.

Let us suppose that each of the networks is operated by a different provider. We can distinguish two cases:

1. The edge device is on the premises of the IP network provider (which is an ATM services customer of the ATM network provider), as e.g. for IP network provider 1 and 3. In this case, the edge device will make its VC management decisions depending mainly on the ATM tariffs offered by the ATM network provider. Therefore, we call it a *cost-oriented edge device*.

2. The edge device is on the premises of the ATM network (which is now offering RSVP/IP services to its customer, the IP network provider), as e.g. for IP network provider 2. Here, the edge device will try to minimize the resource consumption when taking decisions for VC management. Thus we call it *resource-oriented edge device*.

If, for example, IP network provider 1 and the ATM network provider would be the same administrative entity, then we would have the same situation as for case 2, i.e., a resource-oriented edge device.

While the ATM tariffs are the most important criterion for assessment of different alternatives for VC management decisions in case 1, the local resources consumed by a VC management strategy should also be taken into consideration, but rather as a constraint than an optimization criterion.

In most cases, prices will probably correlate positively with resource consumption, however, they will for several reasons not be related directly to them or in a much coarser granularity. Therefore, from a global perspective, case 2 is potentially a “better” configuration, because it will tend to use resources more efficiently than case 1, except if prices are a very accurate representation of the actual resource consumption. It is difficult to judge today, which configuration will be more likely. While telecommunication providers try to provide more value-added services and would thus be interested to operate the edge device, Internet service providers increasingly tend to use their own backbones instead of leasing lines from telecommunication providers, so that the edge device and the ATM network would be on the same premises.

In the VC management algorithms below it is ensured that subnet-receivers get at least the QoS they requested, but may even get better service and must thus be prepared to cope with additional data. If some of them cannot cope with the additional data then these restrictions have to be incorporated as additional constraints into the VC management strategies.

3.3 VC Management for Cost-Oriented Edge Devices

We will start considering the problem of supporting heterogeneity over an ATM subnetwork by VC management strategies for the case of a cost-oriented edge-device.

3.3.1 Static Case

In the static case, it is assumed that all receivers and their requests are known and that nothing changes throughout the session. While this is an idealistic view, the dynamic case discussed later can make use of the algorithms for the static case, since it can be viewed as a concatenation of static intervals. Let us start with a formal problem statement.

Problem Statement

Assume we have N different resource requests/RESV messages arriving at the ingress edge device.
Suppose the receivers are ordered by the size of their QoS request (if that is reasonably possible, e.g. by regarding only their bandwidth requirements) and denote them from 1 to N, i.e., 1 is the highest and N the lowest request.
Call R the set of all receivers, R = {1,..., N}.
Let

f(S,q) = costs for a point-to-multipoint VC from the subnet-sender to all $r \in S$ with QoS q;

c(S) = f(S, q(min S)) for $S \subseteq R$;

Call p = $\{R_1,...,R_n\}$ a partition of R, if $R_1 \cup \ldots \cup R_n = R$ and $\forall i,j$: $R_i \cap R_j = \emptyset$.
Thus, the problem is:

find p of R such that $\sum_{i=1}^{n} c(R_i)$ is minimized.

Note that p = {R} is the homogeneous model, while p={{1},..., {N}} is the full heterogeneity model.
To assess how difficult it is to find a cost-optimal p, consider the size of the partition space, $S_p(N)$:

$$|S_P(N)| = \begin{cases} \sum_{k=0}^{N-1} \binom{N-1}{k} |S_P(N-k-1)| & \text{if N > 1} \\ 1 & \text{if N = 0, 1} \end{cases}$$

This recursive formula can be explained by the observation that all partitions can be viewed as having receiver 1 and a k-elementary subset of the remaining (N-1) receivers as one point-to-multipoint VC and for the remaining point-to-multipoint VCs of the (N-k-1) receivers we have $|S_p(N\text{-}k\text{-}1)|$ alternatives (per definition). In Table 1 some example values of $|S_p(N)|$ are given.

It is obvious that for a high number of different reservation requests the partition space becomes too large to be searched exhaustively, while for smaller numbers this should still be possible. Keep in mind that N is the number of different reservation requests which should be bounded by the number of scaling levels the data transmission system is able to support (ignoring the possibility that receivers reserve different QoS levels even without a filtering support by the data transmission system, since they may accept that some of their traffic is degraded to best-effort).

Ways to Search the Partition Space

For larger N, the question is whether and how a search in the partition space can be kept feasible taking into account that the system must provide short response times (flow setup times are also a QoS issue). There are potentially two alternatives to achieve this:

- giving up the search for the optimal solution and just looking for a "good" solution using a heuristic to search the partition space, or,
- showing that some parts of the partition space can be excluded from the search either because it is impossible to find the global minimum there, or it is at least unlikely (using a heuristic to limit the reasonable partition space). In the following, we describe an approach for that.

For large N (take e.g. N=15, then one obtains $|S_p(15)|$ = 1,382,938,768 possible partitions) even a combination of these two techniques might be necessary.

Limiting the Search Space

An example how the characteristics of the price function can simplify the problem by allowing to limit the search on a sub-space of the complete partition space (without giving up the search for the optimum) is given by:

Theorem 1: If the price function f is subject to

$$f(S \cup r, q) - f(S, q) = K(q) \;\; \forall r \in R, S \subset R, S \neq \emptyset \;\; \wedge \;\; K(q) \text{ strictly increasing in q}$$

then the cost-optimal partition p^{opt} is an "ordered partition" (see definition below).

The proof of Theorem 1 can be found in the appendix.

Definition: The partition p = $(R_1,..., R_n)$ is called ordered if for all R_i and any k,l $\in R_i$ with k < l, it applies that k+1,...,l-1 are also $\in R_i$.

The above shows that under the assumptions being made it is possible to restrict the search on the sub-space of ordered partitions, which gives a considerable reduction on the number of candidates for the optimal solution. The assumption about the price function essentially means that the price of adding a receiver to an existing VC is not dependent on the particular receiver to be added or the already existing point-to-multipoint VC. However, it is depending on the QoS of that point-to-multipoint VC in a positively correlated manner, i.e. for a higher QoS it is more expensive to add a receiver to an existing point-to-multipoint VC. It may be arguable whether real price functions actually conform to the prerequisite of Theorem 1 or not. The point is that if they do, the search can be restricted to ordered partitions.

The sub-space of ordered partitions, $S_{oP}(N)$**,** is considerably smaller than the complete partition space:

$$|S_{oP}(N)| = \sum_{k=1}^{N} A(N, k)$$

where A(N,k) is the number of partitions with n = k (with n as the number of point-to-multipoint VCS as defined above) and is defined as follows

$$A(N, k) = \begin{cases} \sum_{i=1}^{N-k+1} A(N-i, k-1) & \text{if } 1 < k < N \\ 1 & \text{if } k = 1, N \end{cases}$$

Actually, it turns out that (see appendix for proof):

Theorem 2: $|S_{oP}(N)| = 2^{N-1}$.

The actual sizes of the complete partition space and the ordered partition space are given in Table 1.

N	2	3	4	5	6	7	8	9	10	15
$\lvert S_P(N)\rvert$	2	5	15	52	203	877	4140	21147	115975	1382938768
$\lvert S_{oP}(N)\rvert$	2	4	8	16	32	64	128	256	512	16384

Table 1: Growth of the Complete Partition Space and of the Ordered Partition Space.

Even if a price function does not conform to the prerequisite in Theorem 1, then it is probably still very reasonable for larger N to only explore the ordered partition space, where at least some "good" solutions should be found. However, optimality can no longer be guaranteed. It depends on the actual form of the price function how far the actual optimum may be away from the optimum within the ordered partition space. Our conjecture is that for realistic price functions it should not deviate too much, yet more work on the topology of cost functions over the partition space would be needed to prove this quantitatively.

One may argue that even the ordered partition space is too large for higher values of N. In that case heuristic search methods on the ordered partition space would be needed. (In the section on resource oriented edge devices we present such a heuristic which can easily be adjusted for a cost-oriented edge device).

3.3.2 Dynamic Case

Now we take a dynamic view on the problem and investigate VC management strategies when the set of different receivers is changing in time, i.e., instead of R we now have R^t with discrete time steps t=0,1,2,...Thus we can view the search for the cost-optimal partitions of R^t as a series of static case problems, which however have a certain relationship. This observation leads to the idea of reusing the approaches for the static case, where the crucial question is how to take the relationship between the series of static problems into account.

A straightforward, but compute-intensive algorithm could be to always recompute the statically optimal partition and then make the minimally necessary changes to the current partition to transform it into the new one.

Besides its high computational complexity this algorithm may potentially produce a lot of changes in the membership of receivers because it does neglect the relationship between successive R^t. Such changes of receivers from one point-to-multi-point VC to the other produce costs, which should be incorporated into the decision process, i.e., we need to minimize a transformed cost function:

$$min.\ c^*(p) = c(p) + t(p^{old}, p)$$

where

$t(p^{old}, p)$ are the costs of transforming the existing partition p^{old} into the partition p.

Both algorithms have the same complexity in principle, but the transformed cost function c* will likely be amenable to a local search in the neighborhood of the existing partition, since partitions far "apart" in the partition space get a high penalty from the transformation costs t.

A simple idea for such a local search could be to always try all incremental "adds", i.e. either adding the new (or modified) receiver to an existing point-to-multi-point VC or setting up a new VC for that receiver, and take the one that minimizes c*.

However, it must be realized that after a certain number of time steps this algorithm might deviate considerably from the optimum VC management strategy. Therefore, an improvement may be to compute the statically optimal partition from time to time and compare it to the current partition with respect to the original cost function c. If it deviates too much, a substantial reorganization of the partition may pay off in the long term, even if c* is higher at the moment. The idea of this approach is to use the optimal VC management strategy from the static case as a corrective measure for the dynamic case.

What is missing from all these considerations for cost-oriented edge devices is the local resource consumption at the edge device. This will be higher for strategies consuming more VCs and should thus be taken into account as

$$\bar{c}(p) = \sum_{i=1}^{n} c(R_i) + C(n)$$

where C(n) represents the local resource consumption for managing n point-to-multipoint VCs.

This is however difficult since the two terms are incommensurable and the addition is thus not easily possible (it would require a translation of local resource consumption into monetary costs). Therefore, we propose to either assume that the VC management at the edge is not a bottleneck (i.e. the edge device is dimensioned so that it is powerful enough to manage very large numbers of VCs), or to incorporate its limitations as a constraint into the search. An example could be to require for all partitions $p=\{R_1,...,R_n\}$, that, e.g., $n < 6$, or a similar possibly more sophisticated condition.

3.4 VC Management for Resource-Oriented Edge Devices

Now we will consider the case where the edge device is operated as part of the ATM network and thus manages its VCs with the objective of minimizing the resource consumption inside the ATM network. Resources inside the ATM network can be viewed on different abstraction levels, with the lower levels containing details like internal buffers of the ATM switches, switching fabrics, control processors, etc. For our purposes it is however necessary to look at higher abstraction levels of the resources of an ATM network in order to keep the complexity of the problem manageable. Thus, the resources we take into consideration are:

- bandwidth of links between ATM switches or ATM switches and edge devices, and/or
- VC processing at switches and edge devices.

At first, we consider again the static case, before taking into account the dynamic nature of the problem following the same rationale as for cost-oriented edge devices.

3.4.1 Static Case

The situation is actually very similar to that of cost-oriented edge devices with the difference that resource consumption is taken as a substitute for the cost function. If resource consumption can be expressed as a single valued function then, more or less, the same considerations apply as for a cost-oriented edge device, although it is very unlikely that assumptions like that of Theorem 1 will apply for resource consumption functions, since these functions will be much more complex due to their topology-dependence. Moreover, if we really want to make use of the further information that is available to a resource-oriented edge device (e.g. by taking part in the PNNI protocol or by static configuration), then different resources must be taken into account, which again raises the incommensurability problem. Now we can either treat it as a multi-criteria decision making problem or we try to find a translation and a weighting between the different criteria. As mentioned above, we will restrict our considerations to the abstract resources link bandwidth and VC processing in order to alleviate such complexities.

At first, let us even assume that only link bandwidth is taken into account. A greedy algorithm that always picks the locally best decision and operates on the sub-space of ordered partitions would be the following:

```
k = 1;
V = R;
WHILE (V NOT empty) DO
   R[k] = min V;
   V = V - {min V};
   L' = INFINITY;
   WHILE (V NOT empty) AND (L < L') DO
      H = union(R[k], {min V});
      L = link bandwidth consumption of H;
      L' = link bandwidth consumption of R[k] +
           link bandwidth consumption of {min V};
      IF (L <= L')
         R[k] = H;
         V = V - {min V};
   k++;
```

With link bandwidth consumption of a set of receivers we mean the sum of bandwidth consumptions per link for the point-to-multipoint VC which would be built from the edge device to the subnet-receivers, while the rest of the notation is analogous to the definitions in the section on cost-oriented edge devices (with `V` and `H` as auxiliary sets of subnet-receivers).

The heuristic that is essentially applied by that greedy algorithm is to group together adjacent requests, where adjacency is defined with respect to topology and resource requirements. This is due to the observation that it will make little sense to have very different (with respect to their reservations) receivers in the same point-to-multipoint VC if they are far apart from each other, because that would waste a lot of bandwidth for the part of the point-to-multipoint VC that is unique to a receiver with low resource requirements.

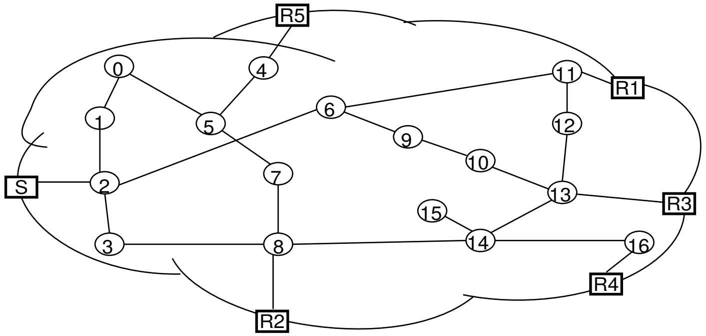

Figure 4: Example Network.

To show what results can be achieved with that simple algorithm consider the example network in Figure 4, which represents a model of the topology of the NSF backbone as of 1995 [13]. Here, circles are ATM switches and boxes are edge devices, which either act as subnet-sender or subnet-receivers. Let us suppose that the following reservations have been issued by the subnet-receivers:

R1 = 10 Mb/s, R2 = 8 Mb/s, R3 = 4.5 Mb/s, R4 = 3 Mb/s and R5 = 2 Mb/s.

Applying the algorithm to the example network gives the partition GA={{R1,R2}, {R3,R4},{R5}} with L(GA)=118 as the sum of link bandwidth consumption of the three point-to-multipoint VCs (using Steiner trees). Compare this to the full heterogeneity model, FH={{R1},...,{R5}}, with L(FH)=129, or the homogeneous model, H={{R1,...,R5}}, with L(H)=180. So, H consumes about 50% more bandwidth inside the ATM network than R. Actually (as a total enumeration shows), GA is the optimal partition (with respect to link bandwidth consumption). Interestingly, if VC consumption is taken into account then FH is dominated by GA, i.e., it is worse with respect to both, link bandwidth consumption and VC usage. This is certainly not the case for H, but the saved bandwidth will probably still be a major point for choosing GA.

The greedy algorithm, of course, does not guarantee an optimal solution. Consider for example that now R3=5Mb/s, and everything else unchanged. Then the algorithm gives GA={{R1,R2,R3}, {R4},{R5}} with L(GA)=130, but the optimal partition O={{R1,R2},{R3,R4},{R5}} has L(O) = 122 (L(FH) = 132 and L(H)=183 for this configuration).

While for these examples only ordered partitions were optimal, it should be noted that this is not necessarily the case as the simple example in Figure 5 shows:

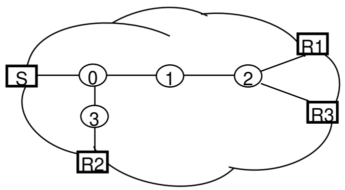

Figure 5: Example of an Unordered Optimal Partition.

Suppose that: R1 = 9 Mb/s, R2 = 5.5 Mb/s and R3 = 3 Mb/s.

Then the algorithm gives GA={{R1},{R2},{R3}} with L(GA)=64.5, while the optimal partition is O={{R1,R3},{R2}} with L(O)=61,5 (L(FH=GA) = 64.5, L(H) = 63).

We have discussed above how to take into account the VC processing resource in principle. For the greedy algorithm there is a straightforward extension in order to incorporate the additional criteria into the construction of a “good” partition. This would be to change the IF statement at the end of the inner loop into:

```
IF (L <= L' + delta) // saves VCs
```

where `delta` would have to be chosen reasonably in order to force the construction of larger point-to-multipoint VCs with respect to number of members. It is certainly not obvious how to choose `delta`, so further study of that parameter is needed.

3.4.2 Dynamic Case

The results for cost-oriented edge devices when considering the dynamic case are directly applicable to resource-oriented edge devices as well. Again the dynamic problem can be regarded as a series of static problems, whereby the current partition should somehow be taken into account when reacting to changes and building a new partition.

A particular issue for resource-oriented edge devices when considering the dynamic case is the dynamics of existing reservations. While the changes due to these dynamics can be treated just like a new receiver joining the session with the modified reservation and the existing receiver leaving it, these actions should be minimized since they are either leading to temporary double reservations in the ATM network or to service interruptions for the receivers depending on the order of joining and leaving (presumably only joining before leaving is a commercially feasible option). The dynamics due to modified reservations are affected by the VC management strategy for heterogeneity support in the following way: they will be more probable for a fine-grained partition (larger n) than for a coarse-grained partition (smaller n).

4 Implementation Aspects: RSVP’s Traffic Control Interface

When considering the implementation of some of the above or any other VC management strategies in support of heterogeneity over an ATM subnetwork, RSVP’s Traffic Control Interface (TCI) and the relevant part of the protocol message processing rules as specified in ([7],[6]) must be made more flexible than they are (this does not violate these standards, because these parts are only informational). Currently, RSVP merges all downstream requests and then hands the merged reservations to the traffic control module via the TCI. This leads to two problems if operating over ATM, or in general, a NBMA subnetwork with capabilities for multipoint communication:

- potential for not recognizing new receivers,
- solely support for the homogeneous QoS model.

These problems are already realized in [7], where it is conceded that the proposed TCI is only suitable if data replication takes place in the IP layer or the network (i.e. a broadcast network), but not in the link-layer as would be the case for ATM. Here, different downstream requests should not necessarily be merged before being passed to the traffic control procedures.

A new general interface is needed that supports both, broadcast networks and NBMA networks, where the replication can also take place in intermediate nodes (e.g. ATM switches) of the NBMA subnet. Only such modifications will allow for heterogeneity support over an ATM network, i.e. different VCs for different QoS receivers. However, even without taking into account heterogeneity support, there is a need for a modification of the TCI and the message processing rules due to the different nature of NBMA networks.

If a reservation request is received from a new next hop in the ATM network that is lower than an existing reservation for the session, then according to the currently proposed processing rules no actions will be taken, since it is assumed that all the next hops within the same outgoing interface will receive the same data packets. That is of course not the case for an NBMA network like ATM, and some actions must be taken to add this new receiver to the existing point-to-multipoint VC. The same situation arises when a receiver tears down its reservation.

If the LUB (least upper bound) of the other reservations does not change, nothing will be done with the current processing rules. However, the receiver must be deleted from the point-to-multipoint VC.

The problem with the current message processing rules and TCI is that, since they are based upon broadcast mediums, they do not allow any heterogeneity within a single flow and an outgoing interface. This is due to the fact that broadcast networks do not allow for heterogeneity of the transmission anyway. That is the reason why the LUB of the reservations requested for that interface is computed, thus making downstream merging.

A VC management strategy that supports heterogeneity does not need this downstream merging, or at least, no downstream merging of all the next hops in the interface. A more flexible scheme is necessary, that permits different "Merging Groups" within a certain interface. This general model includes the current model, if all next hops are considered as one merging group. A *Merging Group* (MG) is defined as the group of next hops with the same outgoing interface, whose reservation requests for a certain flow should be merged downstream, in order to establish a reservation (Figure 6).

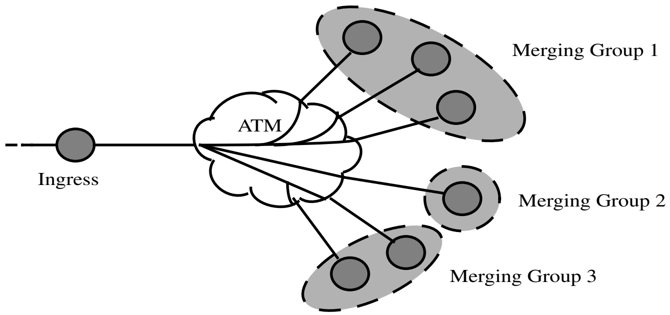

Figure 6: Merging Groups.

For a single flow and outgoing interface, there may be several MGs. The two extreme cases are:

a) Only one MG: This is the case when no heterogeneity is allowed within the interface. Examples of this situation are:

- the homogeneous model when implementing RSVP over ATM,
- the underlying network technology is broadcast (e.g. Ethernet).

b) As many MGs as next hops: this would be the case if each of the next hops requires a dedicated reservation. Example applications of this are:

- NBMA networks which do not allow point-to-multipoint connections, and therefore, a point-to-point connection is needed for each of the receivers,
- the full heterogeneity model when implementing RSVP over ATM.

The most interesting options of this model from our point of view are the intermediate points between these two cases, where we allow a certain degree of downstream merging, so that it is possible to take advantage of the VC management strategies for heterogeneity support. The TCI and the message processing rules should be independent of the number of MGs for a specific flow and the decision of including one next hop into a group or another should be taken by the traffic control module and not as part of the RSVP message processing. For details on how RSVP's TCI and its message processing rules need to be modified to allow for VC management strategies in support of heterogeneity, see [17].

5 Summary and Conclusion

In this paper we presented approaches to the efficient solution of one of the difficult problems when mapping RSVP onto ATM subnetworks, namely the problem of providing heterogeneous reservations across an ATM subnetwork. Since ATM only provides homogeneous QoS within one connection, we argued for using several ATM VCs to provide different levels of QoS for subnet-receivers that requested different resources. The management of several VCs per RSVP session gives a large number of possible strategies. We introduced some algorithms which try to minimize costs respectively resource consumption depending on the administrative location of the IP/ATM edge device. Furthermore, we discussed briefly how the RSVP TCI and the RSVP message processing should be extended/generalized in order to support heterogeneity over an NBMA network like ATM.

It can be concluded that if heterogeneity turns out to be an interesting feature of a reservation mechanism on the network layer, then different alternatives for "emulating" heterogeneity over an ATM network can vary considerably with respect to their resource consumption and costs. Thus it will be commercially attractive to choose a "good" alternative (preferably the optimal one, if it can be determined).

This paper studied only one of the problems of mapping RSVP/IntServ onto ATM and proposed solutions for this – much remains to be done. As pointed out in Section 2, there are several other difficult problem areas. For further work in the direction of supporting heterogeneity over an ATM network via VC management strategies, it will be interesting to evaluate more quantitatively the effect of different cost/resource consumption functions, different topologies, and different combinations of heterogeneous reservations and how much can be gained by using an “intelligent” VC management strategy.

References

[1] ATM Forum Technical Commitee: Traffic Management (TM) Specification 4.0, April 1996.

[2] ATM Forum Technical Committee: Private Network-Node Interface (PNNI) Signalling Specification, March 1996.

[3] L. Berger, E. Crawley, S. Berson, F. Baker, M. Borden, and J. Krawczyk. A Framework for Integrated Services with RSVP over ATM, August 1998. RFC 2382.

[4] D. Black, S. Blake, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services, December 1998. RFC 2474.

[5] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: an Overview, June 1994. RFC 1633.

[6] R. Braden and L. Zhang. RSVP Version 1 Message Processing Rules, September 1997. RFC 2209.

[7] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource Reservation Protocol (RSVP) - Version 1 Functional Specification, September 1997. RFC 2205.

[8] A. Corghi, L. Salgarelli, H. Sanneck, M. Smirnov, and D. Witaszek. Supporting IP Multicast Integrated Services in ATM Networks, November 1997. Internet Draft, work in progress.

[9] P. Francis-Cobley and N. Davies. Performance Implications of QoS Mapping in Heterogeneous Networks Involving ATM. In *Proc. of IEEE Conference on ATM ’98 (ICATM’98)*. IEEE, June 1998.

[10] M.W. Garrett and M. Borden. Interoperation of Controlled Load and Guaranteed Service with ATM, August 1998. RFC 2381.

[11] ISO/IEC JTC1/SC29/WG11: MPEG-4 Systems Final Committee Draft, March 1998.

[12] ITU-T: Rec. Q.2931: B-ISDN User-Network Interface Layer 3 Specification for Basic Bearer/Caller Control, March 1994.

[13] J. Jamison and R. Wilder. vBNS: The Internet Fast Lane for Research and Education. *IEEE Communications Magazine*, 35(1), January 1997.

[14] V.P. Kumar, T.V. Lakshman, and D. Stiliadis. Beyond Best Effort: Router Architectures for the Differentiated Services of Tomorrow’s Internet. *IEEE Communications Magazine*, 36(5), May 1998.

[15] S. McCanne, V. Jacobson, and M. Vetterli. Receiver-driven Layered Multicast. In *Proc. of ACM SIGCOMM’96*, August 1996.

[16] L. Salgarelli, A. Corghi, H. Sanneck, and D. Witaszek. Supporting IP Multicast Integrated Services in ATM networks. In *Proc. of SPIE Voice and Video ’97, Broadband Networking Technologies*. SPIE, November 1997.

[17] J. Schmitt. Extended Traffic Control Interface for RSVP. Technical Report TR-KOM-1998-04, Darmstadt University of Technology, July 1998.

[18] S. Shenker, C. Partridge, and R. Guerin. Specification of Guaranteed Quality of Service, September 1997. RFC 2210.

[19] J. Wroczlawski. Specification of the Controlled-Load Network Element Service, September 1997. RFC 2211.

[20] L. Wu, R. Sharma, and B. Smith. Thin Streams: An Architecture for Multicasting Layered Video. In *Proc. of NOSSDAV ’97*. IEEE, May 1997.

Appendix

Proof of Theorem 1:

Suppose $p^{opt} = \{R_1, ..., R_n\}$ is not ordered, then there is at least one pair $R_i = \{i_1, ..., i_k\}$, $R_j = \{j_1, ..., j_v\}$ with $i_1 <. ... < i_m < j_1 < ... < i_k < ... < j_v$ (without loss of generality we assume $j_l < i_k$).

Now let $\overline{R}_i = \{i_1, ..., i_m\}$ and $\overline{R}_j = \{j_1,...,i_k, ..., j_v\}$.

Thus, we have:

$c(\overline{R}_i) + c(\overline{R}_j) = f(\overline{R}_i, q(i_1)) + f(\overline{R}_j, q(j_1))$

$= f(R_i, q(i_1)) - (k-m)K(q(i_1)) + f(R_j, q(j_1)) + (k-m)K(q(j_1))$

$= f(R_i, q(i_1)) + f(R_j, q(j_1)) + (k-m)(K(q(j_1)) - K(q(i_1)))$

$< f(R_i, q(i_1)) + f(R_j, q(j_1))$ (since $q(i_1) > q(j_1)$ and K is strictly increasing in q)

$= c(R_i) + c(R_j)$

That means for $\overline{p} = (p^{opt}/\{R_i, R_j\}) \cup \{\overline{R}_i, \overline{R}_j\}$ applies:

$c(p) < c(p^{opt})$

which contradicts the cost-optimality, and thus p^{opt} must be an ordered partition (under the assumptions being made).

■

Proof of Theorem 2:

by induction:

N=1: $|S_{oP}(1)| = 1 = 2^0$

'N->N+1': $|S_{oP}(N+1)| = \sum_{k=1}^{N+1} A(N+1,k) = 2 + \sum_{k=2}^{N} \sum_{i=1}^{N-k+2} A(N+1-i,k-1)$

$$= 2 + \sum_{k=2}^{N} \sum_{i=0}^{N-k+1} A(N-i,k-1) = 2 + \sum_{k=2}^{N} \left(A(N,k-1) + \sum_{i=1}^{N-k+1} A(N-i,k-1) \right)$$

$$= 2 + \sum_{k=1}^{N-1} A(N,k) + \sum_{k=2}^{N} A(N,k) = 2\sum_{k=1}^{N} A(N,k) = 2(|S_{oP}(N)|) = 2^N$$

■