Darmstadt University of Technology

Issues in Overlaying RSVP and IP Multicast on ATM Networks

Jens Schmitt, Javier Antich

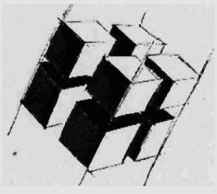
August 1998

Technical Report TR-KOM-1998-03

Industrial Process and System Communications (KOM)

Department of Electrical Engineering & Information Technology
Merckstraße 25 • D-64283 Darmstadt • Germany

Phone: +49 6151 166150
Fax: +49 6151 166152
Email: info@KOM.tu-darmstadt.de
URL: http://www.kom.e-technik.tu-darmstadt.de/



Issues in Overlaying RSVP and IP Multicast on ATM Networks

Jens Schmitt[1]
[1]
Industrial Process and System Communications
Dept. of Electrical Engineering & Information Technology
Darmstadt University of Technology
Merckstr. 25 • D-64283 Darmstadt • Germany

{Jens.Schmitt}@kom.tu-darmstadt.de

Abstract

Internet and ATM both aim at providing integrated services. Therefore they independently (more or less) developed QoS architectures. A realistic assumption certainly is that both will take their place and that they will coexist for quite some time. A likely place for ATM is in the backbone, while IP will probably keep its dominance on the desktop. It is thus valid to assume an overlay model for the interaction between the two QoS architectures.

Crucial components of the QoS architecture of the Internet are its signalling protocol RSVP and the IP multicast architecture. Their integrated support by an ATM subnetwork and the issues arising with this are the focus of this paper. While other components of the QoS architectures like the QoS models, the management frameworks, the charging mechanisms, etc. need to be dealt with for a complete solution to the problem of overlaying IP's QoS architecture onto that of ATM, we will concentrate on the aforementioned points.

Keywords: QoS, Integrated Services, RSVP, IP Multicast, ATM.

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 [BZB+97],[SPG97],[Wro97]) 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 [BBC+98] 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 [KLS98]. 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.

One of the most important points in this integration is the mapping of the Internet's signalling protocol RSVP onto ATM mechanisms. Many and the most difficult problems in this area arise for the multicasting of data. The anticipated new services of a future Internet will beyond others be 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.

This work is supported in part by a grant of Volkswagen-Stiftung, D-30519 Hannover, Germany.

IP Multicast, particularly in conjunction with RSVP, has some characteristics that make its support over ATM networks a difficult problem – or viewed from another perspective: ATM falls short in providing mechanisms to support IP multicast efficiently. While IP Multicast allows for an anonymous, egalitarian, dynamic n:m multicast model, ATM supports a non-anonymous, master-slave 1:n model. The key tasks in order to support IP multicast over ATM are:

- group membership management,
- VC management and
- heterogeneity support.

Existing approaches for overlaying RSVP and IP multicast over ATM were mostly separated from each other and typically tailored to specific environments like best-effort transmission, restricted scale, exclusion of multicast, specific and limited ATM network structures, etc. Therefore none of these is *the* solution to the general problem of overlaying RSVP/IP multicast onto ATM networks. We claim that a more integrated view is necessary. However, we perceive the intractability of the general problem in one step, thus assumptions are necessary to relax the complexity. The main assumption is with regard to the role of ATM networks in future. Different views can be taken:

1. ATM networks will not be deployed widely, not even in backbone networks.
2. ATM networks will only work as backbone of the Internet and will be used only to provide big pipes of data by means of point-to-point connections. In this case ATM would work as a pure data link layer.
3. ATM networks will be widely deployed and interconnected in such a way that all networks can be viewed as a unique network. In this case, LANs may not be ATM based, but the LANs would be connected to the large ATM network.
4. Hosts will be directly connected to the ATM network.

Henceforth, we will consider that ATM networks are widely deployed and that mainly but not only LANs (by means of ATM attached routers) but also hosts, can be connected to the ATM network. Another assumption that should be made is that multicast services will be multimedia services, not only with QoS requirements, but also with long duration (minutes, maybe hours), which makes the connection establishment time negligible.

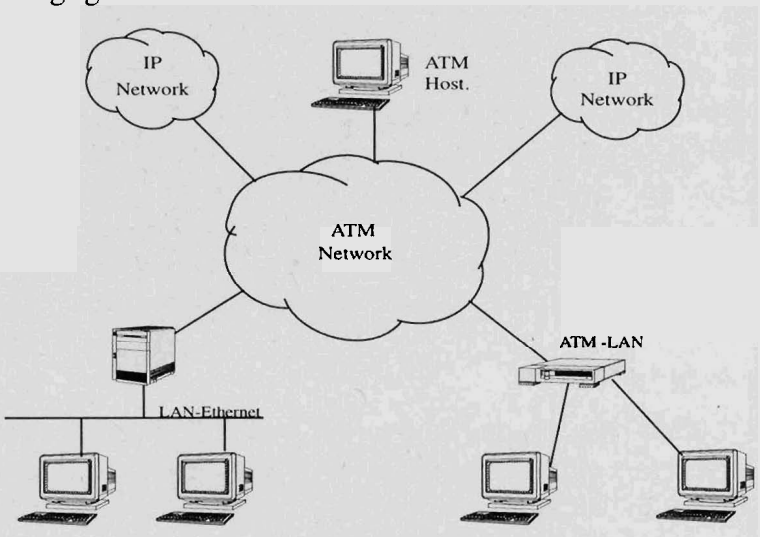


Figure 1: Network structure assumption.

Since nowadays the Internet is a multi-provider network even if only the backbone is regarded it is certainly 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.

In this report we take a look at some of the harder problems when mapping the combination of RSVP and IP Multicast onto ATM networks and give solution approaches to these under some restricting assumptions as, e.g., with regard to the scale of the ATM network. 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. We will treat this particular problem in more detail in section 13, where we will show how this difference can be bridged to allow for efficient support of RSVP over ATM. After this detailed treatment of one particular issue for mapping RSVP and IP Multicast onto ATM we will conclude the report and will give the literature references to standards and related work being made throughout the text.

1 Integrated Services IP Multicasting

In this section we review the most important characteristics of IP multicast in conjunction with RSVP.

1.1 IP Multicasting

The notion of a group is essential to the concept of IP multicast. By definition a multicast message is sent from a source to a group of destination hosts. In IP multicasting, multicast groups have an ID called multicast group ID. Whenever a multicast message is sent out, a multicast group ID specifies the destination group. These group ID´s are essentially a set of IP addresses called "Class D". Therefore, if a host (a process in a host) wants to receive a multicast message sent to a particular group, it needs to join that group. If the source and destination of a multicast packet share a common bus in a LAN, multicasting is easy within that LAN. However, if the source and destination are not on the same subnetwork, forwarding the multicast messages to the destination becomes more complicated. To solve the problem of Internet-wide routing of multicast messages, hosts need to join a group by informing the multicast router on their subnetwork. The Internet Group Management Protocol (IGMP [Fen97]) is used for this purpose. This way multicast routers of networks know about the members of multicast groups on their network and can decide whether to forward a multicast message on their network or not. However, for delivering a multicast packet from the source to the destination nodes on other networks, multicast routers need to exange the information they have gathered from the group membership of the hosts directly connected to them. There are many different algorithms such as "flooding", "spanning tree", "reverse path broadcasting" and "reverse path multicasting" in order to exchange the routing information among the routers. Some of these algorithms have been used in dynamic multicast routing protocols such as Distance Vector Multicast Routing Protocol (DVMRP [Pus98]), Multicast extension to Open Shortest Path First (MOSPF [Moy94]), and Protocol Independent Multicast (PIM [EFH+97]). Based on the routing information obtained through one of these protocols, whenever a multicast packet is sent out to a multicast group, multicast routers will decide whether to forward the packet to their network(s) or not. Finally, the leaf router will see if there is any member of that particular group on its physically attached networks based on the IGMP information and decide whether to forward the packet or not.

1.2 IP Multicasting with RSVP

The use of RSVP [BZB+97], as signalling protocol for guaranteeing a specified quality of service to a flow, with IP multicasting, allows the emergence of multimedia applications that require both, multicast and quality of service. In fact, RSVP messages, which are encapsulated in IP packets (UDP is also possible), make use of the multicast extensions of IP.

When a source needs to send out multicast data to a multicast group, it sends IP packets to the IP group address of that group. In case the source is RSVP capable, it will also send PATH messages, in the same way it sends out its data packets, with the destination address of the group. Once, the members of the group receive the PATH messages, they can decide whether to ask for a reservation or not. In case

a receiver decides to make a reservation, it will send RESV messages upstream and the reservations will be established, according to the requested resources, available resources and other existing reservations.

Depending on the multicast routing protocol used, there will be a multicast tree with different reservations in its branches, maybe without reservation in some of them. This is possible if all the routers implement the RSVP protocol. Otherwise, techniques like tunneling, for example, may be used to connect RSVP capable nodes..

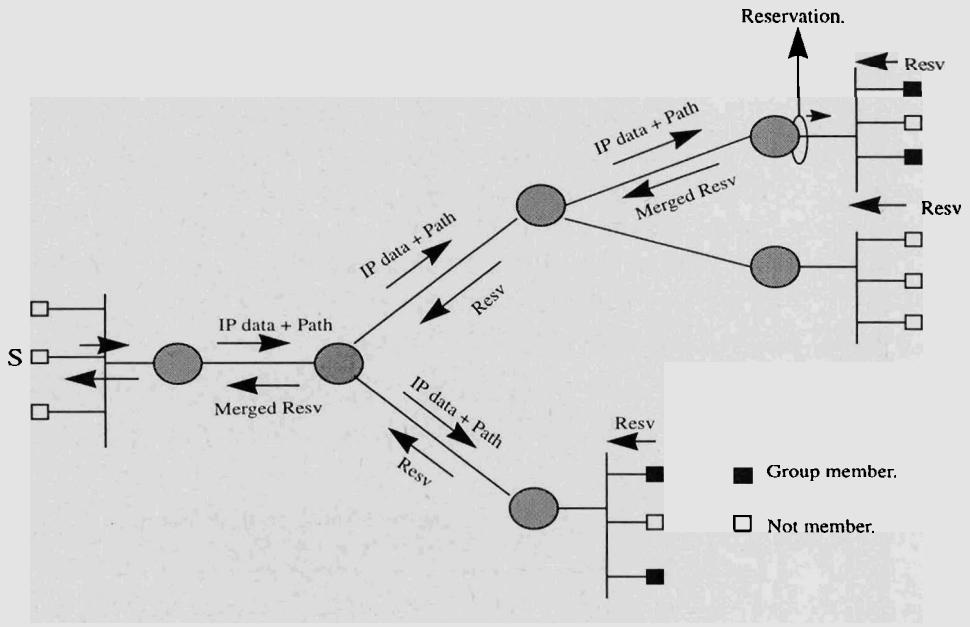


Figure 2: RSVP Concept.

The main characteristics of the pair RSVP-IP Multicast are:

- **Heterogeneous Receivers.** RSVP allows for receivers with heterogeneous resource requirements but in practice they are only allowed if they belong to different networks, that is to say, if they are connected to the router through different interfaces. Two or more receivers in the same LAN (i.e. Ethernet) requesting certain QoS for a flow, will all receive the same QoS, the largest of all the QoS requested.
- **Many-to-Many Multicast Communications.** The fact that the sender does not need to know the members of the group, but only the IP group address, facilitates multipoint-to-multipoint communications. A source must only send out its packets to the right group address and the routing protocols will find the "best" delivery tree for the packets.
- **QoS Renegotiation.** If the receiver sends out a new RESV message with different resource requirements, the reservations are changed (if necessary) in the nodes along the data path and also new RESV messages (if necessary) are sent out. These QoS changes can be done in any time during the data transmission, because the reservation is independent of the data transmission. In fact, there can be also data transmission without a reservation, like the normal IP best-effort service.
- **Soft state.** One of the main principles of the TCP/IP protocols is robustness.They are independent of the underlying network technology and able to work even if failures in the network appear. The connectionless characteristic of IP is an example. In order to allow resources reservation in the data path and go on being a connectionless service, RSVP has been designed to use soft state, which consists of the existence of state information of the reservation, in the nodes along the data path, that will be deleted if it is not refreshed periodically. This feature allows that if a change in the data path occurs, the state in the old path will be timed out and deleted.

2 Multicasting over ATM

In this section we review the multicast facilities available in ATM networks.

2.1 UNI 3.0/3.1

Multicasting is supported in ATM UNI 3.0/3.1 by means of point-to-multipoint VCs. In [ATM95], a point-to-multipoint connection is defined as a collection of associated VC or VP links connecting end-point nodes, of which one, the root node, has the property to send information, while all of the remaining nodes of the connection, called leaf nodes, receive copies of that information.

A point-to-multipoint connection is set up by first establishing a point-to-point connection between the root node and one leaf node. After this set up is complete, additional leaf nodes can be added to the connection by "ADD PARTY" requests from the root node. A leaf node may be added or dropped from a point-to-multipoint connection at any time after the establishment of the connection. A leaf node can be dropped from a connection as a result of a request sent by either the root node or by the leaf node to be dropped (but not by another leaf). Leaf nodes are identified by their unicast ATM address, since no multicast or group ATM address has been defined yet.

The ATM signalling messages utilized for establishing, adding and deleting nodes from point-to-multipoint VCs are shown in Table 1. See [ATM95] for more details on the message contents and connection establishment procedures.

ATM SIGNALLING MESSAGES
ADD PARTY
ADD PARTY ACKNOWLEDGE
ADD PARTY REJECT
DROP PARTY
DROP PARTY ACKNOWLEDGE

Table 1: ATM UNI 3.0/3.1 Messages.

2.2 UNI 4.0 Leaf Initiated Join

In previous versions of ATM signalling, only the root node was able to add leaf nodes to a point-to-multipoint connection. With version 4.0 of ATM UNI [ATM96b], leaf nodes can join to point-to-multipoint connections with or without intervention from the root of the connection. In [ATM96b], two different modes of operation are described, *Leaf-prompted join without root notification* and *Root-prompted join*. In the first mode, the root is not notified when a leaf node is added or dropped. In the second case, the leaf's request is handled by the root of the connection. This type of connection is referred to as *Root LIJ* connection. In order to join a specific point-to-multipoint connection the leaf must specify the so-called Global Call IDentifier (GCID) of that connection.

The new signalling messages included to support LIJ are:

ATM LIJ MESSAGES
LEAF SETUP REQUEST
LEAF SETUP FAILURE

Table 2: Messages for supporting LIJ.

3 Issues in Mapping RSVP/IntServ onto ATM Networks

Before going into the details of mapping RSVP/IP Multicast onto ATM networks we want to reconsider briefly 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.

3.1 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 [ATM96a] and the IntServ specifications ([SPG97], [Wro97]), 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 [GB98], but these have been shown to be arguable in [FCD98].

3.2 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 [BCS94]. The most prominent differences between RSVP and ITU-T's Q.2931 [ITU94], 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 [ATM96c] 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.

4 Issues in Implementing IntServ IP Multicast over ATM

After reviewing the general issues for mapping RSVP/IntServ onto ATM, let us now turn to the specific aspects which must be resolved for an efficient support of IntServ IP Multicast flows over ATM networks. These are:

- Group membership management.
- VC management for control and data traffic.
- Advanced VC management issues for data traffic:
  - heterogeneity support,
  - shortcut support,
  - aggregation,
  - dynamic QoS,
  - MC data distribution.

These aspects will be discussed in the following sections and potential solutions will be presented.

5 Group Membership Management

One of the main features of IP Multicast is how multicast group information is managed and how receivers join and leave multicast groups. As already explained, in IP this function is carried out by the IGMP protocol. By means of the group membership information, routers deliver multicast packets to the group members. For routers attached to broadcast networks (e.g. Ethernet), the required information is only if there are or not group members within the network, but not which and which IP or Ethernet addresses they have. In a shared media LAN every endstation sees every packet that is sent across the LAN. In ATM, a connection must terminate at the endstation in order for it to receive packets. Since ATM specifications do not provide the multicast address abstraction, it is necessary for an ATM attached source or and ingress edge device, to know which the receivers are and which ATM addresses they have, in order to explicitly establish a VC with itself as the root node and the recipients as the leaf nodes.

5.1 Multicast Address Resolution Server - MARS

The Multicast Address Resolution Server is, as explained in [Arm96], an extended analogon of the ATM ARP Server introduced in RFC 1577 [Lau94]. It is intended to be a registry, associating layer 3 multicast group identifiers with the ATM interfaces representing the group's members. MARS messages are used to distribute multicast group membership information between the MARS and endpoints wishing to take part in an IP multicast group. This section offers a general description of MARS. For more details on MARS behavior and its architecture see [Arm96]. Other documents related to MARS are [GKW97] and [Arm97a].

MARS Clusters

The MARS cluster is defined as the set of ATM interfaces choosing to participate in direct ATM connections to achieve multicasting of AAL_SDUs between themselves. This involves that if multicast communication is needed between nodes belonging to different clusters, an inter-cluster device must be used, or else some extensions to MARS are needed. Some proposals for these extensions are explained in Section 8.3.

Overview of MARS

As mentioned above, MARS is the multicast evolution of ATMARP. While the ARP Server keeps a table of (IP,ATM) address pairs for the IP endpoints within a LIS, MARS keeps tables like:

(layer 3 mc-address, ATM.1, ATM.2, ... , ATM.n)

IP nodes within the cluster, joining and leaving IP multicast groups, send appropriate messages to the MARS, indicating these changes. This way MARS always keeps an updated table of group membership information.

When a source needs to send IP multicast data through the ATM network, it requests the MARS to send the list of group members in the cluster, and their ATM addresses.In order to communicate the sources, the changes that occur in group membership, the MARS maintains a point-to-multipoint control VC, with all cluster members as leaf nodes. When a change in a group occurs, for example, a new node joins the group, the MARS sends one message on this VC which is received by all nodes in the cluster, irrespective of whether they are members of that group or not. The nodes which are not members and do not want to send data to the group, simply discard or ignore the message, but those that want to send data to the group, use this information to update their cache tables or to modify the point-to-multipoint VC, that they use to send multicast data for that group.

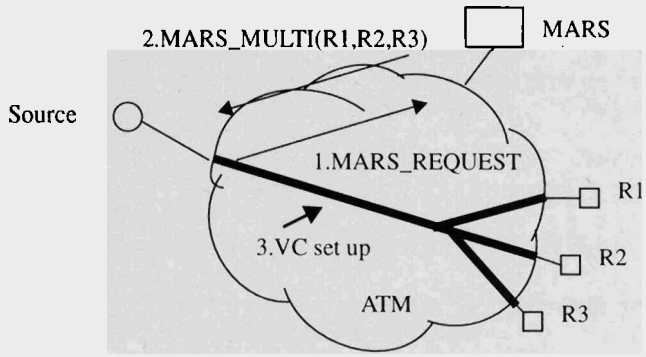


Figure 3: MARS Concept.

The Use of Multicast Servers with MARS

If multicast servers (MCS) are used, the behavior of MARS must be different, since it must provide the ATM address of the MCS instead of the list of ATM addresses of the group members. However, the MCS still needs to know this list of addresses in order to set up the appropriate point-to-multipoint VC. Therefore, some messages must be interchanged between the MARS and the MCS.

First of all, a MCS must register as it, in a similar manner as nodes wishing to be cluster members register within the MARS. As a consequence of using MCS, MARS must maintain another point-to-multipoint control VC called ServerControlVC. MARS adds to this VC all the MCSs in the cluster, so that, for those groups in which MCS is being used, membership change messages are not delivered in the ClusterControlVC, but on this new ServerControlVC. This way, these messages will reach only the MCS, thus shielding the sources of the group from the membership changes.

Support for IP Multicast Routers

Since MARS defines the propagation of group membership information within the cluster, extra devices are necessary to allow inter-cluster multicast communications. Multicast routers are expected to have the same IP/ATM interface that a multicast host would use. Within this interface, multicast routers will join and leave groups as any other ordinary cluster member. However, routers may belong to different clusters at the same time, thus providing routing between these.

MARS Messages

In this section, some of the messages utilized by MARS are explained, because in later sections they will be referenced. For more detail in these and other messages used in MARS, see [Arm96].

Several groups of messages can be identified, depending on their functionality:

- Messages for joining and leaving multicast groups: These messages will be sent by ATM hosts or routers which want to join or leave multicast groups.

MESSAGE	DIRECTION	DESCRIPTION
MARS_JOIN	HOST->MARS	Allows a Host to join a goup
MARS_LEAVE	HOST->MARS	Allows a Host to leave a group

Table 3: MARS join/leave messages

- Messages for sources sending multicast traffic: When a source/ingress device must send multicast packets, these messages are used in order to obtain the list of ATM addresses of the end points which are members of the multicast group.

MESSAGE	DIRECTION	DESCRIPTION
MARS_REQUEST	HOST->MARS	A Host requests the ATM address list of the group.
MARS_MULTI	MARS->HOST	Answer to the MARS_REQUEST message.
MARS_MIGRATE	MARS->HOST	Allows MARS to force cluster members to shift from VC mesh to MCS based forwarding tree in single operation.

Table 4: MARS messages for sources.

- Messages for Multicast Servers:

MESSAGE	DIRECTION	DESCRIPTION
MARS_MSERV MARS_UNSERV	MCS->MARS	Allow multicast servers to register and deregister themselves with the MARS.
MARS_SJOIN MARS_SLEAVE	MARS->MCS	Allows MARS to pass on group membership changes to multicast servers.

Table 5: MARS messages for multicast servers.

5.2 Other Models

Other approaches related to the problem of group management can be found in [Smi97], [Mil95] and [FMR97]. In [Smi97], a server (EARTH server) is used in a similar manner like MARS, and the concept

of a Multicast Logical IP Subnet (MLIS), 'spanning' the whole physical ATM network, is introduced. This way, multicast communications would be carried out at the MLIS level, and not at the LIS level, of the Classical IP model. Thereby intermediate routers between different LISs and the additional delay and overhead introduced by them would be avoided. This model tries to provide shortcut capabilities for IP multicast over ATM, which is achieved by having a unique MLIS. However, taking into account the scalability problems of this scheme, [Smi97] also includes the possibility of using multiple EARTH servers which partition the ATM cloud into service zones (clusters) and use a protocol among themselves to synchronize their caches. This is of course a solution very similar to MARS, with some extensions to support shortcuts.

In [Mil95] a simpler solution is proposed which uses IGMP for reporting group membership changes, like a usual IP network. There is no special device to map IP group addresses onto ATM addresses. Routers forward IP multicast packets to the member groups and other routers, known by IGMP and IDMR protocols. In broadcast networks like Ethernet, after a router requests group membership of the hosts in the network by means of IGMP Queries, only an IGMP Report is sent for each multicast group, due to the fact that after the first one has been sent by any of the members in the network, the others will receive this packet and will not send their own IGMP report. This host behavior was selected because, in broadcast networks, the router does not need to know the identity of all the members inside a particular network, but only whether there are members or not. In the solution proposed in [Mil95], IGMP packets are never forwarded by the routers attached to the ATM network. This requirement not to forward IGMP messages ensures that no host hears another's IGMP Host Membership Reports, thus every host will send them to the router in response to an IGMP Host Membership Query, and the router will get the addresses of all the members of the multicast group. The IP to address mapping will be in this case unicast IP address to ATM address, using ATM-ARP, NHRP, or any other method.

In [FMR97] another proposal is made for Intra-LIS IP multicast among routers, using PIM-SM (Protocol Independent Multicast- Sparse Mode).This model is actually a less complex solution for a part of the functionality provided by MARS. In this case, host-rooted point-to-multipoint multicast distribution VCs have not been considered. MARS allows point-to-multipoint VCs rooted at either a source or a multicast server (MCS). The approach taken here is to constrain complexity by focusing on PIM-SM (taking advantage of information available in explicit joins), and by allowing point-to-multipoint VCs to be rooted only at the routers. In summary, the method described in [FMR97] is designed for the router-to-router case, and takes advantage of the explicit-join mechanism inherent in PIM-SM to provide a simple mechanism for intra-LIS multicast between routers. By means of this explicit-join of PIM-SM, one router knows the IP addresses of other routers which have member hosts downstream, and using ATM-ARP or NHRP, it is able to find out their ATM address.

6 Basic VC Management for Data Traffic

In RSVP/IP, the reservation establishment is independent of the data transmission, because the path for the data transmission already exists before requesting and allocating resources. However, with ATM networks, the appropriate resource reservation must be done before data can be sent, that means an appropriate VC must be set up. Otherwise, there is no path for the data to be transmitted. Therefore routers at the ATM network edges need to manage the opening and closing of ATM connections, when RSVP reservations are made and released. The optimal scheme for connection setup and tear down will depend on:

- The cost of setting up a connection vs.
- The cost of keeping the connection open for future use by another flow.

Different flow to VC mapping strategies can be imagined. In the next sections some of them will be shown and their advantages and disadvantages analyzed.

6.1 PVCs vs. SVCs

Both, PVCs and SVCs can be used to provide data paths for the IP packets. The use of PVCs lets the ATM layer become a simple data link layer, similar to a leased line. PVCs are set up manually, and are expected to be long-lasting. With PVCs, there is no issue of when or how long it takes to set up VCs, since they are made in advance. However, the resources of the PVC are limited to what has been pre-allocated. The utilization of SVCs makes more flexible usage of the ATM network and allows more efficient flow mappings, but is on the other certainly more complex. If SVCs are used the cost of setting up a VC (not only with respect to time but also economically) turns out to be an important parameter, as well as the scalability characteristics of the mapping between RSVP flows and ATM VCs if the limited VC space is taken into account.It is obvious that an SVC scheme uses ATM's capabilities more efficiently. However, the drawback is the setup time.

One could certainly think of some more complex usage of both types of VCs which would establish PVCs between nodes inside the ATM network acting as stable aggregated traffic pipes while at the periphery of the ATM network SVCs would still be used. A similar kind of aggregation model will be investigated in Section 9.

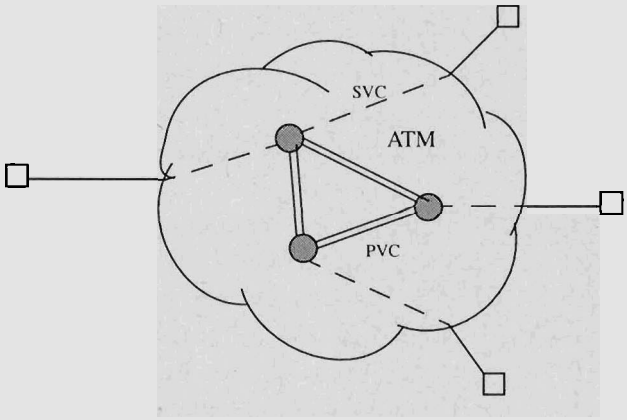


Figure 4: Mixed scheme. PVCs and SVCs.

6.2 Types of Traffic

It is helpful to identify which types of traffic shall be multicast over an ATM subnetwork and investigate the VC management issue along these types. We can distinguish three types of traffic:

1. Best-effort multicast data
2. QoS multicast data
3. RSVP control messages

Other traffic types could be included in this list, as e.g., control messages belonging to multicast routing protocols or to IGMP. Especially for control messages, it would be an advantage if they receive a better service than a simple best effort service.

6.3 Best-Effort Multicast Data

This type of data is the multicast data that would be transmitted in the IP architecture if RSVP is not used. In this case no QoS is requested.

The straightforward solution for VC Management would be to setup a point-to-multipoint VC to the members of the group in the same LIS/Cluster, when there is data to send, as proposed in [Arm96]. Inter-LIS/Cluster communications can be done through multicast routers (see also Figure 5).

Shortcuts could also be used, even though it does not seem strictly necessary. The utility of shortcuts is to avoid the increased delay due to processing in intermediate nodes (AAL5 reassembly and IP processing), and to off-load intermediate routers. However, since this traffic is best-effort, it is expected not to have critical delay requirements. The discussion of using shortcuts or not will be treated in more depth in section 8..

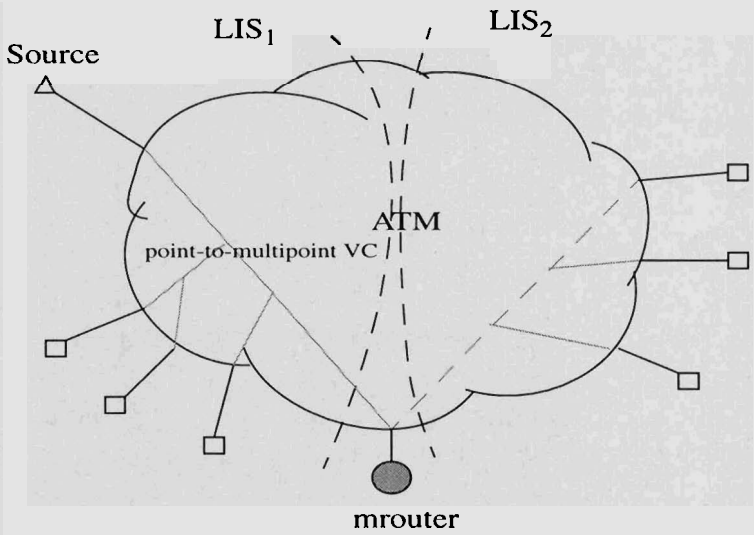


Figure 5: Inter-LIS multicast with multicast routers.

6.4 QoS Multicast Data

The VC management strategies for QoS multicast data is certainly one of the most important issues in implementing RSVP/IP Multicasting over ATM. The key points that should be analyzed in this case are:

- how different RSVP reservation styles influence the flow to VC mapping,
- when to setup and tear down a VC, and
- how to map RSVP flows onto VCs.

6.4.1 RSVP Styles and VCs

In [BCB+98] there is an analysis how the different RSVP reservation styles influence in the flow to VC mapping model. That means how the different reservation styles of RSVP should be translated into a number VCs.

Wildcard Filter (WF)

In this style of reservation, the receiver requests some quality of service for an unspecified set of sources of the group, i.e., all the sources that send to the group would share the same reservation. This style of reservation is intended for communication scenarios where only a limited subset of sources is expected to send data at the same time. For example, in the case of audioconference, there is usually only one speaker at anytime. If only one source is sending at the same time, it will use the whole reservation established (i.e. the whole bandwidth) for its data.

In this case, it seems that the use of one VC, point-to-point or point-to-multipoint depending on the number of receivers, fits well to this style of reservation. The QoS of the VC should be the QoS requested by the receiver/next hop, and all the traffic addressed to that receiver/group, regardless of the source, should be sent out on this VC.

However, it has to be realized that this solution does not support the sharing of reservations as well as it was planned in the RSVP design. It is only the best solution among the currently possible solutions. It

is not the best way to map wildcard filters in general. This fact is illustrated in figure 6, where there are two senders and two receivers, which request a WF reservation.

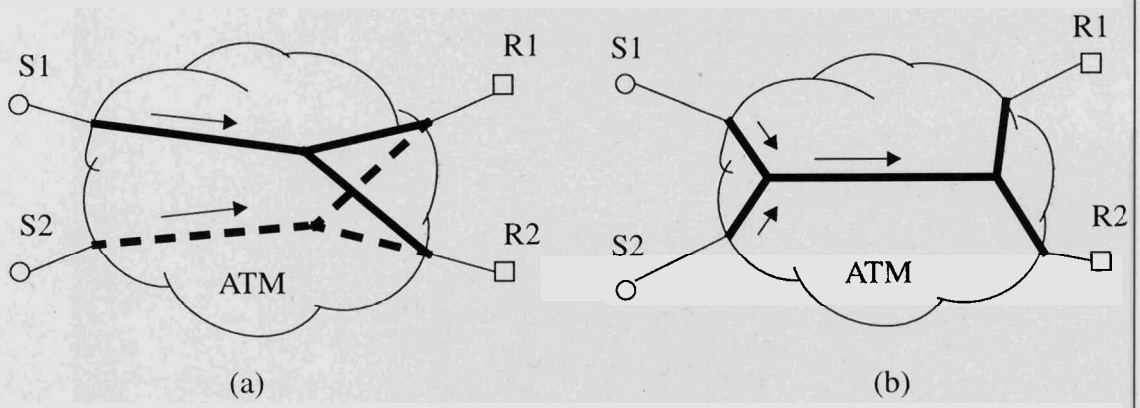


Figure 6: Wildcard Filter sharing.

In case (a), the currently possible model is shown. Each source establishes its point-to-multipoint VC towards R1 and R2. As can be seen, the reservation is not actually being shared among both sources, because each one of them has its own VC. In (b), the reservation would be really shared among all the sources. However, what is necessary for this approach, is a multipoint-to-multipoint VC, which is not (yet) supported by ATM. Anyway it must be noted that this would be the optimal mapping for wildcard filters.

Shared Explicit (SE)

This style is similar to WF, but in this case a specified set of sources is signalled. The same arguments as in the WF case are valid for this style with respect to using the same VC for traffic coming from the set of specified sources. However, there might be more sources for the group than those specified in the reservation. Traffic from these sources should be treated as best-effort traffic, using the VC management strategy for this kind of traffic, for example, setting up another VC to the receiver, sending best-effort traffic to a multicast router, and so on.

Fixed Filter (FF)

With this kind of reservation style the receiver/next hop requests a certain QoS for the traffic from a specified source. This leads to the straightforward solution of mapping such a flow onto one specifically tailored VC with adequate QoS. As in the case of SE, other VCs may be necessary for traffic coming from other sources (best-effort or QoS), and even for the same source and other receivers/next hops, if they do not request quality of service, or the heterogeneity model being used permits the use of several VCs of different QoS (see also section 12).

6.4.2 VC Initiation

When mapping the RSVP mechanisms onto ATM there is the obvious question about where and when to setup a VC for a RSVP flow. Two approaches are distinguished in [SWS97]: the subnet-sender approach and the subnet-receiver approach.

Subnet-Sender Approach

Even though RSVP is receiver-oriented, since the receiver requests a reservation by means of a RESV message, the actual allocation of resources for the downstream link takes place at the subnet sender. In

an RSVP over ATM implementation, this means the sender/previous hop should set up the required VCs for the communication.

With respect to the timing of the VC setup, in principle two different options exist. The earliest possible point in time would be when the first PATH message is being received. However this choice seems to be too hasty since, though there are PATH messages, it is possible that no reservation requests are issued and therefore, no specifically tailored QoS VC would be necessary. In particular, for multicast communications too many resources would be wasted if no reservation is requested and a point-to-multipoint QoS VC to all receivers is established just because a PATH message was received. Moreover, it is by no means for sure that the receiver requests what was specified in the PATH message and thus a wrongly dimensioned QoS VC could be setup.

The other and more reasonable option is to setup the QoS VC when a RESV message is received, and therefore a reservation is actually requested. This leads to reserving resources only when necessary but incurs additional delay for the VC setup before the RESV message can be passed on upstream. From the point of view of VC management for multicast, this is a convenient choice since it is simple for the virtual source to setup point-to-multipoint VC(s), and add more leafs nodes as their reservation requests arrive. Furthermore, this option allows the ingress edge devices to use different VC management strategies, and all of them could interoperate, since the decision of setting up a VC or not is local. Its main problem, however, is the scalability. This model would not support large multicast groups as the ingress edge device might become overloaded due to the load of setting up and tearing down branches of a point-to-multipoint VC for a large and dynamic multicast group. Nevertheless it is the only choice if UNI 4.0 LIJ is not available.

Subnet-Receiver Approach

In this approach it is the receiving/egress edge device who sets up the QoS VC. If only UNI 3.0/3.1 is available, there is one way to achieve this: the receiver/next hop sets a VC up, requesting resources only in the reverse path. This would work for the unicast case, but not for the multicast case. So, the subnet-receiver approach is very limited when only UNI 3.0/3.1 signalling is provided. Here it seems more attractive to use the subnet-sender approach.

However, if UNI 4.0 is available, the LIJ mechanism may be helpful for the multicast case of the subnet-receiver approach, since the processing load of the virtual source, related to the VC setup process, can be distributed among the receivers. This solution would have a higher scalability than the subnet-sender one with its centralized VC management. However, if more complex VC management algorithms are desired, which allow some degree of VC space saving and heterogeneity support, it may be difficult to integrate them with the LIJ facility. These algorithms need centralized knowledge and would therefore most easily be executed by the ingress edge devices. This is due to the fact that the decision of adding a receiver to a certain already existing VC or to create a new one for it must eventually be taken at the ingress edge device. Extra signalling would be necessary to integrate the centralized VC management algorithms with the distributed philosophy of LIJ.

6.4.3 VC Teardown

RSVP has its own flow teardown mechanisms (tear down messages or timeouts), thus inactivity timers as they are used in the Classical IP model and proposed in [Lau94] are no longer needed.

In case that there is a one-to-one mapping between RSVP flows and VCs, a VC should be torn down, when it corresponding RSVP reservation is deleted or timed out. However, if more complex VC management strategies are being carried out, the teardown of a VC would depend on whether there are still flows using the VC or not. In any case, the VC teardown should be governed by the RSVP flow terminations, and not by RSVP-independent timers. Otherwise, valid VC(s) with QoS support could be torn down unexpectedly [BCB+98].

6.4.4 Flow to VC Mapping

There are four categories with regard to this issue:

1. **"1 VC per flow"**: This case is suitable for unicast and homogeneous multicast communications. The main advantage of this solution is that the traffic control and scheduling capabilities of the ATM network can be directly utilized. The drawbacks of this scheme arise when receivers request different qualities of service. In this situation the following problems can occur [BCB+98]:
   - A user making a small or no reservation at all would get a "free ride" across the ATM network on any receiver making a (larger) reservation.
   - A user might not be able to join the QoS VC because of lack of local resources to process the high quality data flow. However, the receiver could still want to receive data on a best effort basis. With only one VC per flow this would not be possible i we assume that always the highest QoS in a session is taken for the setup of a VC.
   - Resources would be wasted and blocking probability could be higher than necessary

   A "more heterogeneous" model is needed to deal with this problems.
2. **"n VCs per flow"**: This is the option that allows more heterogeneity and that, generally speaking, consumes more ATM resources (with respect to the number of VCs). The number of VCs per flow, will depend on the degree of heterogeneity that shall be allowed. The heterogeneity problem is treated in more detail in section 12 and a flexible algorithmic framework for managing the heterogeneity support is proposed.
3. **"1 VC per n flows"**: That is an aggregation model without heterogeneity support. This scheme involves much more complexity, especially for the multicast case, since it requires that the different groups have the same virtual receivers. Otherwise, virtual receivers/egress edge devices must be prepared to receive data which is not addressed to them thereby certainly wasting bandwidth.
4. **"n VCs per m flows"**: Aggregation model with heterogeneity support. This is certainly the most complex model of the four, and its use may be only justified in the core of the network, where there is enough potential for multiplexing traffic of different groups and QoS levels in order to take advantage of this model.

Aggregation models will be discussed in section 9.

7 VC Management for RSVP Control Messages

The two most important RSVP control messages are PATH and RESV. While PATH messages can be sent to a multicast address, RESV messages are sent to a unicast address, the previous hop address. How to manage VCs for RSVP messages depends on several factors [BCB+98]:

- Number of additional VCs needed for RSVP signalling.
- Degree of multiplexing on the RSVP control VCs.
- Latency in dynamically setting up new RSVP signalling VCs.
- Complexity of implementation.

Different options to assign VCs for RSVP signalling messages are proposed in [BCB+98]:

- Use the same VC(s) as for the data.
- Use a single VC per session.
- Use a single point-to-multipoint VC multiplexed among sessions.
- Use multiple point-to-point VCs multiplexed among sessions.

7.1 Same VC for Data and Control Traffic

RSVP signalling messages are sent on the same VC as the data traffic. The main advantages of this scheme are:

- No additional VCs are needed beyond what is needed for the data traffic.
- There is no ATM signalling latency for PATH messages.
- There is also no multiplexing with control messages from other RSVP sessions, therefore the complexity is very low.

Its disadvantages are:

- When data traffic is nonconforming, RSVP messages may be dropped ("fate sharing" with data). Even though RSVP messages are resilient to some level of dropping, this may lead to repeated tearing down and reestablishing QoS VCs.
- If the communication is multicast, PATH messages will be able to use the point-to-multipoint VC that was setup for data transmission, but in the upstream direction, the RESV messages cannot be sent using that VC, since point-to-multipoint VCs are unidirectional. In this case, RESV messages will have to use another VC.

7.2 Single RSVP VC per RSVP Reservation

This option means using a separate VC for RSVP signalling traffic, in parallel with the QoS VC for the data. In this case the number of VCs needed is twice the minimum required, however there is still no multiplexing between sessions and therefore the implementation complexity is still low. Once a data VC is created, a separate signalling VC is also created.

In the case of multicast this solution again shows deficiencies. Since the RESV messages have no way to be sent in the reverse direction of the PATH messages they still need a special VC.

7.3 Multiplexed Point-to-Multipoint RSVP VCs

In this scheme each ingress edge device uses a point-to-multipoint RSVP signalling VC for each unique set of egress edge devices. With this approach the number of VCs needed is much lower, since it allows multiplexing among control traffic that shares the same ingress edge device and the same set of egress edge devices. The likelihood for a multiplexing gain is greater if the number of edge devices surrounding the ATM network is not too large, since otherwise the probability that two different groups have exactly the same egress routers can become very low.

A problem of this scheme is due to dynamic membership in IP multicast groups, which might lead to a different set of egress edge devices for a certain multicast group. The RSVP control traffic must now use a different point-to-multipoint VC to be transferred.

Nevertheless, the number of VCs used will be lower than in the one signalling VC per reservation approach. The exact savings depend on the patterns of the traffic and the topology of the ATM network.

7.4 Multiplexed Point-to-Point RSVP VC's

This approach uses one point-to-point VC from each ingress device to each of the egress devices. By using bidirectional point-to-point VCs it is now possible that PATH and RESV messages follow the same path through the ATM network. This allows for a certain saving of VC space. While the scheme allows for multiplexing between sessions, it requires the same traffic to be sent on each of several VC's.

The number of VC's will be at most n(n-1)/2, where n is the number of edge devices.

7.5 Alternative Scheme

The options presented above all share the idea to let RSVP control messages follow approximately the same path as the data, reaching their next/previous hops as in an IP network. An alternative scheme, as

proposed in [SCSW97], uses a centralized server for receiving and sending RSVP control messages over the ATM network. Moreover, this server also acts as a multicast address resolution server using EARTH [Smi97]. This way, regardless of the next/previous hop to which a RSVP message must be sent, it will always be sent to this server, the so-called Multicast Integration Server(MIS). In fact, this scheme utilizes a completely new signalling between edge-devices and the MIS, for establishing reservations at the layer 2 level, which unifies the multicast address resolution messages with the QoS requests messages.

The main problem of this model is its scalability, since the server, following the model proposed in [SCSW97], must keep state of all RSVP sessions, senders and receivers, and carry out RSVP processing for all these sessions. Moreover, this RSVP processing is also necessary in the ingress edge-devices, thus duplicating not only the RSVP State Information but also the RSVP processing.

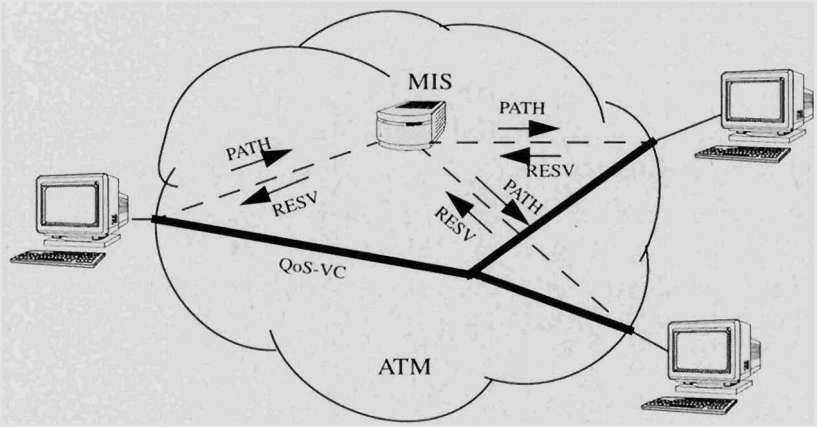


Figure 7: Multicast Integration Server Architecture.

7.6 Observations and Own Scheme

Probably none of the above solutions is optimal for every environment. An approach that send RSVP messages through the same VCs as the data packets tries to imitate the model followed in legacy IP networks. In these networks RSVP messages are sent like normal data packets because they cannot be treated in any other way. However ATM networks allow for isolation between data and signalling by using different VCs. On the other hand, using one signalling VC for each RSVP session may be too resource-intensive. Schemes for saving signalling VCs by multiplexing control traffic between sessions may be too complex to implement. Moreover, it is not always obvious that the effort is worthwhile the potential for control traffic multiplexing. The multiplexing approaches seem to make most sense in the core of the network.

If the problem is restricted onto the unicast case it is much easier to solve. In this case, RSVP messages could simply be sent on the point-to-point VC set up towards the receiver/next hop, enabling some bandwidth if necessary in the reverse direction of the VC, for the upstream control traffic, as e.g., RESV messages.

The main difficulties arise with the multicast case, where some nodes may be reachable through best-effort VCs, and others through QoS VCs. Of course, none should stop receiving RSVP messages. Due to the fact that in an IP Multicast over ATM implementation there will probably always be an instance for address mapping, as e.g., MARS or EARTH. This instance could be used for RSVP messages delivery. For example, MARS could be used as a Multicast Server for RSVP messages, thereby making possibly use of the VC connections that it maintains already with all the cluster clients, or at least utilizing its knowledge about the IP multicast group-ATM addresses relation. This model can be viewed as a simplification of the model proposed in [SCSW97]. The simplification lies in not making any RSVP processing in the extended MARS but only forwarding RSVP messages to the appropriate destinations.

This way, all RSVP messages should be sent always to the server, which will only forward them to the appropriate destinations. Therefore, this server becomes a RSVP message multicast server in addition to being a multicast address resolution server. With these two properties some processing with the RSVP messages could be done in this server, in order to simplify things as for example shortcuts. The receiver could send RESV messages directly to the source, if the server does not modify the previous hop object of PATH messages. Alternatively they could be sent to the server, if it included its address in the previous hop object of PATH messages. In this case, the server could include some objects into the RESV messages as for the ATM address of the virtual receiver/next hop, in order to simplify a shortcut from the ingress to the final egress edge device.

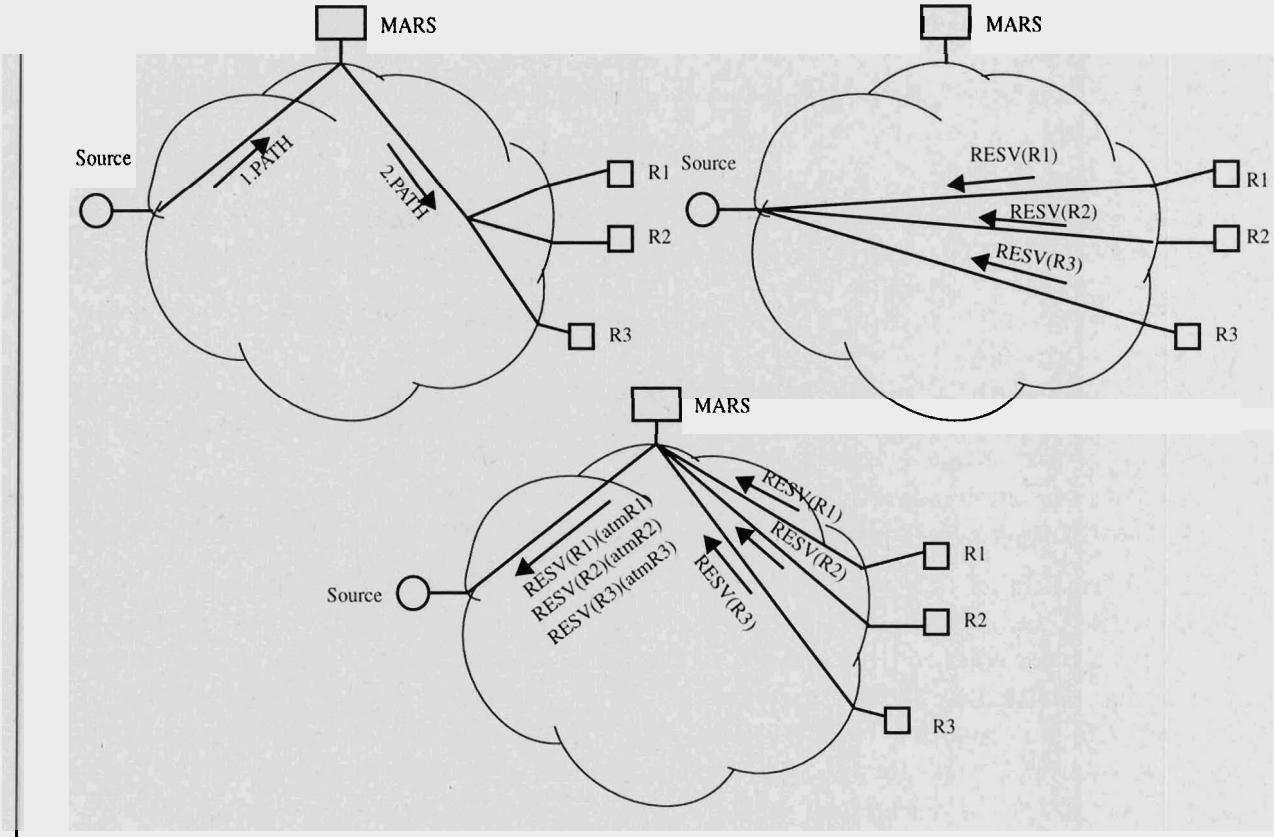


Figure 8: MARS as RSVP Message Server.

RSVP messages could be sent to the MARS as they are, or encapsulated in new MARS messages. The advantage of encapsulating would be the possibility to let MARS clients include some more information, as e.g., from which MARS client a message was sent. This information would be useful, because if the Cluster Control VC is used to deliver RSVP messages all MARS clients will receive them, including the client which sent the message. With this information, the MARS client process will only pass the RSVP message to the RSVP daemon, if the message has not been sent by that cluster client itself, respectively if the cluster client is a member of the group the RSVP message is addressed to.

A possible improvement [Mil95] of all the strategies for delivering RSVP control messages over the ATM network is, that once a QoS VC is set up for a flow there is no need to refresh RSVP state along that segment of the flow. RSVP messages would only need to be sent if the QoS parameters of the flow were to be changed, or if the flow were torn down. This would eliminate the problem of RSVP messages implosions at sources of large distribution trees, except during initial setup, but would produce hard state at the edge-devices of the ATM network.

8 Shortcuts vs. Hop-by-Hop

In figure 9, PATH messages are delivered hop-by-hop to the final egress edge device. If the intermediate routers do standard RSVP processing then they will each "sign" the PATH messages as previous hops. According to the thus established PATH state, RESV messages will be transported back hop-by-hop in the reverse direction setting up a concatenation of VCs between the routers connecting different LISes/ MARS clusters.

Let us suppose now that all intermediate edge device forward PATH messages without modifying the previous hop object. Then the RESV message of the final egress edge device would be sent straight to the ingress edge device and from there a shortcut VC to the egress edge device could be established. This is only an example of a modification in the router behavior that would allow shortcuts for RSVP-signalled data flows.

Before considering approaches to support shortcuts, the general merits and drawbacks of this technique should be recalled:

- Advantages are
  - that lower delays can be achieved due to maximizing the switched path,
  - that ATM's PNNI and its QoS routing capabilities can be utilized over the whole ATM subnetwork and not just a LIS/Cluster, and
  - that routers are off-loaded, thereby avoiding them to become bottlenecks.
- Disadvantages are
  - that the virtual source to the ATM network might become overloaded due to a so-called "VC implosion" problem: if the ATM network becomes large, then there will be too many reservations to manage and too many RESV messages to process,
  - that shortcutting reduces the potential for aggregation of flows at the network layer, since less flows will share the same ingress edge-device the closer the ingress edge device is located to the actual sources.

Hence, shortcutting is not good by virtue. We must thus determine when establishing a shortcut is really worthwhile. A prerequisite to establishing a shortcut is that the amount of data and the lifetime of the flow are large enough to justify the effort. The decision to establish a shortcut should be based on the load of the intermediate routers as well. If those are already very loaded, then a shortcut might be the only possibility to establish data flow across the ATM network. The VC management scheme to support shortcuts should thus take into account state parameters of the ingress edge device and all the intermediate routers of the hop-by-hop path. In general, the following basic rules can be given:

- If the source is not able to support shortcuts, then, in case of an RSVP session, the next reservations requests should be processed and merged by the intermediate routers.
- If the source supports shortcut and any of the routers is overloaded then a shortcut should be established.
- If the source supports shortcut and none of the routers is overloaded, both, shortcut and hop-by-hop are principally possible.
- If the source and the routers are overloaded, none of both schemes will work.

In the next sections we will analyze some existing approaches and propose new ones for shortcutting. This investigation will be made along the criteria of best-effort vs. QoS and uni- vs. multicast traffic.

8.1 Shortcut for Best-Effort Unicast Communications

In the best-effort case, shortcuts should not be necessary, as this kind of traffic has no strict timing requirements. Anyway, if a shortcut is desired, the standard way for unicast to establish it, is using NHRP [LKPC98]. NHRP is the IETF's extension of ATMARP for getting ATM addresses of IP nodes outside the local LIS. NHRP messages are encapsulated in IP packets and sent using IP routing. If one

of the nodes in the IP path has in its cache the ATM address of the IP address requested, a new message is sent back to the source node with this information.

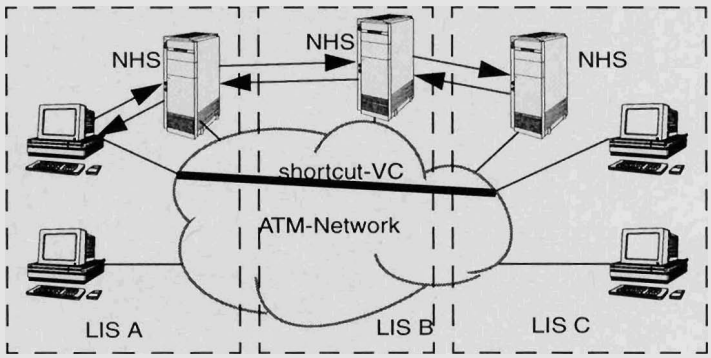


Figure 9: NHRP Protocol.

8.2 Shortcut for QoS Unicast Communications

In this case, both, the subnet-receiver and the subnet-sender, could setup the shortcut VC since point-to-point VCs are bidirectional and asymetric. However, a subnet-sender approach is probably more reasonable, since the ingress edge device certainly knows best about his current load due to processing of shortcuts. So, we only regard the subnet-sender-initiated shortcuts here. If shortcut is desired for this case and a reservation request has actually been issued by the receiver, then the RESV messages should only be processed by the ingress edge device, and not by any of the intermediate routers. There are different approaches to achieve this:

1. The virtual source could include into the PATH message an object that contains its ATM address, so that the subnet-receiver who requests a reservation can send its RESV message right to it, as suggested in [BFGK96]. That would, of course, mean modifying the RSVP protocol by including this new object and adequate processing for it.

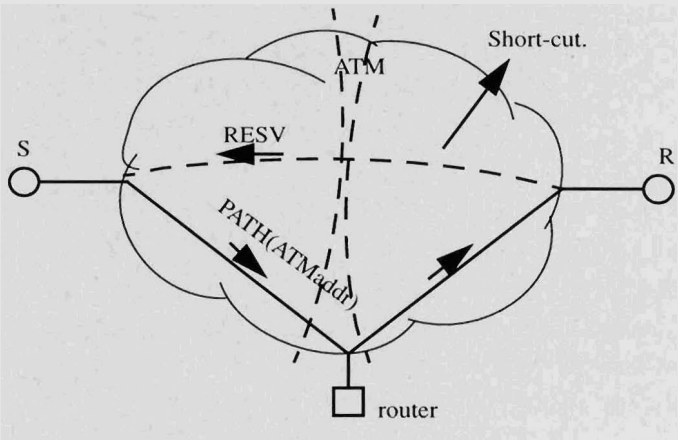


Figure 10: PATH message with source ATM address.

2. By means of some indication the receiver could tell the routers not to process the RESV message but forward it to the ingress edge device. This way the subnet-source would see the final egress edge

device as next hop and could establish a shortcut to it.

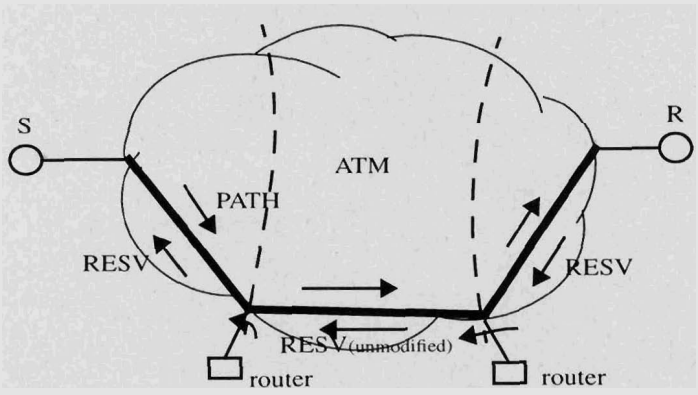


Figure 11: Forwarding "unmodified" RESV message upstream.

Another question is: how does the subnet-source know the ATM address of the subnet-receiver? Two possible solutions are:

1. Use NHRP to get that ATM address. It may take some time, however, since it is expected that the lifetime of the connection be much longer, that should be acceptable.

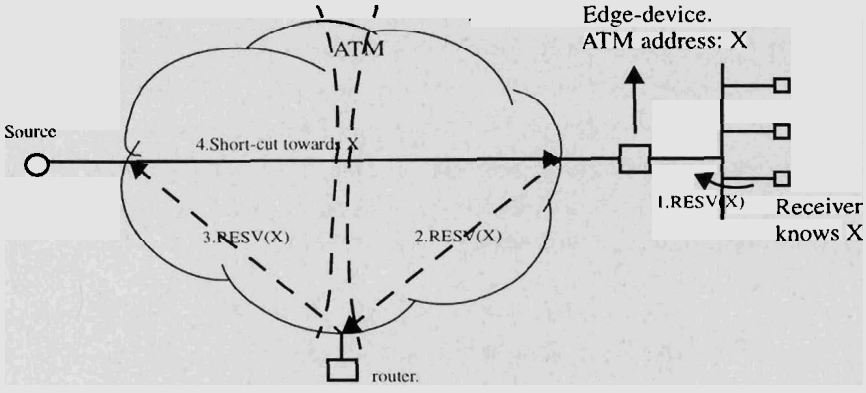


Figure 12: Non-RSVP capable edge device.

2. Include a new object in the RESV message which carries the ATM address of the subnet-receiver to which the shortcut should be established [BFGK96]. This solution would also permit a non-RSVP capable egress edge device. The next RSVP-capable hop would be connected to this edge device. It sends its RESV message including the ATM address of the egress edge device in the new object. This way, the source knows the ATM destination of the QoS VC.

8.3 Shortcut for Best-Effort Multicast Communications - Extending MARS

In this case there are no RSVP messages or, at least, there are no reservations yet. Let us suppose that we are using MARS in conjunction with the VC-mesh approach. Then, for inter-cluster communications, one or several multicast routers will be used as illustrated in figure 13.

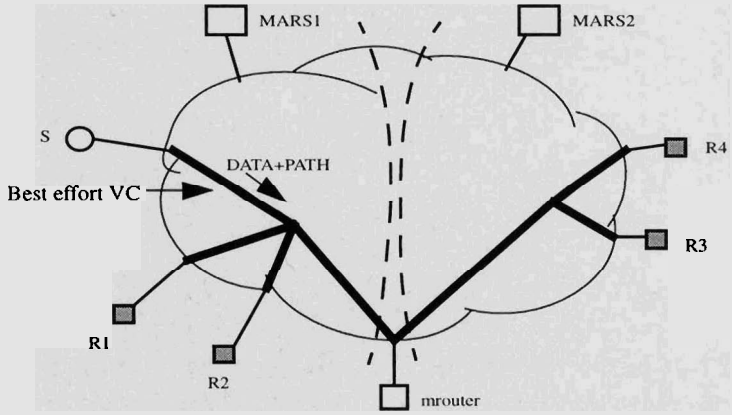


Figure 13: Multicast with multicast router (hop-by-hop).

If we desire to establish a multicast shortcut, MARS needs to be extended in a similar way as ATMARP had to be extended to NHRP in order to support shortcuts for the unicast case. There are however some serious problems when trying to establish shortcuts for the multicast case:

1. How does a source get to know the ATM addresses of receivers outside its own cluster and how should it keep track of membership changes outside the cluster. MARS should be modified to provide this information to the source. Therefore a form of coordination between MARSes is probably necessary.
2. It is possible that the number of receivers or members of a group exceeds the greatest point-to-multipoint VC a source or the ATM network is able to set up, as explained in [Arm97a]. In this case, either the number of group members must be limited, or a mixed scheme of using shortcuts and multicast routers could be designed for this situation. However, both options have their drawbacks. To limit the number of members in a group could certainly be very restricting for future large-scale multicast applications. The mixed scheme of using shortcut and hop-by-hop requires a complicated management due to the fact some receivers receive data through the shortcut VC while others get them from the hop-by-hop path. This would also results in different QoS for those two kinds of receivers.

An alternative to alleviate at least the second problem would be to have some kind of multicast servers in order to aid the source, in case no more leaf nodes can be added to the point-to-multipoint VC. This scheme would result in a cascade of sources. Usually, a very small number of cascaded sources will suffice. In most cases, no more than one of these devices should be needed in any multicast communica-

tion. This is valid if the number of group members is less than twice the maximum number of nodes allowed in a point-to-multipoint VC. The case of one auxiliary source is depicted in figure 14.

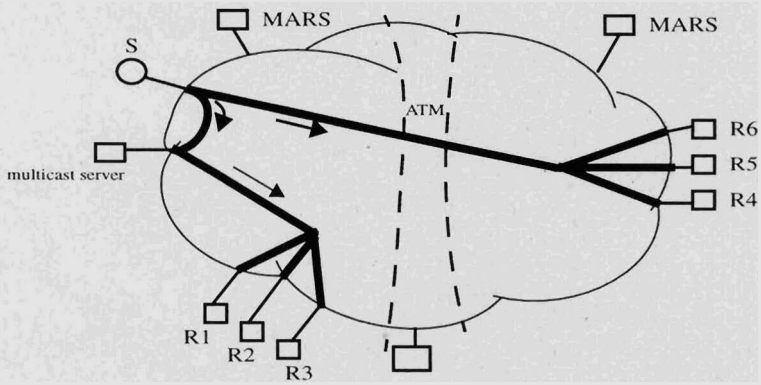


Figure 14: Cascaded Sources.

No IP processing is needed in the auxiliary multicast servers, because its only function is to extend the point-to-multipoint VC of the source. Thus, instead of a multicast server at the IP level it could be a an device which only takes incoming cells and forwards them on a point-to-multipoint VC. It is not even necessary to do AAL processing within this node.

Extending MARS

In this section we propose extensions to MARS in order to allow shortcuts in the multicast case. Another proposal to extend MARS with shortcut capabilities, the VENUS architecture, is explained in [Arm97b].

One of the problems that must be treated is how the source gets to know the ATM addresses of the receivers outside its own cluster. First of all, MARS_REQUEST messages should be modified in order to let the source specify if shortcut instead of normal hop-by-hop routing is desired. This could be achieved by adding a new TLV(Type-Length-Value) field in the MARS_REQUEST message, which indicates to MARS that the source would like to establish a multicast shortcut.

In turn, the MARS should then answer in its MARS_MULTI message with all the ATM addresses of the ATM subnet-receivers of the group. To be able to do that, a scheme that allows MARS to know the addresses of receivers registered at other MARSes is needed. Therefore some messages between MARSes from different clusters is necessary. In the unicast case with NHRP, a request message is sent inside an IP packet, being forwarded to different NHSes(NHRP Servers) until one of them knows the ATM address requested. In case of MARS, this request message should be addressed to others MARSes. However, the requesting MARS does not know which other MARSes have members of the group, so two approaches are possible:

1. Send the request message, one by one, to all the MARSes of the network. This certainly shows scalability problems if the number of MARSes is becoming large. Furthermore, the requesting MARS would need to know the ATM addresses of all MARSes in the ATM network.
2. MARSes should be also IP nodes. This way, they could join a specific IP multicast group dedicated to the inter-MARS communication. Thus, requests for group members of specific IP multicast groups would be received by all MARSes, and the ones that have members of that group in their cluster could answer with a list of the group members and their ATM addresses. The answer could be sent as an IP packet back to the source IP address of the multicast packet received, i.e., the requesting MARS, or, alternatively, to the multicast group of all MARSes. The second option will result in more traffic than the first one in general and seems therefore inferior, but would have the advantage that group membership information could be cached by MARSes even if they have not

yet requested it.

3. MARSes are in higher level cluster with one dedicated MARS to which requests are sent and which sends answers back.

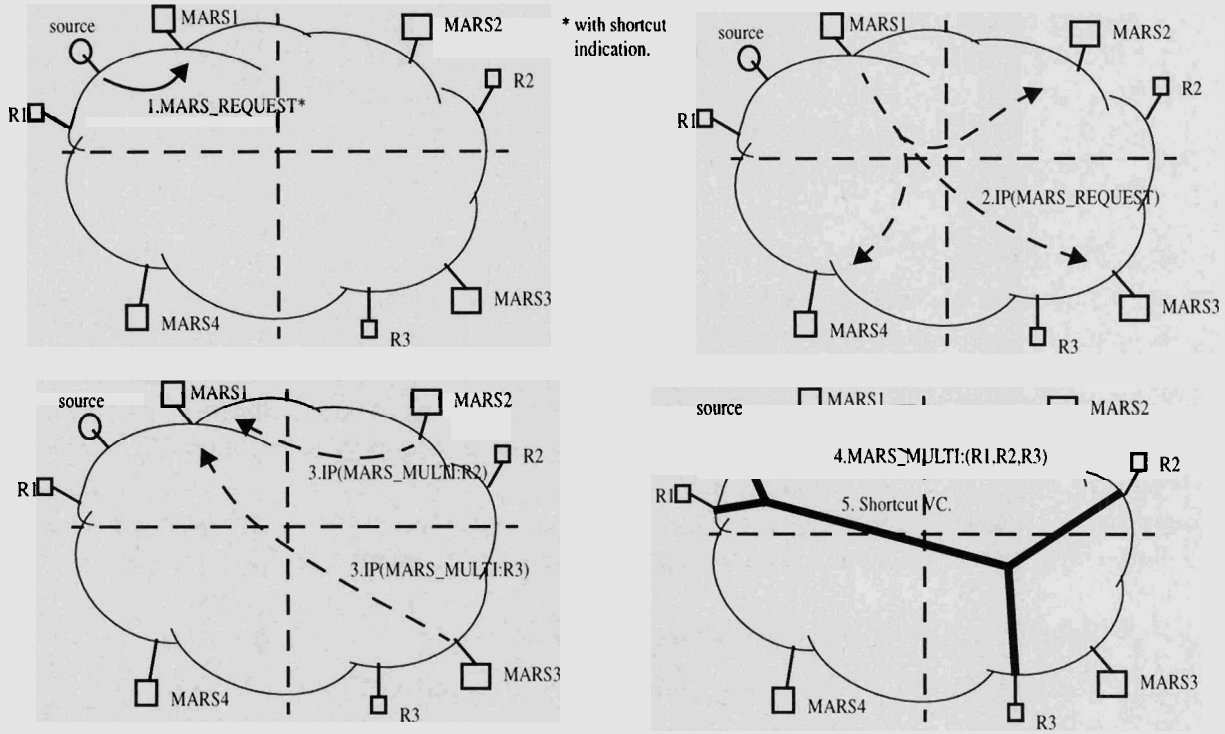


Figure 15: MARS extensions.

With one of theses approaches, it is now possible for the MARS of the cluster in which the source is located to get to know all the receivers that currently belong to the group, as illustrated for the second approach in figure 15. However, IP multicast groups are dynamic, thus membership changes in other clusters must be tracked in some way.

One possible approach is that each of the MARSes adds the requesting MARS to its Control Cluster VC, so that if changes in group membership occur, the requesting MARS is aware of them. This solution is certainly not scalable since the requesting MARS must be added to all the Control Cluster VCs and must process the information received on all of them. It should be noted that it would even need to be added to the Control Cluster VC of clusters that have no members for that group, because they might appear anytime.

A more scalable solution would be to indicate group changes to other MARSes by encapsulating these messages in IP packets. These packets should only be sent in case that, for the group that has changed, a requesting MARS message has been received, i.e., there is a source somewhere in the ATM network using multicast shortcut for that group. This way group changes would be delivered by IP packets between different MARSes. The question is how a MARS knows where the shortcut sources are and whether they are still active. A MARS_REQUEST message with a shortcut indication from a source to its local MARS can be seen as a way to register as a "shortcut source" within this cluster. Similarly, the request message sent to the IP multicast group of MARSes can be a way to register within all other MARSes as "MARS with shortcut sources for that group". With this information each MARS whose cluster have changes for that group could notify them to the MARSes which have a shortcut source for that group. When a source decides to finish its connection, a message should be sent to the local MARS to delete this source as a "shortcut source". This could be done by introducing a new type of message in the MARS protocol, or simply using a MARS_REQUEST message with a TLV field indicating that the source does not need shortcuts any more. A similar message should be sent by the

MARS to the IP multicast group of MARSes in order to be deleted as "MARS with shortcut sources" for a particular group, if it has no "shortcut sources" for that group any more.

The main problem of this solution and the one proposed in [Arm97b] is scalability. The same limitations about the MARS cluster size, as explained in [Arm97a], can be applied to the presented MARS extension. Therefore, as [Arm97b] concludes, in essence, there is little difference between a VENUS domain, or in our case, the set of clusters participating in shortcuts, and a large MARS cluster. This may lead to the conclusion that the problem of multiple-LIS multicasting could also be solved by simply creating a single large MARS cluster as proposed in [Smi97].

UNI 4.0 LIJ Facility

If UNI 4.0 Leaf Initiated Join is available, short-cut for multicast best-effort communications can be simplified to some extent. For best-effort multicast communications the LIJ facility is useful, since now the source does no longer need to know the ATM addresses of the members of the group. With LIJ, it is the receiver who joins the point-to-multipoint VC(s) if it desires to receive best-effort multicast data over a shortcut VC. Therefore, the problem now is to find out the identifiers (GCIDs) of the existing point-to-multipoint VCs for that group. MARS is currently designed to provide the ATM addresses of members of a group. Some extensions or a different protocol would be necessary to provide a receiver which wants to use LIJ with the GCIDs of the point-to-multipoint VCs of the group.

8.4 Shortcut for Multicast QoS Communications

One of the problems in implementing shortcut in the best-effort case is that, because of the IP multicast model, the source neither knows nor is informed about which members there are in the group. Therefore, a procedure for the source to get to know this information is required. MARS is an implementation of such a procedure, but its coverage is limited to the cluster. The problem, thus, was extending MARS so that it works also for inter-cluster communications in a scalable manner even if dynamic membership is taken into account.

For the QoS case, where RSVP signalling is used, establishing shortcuts becomes actually easier than in the best-effort case. Because when a receiver requests a reservation sending a RESV message to the previous hop, it is explicitly notifying its identity (by means of its IP address at least, if no extensions are being made). Therefore, the source knows which are the receivers of the group by means of the RESV messages. No additional mechanisms are necessary for finding the identity of the virtual destinations.

If shortcut is being used for best-effort multicast data and thus for PATH messages, the previous hop of the PATH message where the receiver has to send its RESV message to, is the ingress edge device itself. If hop-by-hop is being used, the PATH message could be modified to contain an indication for the multicast routers to not modify the previous hop object of the PATH message. Hence, the receiver would send its RESV messages straight to the source. With both methods, the ingress edge would know the IP addresses of the receivers to which a shortcut VC shall be established. However, what it needs to know is the ATM addresses of the subnet-receivers, the leaves of the shortcut point-to-multipoint VC. It is the same problem as in the unicast QoS case and thus the same approaches are possibly applicable. The first option is to use NHRP to discover the ATM addresses of receivers outside the cluster. Besides the advantage of using a standardized mechanism, this has the following drawbacks:

1. The delay until a QoS VC is established could be too long, especially if the multicast group becomes larger and more dynamic.
2. There is a problem with non-RSVP capable egress edge devices, because for those next hop and ATM network egress will not be the same machine.

A more efficient solution would be to include a new object into the RESV message, which contains the ATM address to which a shortcut should be established, as described for the unicast case. This way every member of the group which wanted to receive data with QoS could send a RESV message containing additionally the ATM egress point. With this information, the ingress edge device could add it to an existing shortcut point-to-multipoint VC, or could create a new shortcut VC, or could take any other decision depending on the VC management strategy being implemented.

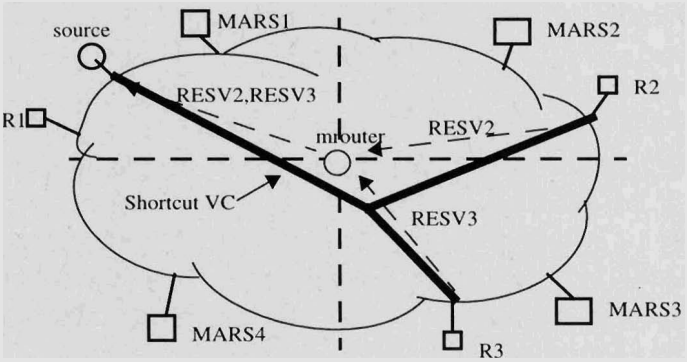


Figure 16: Forwarding RESV messages.

UNI 4.0 LIJ Facility

On first glance, LIJ seems to be a good match with the receiver-oriented philosophy of RSVP. However, when a receiver requests a reservation, a RESV message is sent upstream, but the actual reservations are carried out in the downstream interfaces. Therefore, in the ATM context, it seems reasonable that the subnet-sender should be the one who sets up the branch of the point-to-multipoint VC.

With LIJ, however, it is possible that the branch is setup by the subnet-receiver when the RESV message arrives at the egress edge device. If LIJ is used, then a useful modification of RSVP would be to include the GCID of the shortcut point-to-multipoint VC into the PATH messages sent by the ingress edge device in order to be able to join that VC by the egress edge device. As an advantage of using LIJ the load in the ingress edge device would be lowered and thus a better scalability with the number of receivers could be achieved.

If a VC management strategy that permits the use of multiple VCs for a single RSVP session, in order to e.g. support some degree of heterogeneity, a receiver might either be offered a choice of different VCs which he could join or the source decides according to global criteria which VC is appropriate for a receiver to join and just sends one GCID in the PATH to the egress edge device. Here it becomes obvious that LIJ is not such an elegant solution as one would expect at first. While a choice of different VCs to join does not optimize the VC management according to global criteria, the other option of deciding which VC is appropriate for a receiver at the virtual source before a RESV message has been received is very restricting. The centralized nature of VC management strategies just does not fit very well to the decentralized concept of LIJ.

8.5 Location of the Shortcut Decision

The fact whether shortcuts are used for best-effort traffic or not, affects the way RSVP control messages are delivered over the ATM network. This, in turn, influences the way shortcuts for RSVP flows can be established and which instance decides about the establishment:

- If shortcut is being used for the best effort traffic, establishing shortcut for the QoS case is straightforward, since the RSVP PATH messages travel from the ingress edge device straight to the egress

edge device, without any intermediate IP nodes. Therefore, the previous hop is the ingress edge device. In fact, in this case there is no other choice than using shortcut for the QoS case.

- If hop-by-hop is being used for the best effort case, using shortcut for QoS traffic might be an initiative of:
  - the ingress edge device,
  - the egress edge device,
  - the intermediate routers.
- If the source decides using shortcut, one of the methods explained above methods, as e.g. modifying the PATH message, should be utilized. Some changes are also needed in the intermediate routers, in order to avoid them modifying the previous hop object of the PATH message.
- If the receiver wants to use shortcut, then the RESV messages could be sent right to the source or ingress device, regardless of the previous hop of the PATH message. A possible way to be able to do this is to include the ATM or IP address of the ingress edge device into the PATH message.

  If hop-by-hop is being utilized for best-effort traffic and receiver-initiated shortcuts for the QoS traffic are desired, it requires some coordination between the receiver and the multicast router(s) the best effort traffic goes through. This is needed in order to delete the nodes that are using shortcut VCs from the best-effort VC.

  In case QoS traffic would be delivered hop-by-hop, deleting the receiver from the best-effort VC when it requests a reservation is also necessary, but here an intermediate router knows which receiver requested a reservation, what kind of reservation (i.e. style) and for which source(s). With this information, the mrouter can decide whether that node should be deleted from the best-effort VC and added to another, or whether it should be kept in the best-effort VC because there are other sources for which the receiver has made no reservation request (e.g. if FF is used).

  The problem in the shortcut case is now that an intermediate router does not have this information if the receiver sends its RESV messages directly to the source/ingress device. Therefore, RESV messages should be sent hop-by-hop but with an indication that they are for shortcut (this indication can be simply the presence of the ATM address object inside the RESV message). Then the router should make the appropriate actions in order to permit shortcut.

- If the decision of using shortcut is taken by intermediate router(s), this should be based on parameters like:
  - current load of the router,
  - TSpec contained in the PATH message,
  - and/or FlowSpec of the RESV message sent by the receiver.

  The aim of an intermediate router-initiated shortcut is to optimize its utilization, and at the same time, to avoid congestion (bottlenecks).

  In order to allow for the establishment of a shortcut the intermediate router should forward the RESV message to the previous hop without modifying the next hop object. This would enable the previous hop to set up a shortcut VC bypassing a possibly overloaded router. Note that the previous hop may be the ingress edge device or another router. In the first case, a complete shortcut will be established, while in the second case, two possibilities may occur. This router also takes the decision to be bypassed and also forwards the unmodified RESV message. The other choice is that the router decides not to become the start-point of the shortcut. If this happens, a "partial shortcut" from that router to the receiver will be established and the RESV message sent to the previous hop will contain

as next hop object this router's address.

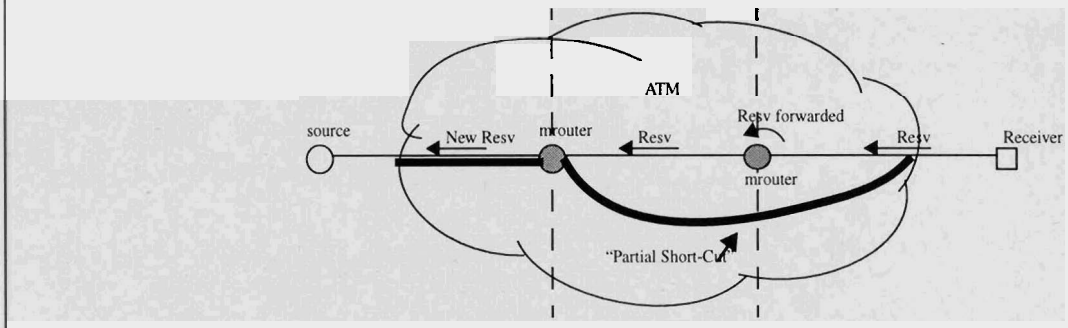


Figure 17: Partial Short-Cut.

9 Aggregation of Flows - Saving VC Space

The use of mapping models for which one flow is mapped onto one or several VCs has the advantage of directly utilizing ATM traffic control and scheduling mechanisms. The drawback of such direct mappings is the potentially large the number of VCs that are being consumed. This might lead to an exhaustion of the available VC space, or at least to a relatively expensive VC management strategy if the ATM tariffs will have a certain amount of fixed costs associated with the setup of a VC.

Therefore, it is interesting to consider VC management strategies where several RSVP flows are multiplexed over a single ATM VC in order to lower the resource consumption and price of the VC management strategy. A consequence of aggregating several flows into one VC would certainly be that ATM's traffic management mechanisms could no longer be utilized as straightforwardly as before. Further traffic control modules between RSVP and ATM would be necessary in order to manage the available resources of a VC among the different flows being multiplexed on it, ensuring the per flow reserved QoS.

The notion of a so-called aggregation model has been introduced in [BCB+98], where VCs are set up between edge devices, in order to multiplex different RSVP flows from different sessions onto them. The main advantages of this method [BCB+98], as explained in, are:

- There is no signalling latency since VCs are expected to exist already in the usual case, therefore no time will be wasted in setting up the VC.
- There is no ATM-specific problem with heterogeneity, since in this case VCs would be managed like configurable point-to-point links.
- The same applies to the dynamic QoS problem.
- It is more scalable than the 1 VC per flow model.

The use of a pure aggregation model, i.e., establishing big "pipes" between edge devices does not make use of all facilities provided by an ATM network. Take for example ATM's PNNI and its QoS routing capabilities: if an aggregation model is used then there might be situations where a 1 VC to flow mapping model would have found a path through the ATM network for a certain reservation request while with the aggregation model the request may be rejected since no special route could be supplied.

The aggregation model seems most suitable for the core of the network, since there is a large potential for multiplexing of different flows to justify the increased complexity conjuncted to it. For edge networks the 1 VC per flow models might make more sense, since scalability is of lesser importance due to a much smaller number of flows.

So, a mixed scheme using both kinds of models could be used, by first separating the IP over ATM network into two parts, as proposed in [SDMT97]:

- Access networks: ingress and egress edge devices would be part of the access network. They are expected to send/receive a "low" number of flows.
- Backbone network: routers inside the ATM network would constitute the backbone network with a large number of flows due to multiplexing of traffic.

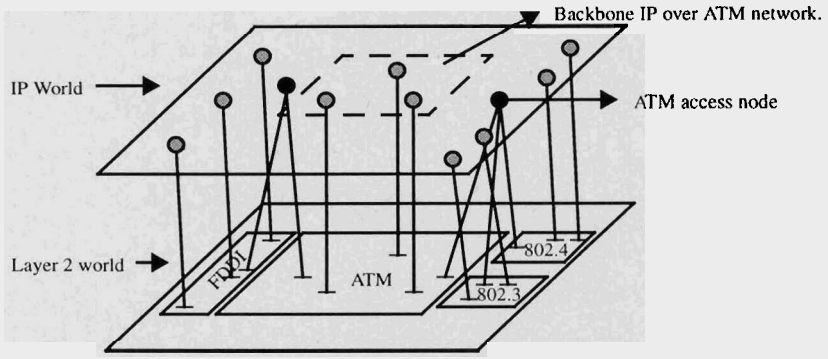


Figure 18: Access and backbone network.

In order to make this distinction between access and backbone networks, different types of nodes or devices can be identified depending on their relationship within the IP over ATM network:

- **Edge devices**: These are nodes connected to both, ATM and non-ATM networks. Their function is to work as ingress/egress points to/from the ATM network.
- **ATM hosts**: These are IP nodes, which are directly connected to the ATM network, but which do not carry out any of the functions of an IP router.
- **ATM-only routers**: These are IP routers, which only route between ATM subnetworks. Due to the high bandwidth available with ATM they are suited for the backbone of a large IP network as the Internet. Thus, ATM-only routers are expected to receive and forward a large amount of traffic.

Access Networks

When we refer to access network, we mean the nodes in the IP network which function as access points to the IP over ATM network (or simply, to the ATM network). It is clear that these nodes might actually be backbone nodes in the overall IP network, since especially nowadays ATM is still being used mainly in the backbone of the IP networks, if at all.

Nodes in the access network would be edge-devices or ATM hosts (this is a special case where the host is the access point to the ATM network for a degenerated IP network represented by the host itself). These devices could use the one or many VCs per flow models, depending on the degree of heterogeneity to be supported.

If a MCS approach is used for multicast communications (see section 11), it is not obvious to which network, the access network or the backbone network, a multicast server should belong. Even though this is an ATM-only attached device, the traffic managed might still be suitable to use a 1 VC per flow or n VC per flow mapping model.

On a hop-by-hop basis, data would be transferred from the access network to the backbone network, and finally to the access network again. Shortcuts would be an exception to this behavior, since data would be transferred directly from the access network to other device in the access network without any intervening routers. The use of shortcuts with the aggregation model seems difficult to coordinate. A shortcut is an ATM connection from an ingress device to an egress device. The purpose of a shortcut is to bypass intermediate routers exploiting a larger ATM connectivity and utilizing the benefits of P-NNI's QoS routing capabilities for a single reservation request. However, these shortcut VCs do not

have much potential for multiplexing of different flows into one VCs since their endpoints are situated in the access network and the route through the ATM network is customized to a specific reservation request. Hence, shortcuts and aggregation have an antagonistic relationship.

Backbone Network

The backbone network consists of ATM-only routers and possibly some multicast servers. In the flow multiplexing model, ATM SVCs are used to build custom bit pipes linking the ATM-only routers [SDMT97]. These routers are then in charge of collecting incoming IP flows and QoS requests, multiplexing them depending on their IP destination, and managing the characteristics of the outgoing ATM SVCs. Traffic Control Modules are necessary at the IP level to control the QoS received by each flow within an aggregated VC. From the perspective of RSVP, the VCs are viewed as logical network interfaces that can be reconfigured with regard to their bandwidth.

An obvious problem for the aggregation model is the handling of multicast transmissions if point-to-multipoint VCs shall be used in order to avoid unnecessary duplication of data in the ATM network. All flows being aggregated in a point-to-multipoint VC should have the same next hops, or otherwise, the next hops must be prepared to receive traffic not addressed to them, which however would waste resources. If the number of nodes in the backbone network is low, then there might be a certain probability that for multicast communications the next hops are the same or at least almost the same in different multicast groups. In this case, aggregating point-to-multipoint VCs may be applicable. Yet, if the number of nodes of the backbone network is becoming larger, the potential for multiplexing different flows rapidly diminishes especially if an exact match for the set of next hops is demanded. That means while the use of point-to-multipoint VC would lower the load at the entrance to the backbone due to the avoidance of data duplication, it would increase the VC usage. On the other hand, the usage of point-to-point VCs would increase the load at the entrance due to the required data replication, but would consume less VC space. Moreover, with the point-to-point VC option, there is no issue with heterogeneity since automatically a full heterogeneity model is supported. If point-to-multipoint VCs are being used, RSVP regards the aggregated VC as a broadcast medium, and thus, since ATM does not allow "variegated" VCs yet, a QoS change request will change the reservation for that flow in the whole VC. Therefore, all receivers/next hops will receive the new, higher QoS (with no delay if the VC does not need to be changed). Hence, using the aggregation model with point-to-multipoint VCs does not allow heterogeneity among the leaf nodes of each flow reserved, the same way that it is not supported in Ethernet networks. The complexity of using the aggregation model with point-to-multipoint VCs leads us to the conclusion that this model is more suitable for point-to-point VCs, between a reduced set of nodes.

As mentioned already, if the aggregation model is being utilized, changes in QoS reservations are managed at the IP level by means of RSVP and the traffic control modules. In the usual case, no changes should be necessary at the ATM level. Only in exceptional cases, if the modified reservation requests causes the resource consumption of an aggregated pipe to fall below or above certain thresholds, a VCs QoS needs to be modified, i.e., with current signalling to be torn down and setup with the new parameters. It is of course far from obvious how to choose these thresholds in order to cause the modification of a VC to be an exceptional event. However, the advantage of incurring no signalling latency for the setup of a reservation depends on that property.

The discussion above suggests that the aggregation model is not very suitable for changing environments, where new receivers are joining or leaving groups rapidly and reservations are changing frequently. However, it is suitable if the number of nodes is small, but the amount of traffic managed is large and on average relatively stable. Accordingly, the aggregation model is only suitable for the backbone network, whereas for the access network, other flow to VC mapping models are preferable.

Alternatively, an aggregation model at the RSVP level, as proposed in [BV98] might be used, which would also decrease the RSVP state necessary in backbone nodes.

10 Dynamic QoS

From the point of view of multicast, the emulation of RSVP's dynamic QoS over the static ATM QoS depends on the heterogeneity model that is being used. The extreme case is using a homogeneous QoS model. If the new QoS requested is larger than the QoS of the existing point-to-multipoint VC, the QoS of the VC must be changed, i.e. the VC must be torn down and setup again with new parameters. This is the same situation as if a new receiver requested more resources than the already existing reservation offers.

For models that permit some degree of heterogeneity, which means different VCs are setup for the same session, a modified reservation might be honored by just deleting a node from one VC and add it to another if the modified QoS can be provided by that VC. This less resource-consuming management of dynamic QoS changes is a further argument for some support of heterogeneity within the ATM network.

Since dynamic QoS changes lead to a certain overhead for signalling and possibly associated costs, the main issue is the question how to avoid such changes as far as they result in VC management actions. A possible strategy could be to allocate some extra resources for the ATM connection [BCB+98]. An RSVP over ATM implementation should allow tuning this amount of extra resources, so that experience with this parameter will demonstrate whether it is really useful and how much more than what was requested should be allocated. This is of course dependent on how dynamic future applications are with respect to their resource reservations. Furthermore, economic parameters will play an important role. The cost of over-allocating some resources during a certain time period must be amortized by the reduced rate of necessary QoS changes at the ATM level, thereby lowering the fixed costs for setting up VCs.

11 Multicast Data Distribution

The IP Multicast model does not require explicit knowledge about which source(s) are sending or may send data to a multicast group. The connectionless characteristic of IP allows that any node in the network can send data to the group address, thus becoming a source for that group (possibly even without being a member of that group). Therefore, multipoint-to-multipoint communications are possible. In this model, the source needs no information about the receivers for that group, either. For IP multicast transmissions, multicast routing protocols like CBT, MOSPF, DVMRP, PIM(SM or DM), etc. create multicast distribution trees whose leaf nodes are those routers, in whose attached networks there are group members.

On the other hand, the non-broadcast characteristic of ATM necessitates the knowledge about the identity of the receivers of a multicast group, in order to be able to setup the data paths using point-to-multipoint VCs. Procedures, as e.g. MARS, for getting this information have already been discussed in section 5. The problem now we are concerned with now is the establishment multipoint-to-multipoint communications over ATM networks. There are different solutions:

1. Make multipoint-to-multipoint VCs available on ATM networks: SMART [GLO96].
2. Set up point-to-multipoint VCs from each virtual source to the ATM network: VC-Mesh.
3. Use of Multicast Servers (MCS) as points of traffic aggregation, i.e., each source sets up a point-to-point VC to the MCS, while the MCS sets up a point-to-multipoint VC to the receivers of the group.
4. Use a mixed solution between 2. and 3.

While 1. is certainly the most complete solution, it is not a reality with current ATM signalling and might never be provided by ATM networks. The other solutions are basing on current signalling facilities and thus take a pragmatic approach. We will discuss these in more detail. Further discussions can be found in [BCB+98], [Arm96] and [TA97].

11.1 VC-Mesh

The VC-Mesh solution is the most straightforward solution. In this case, the multipoint-to-multipoint communication is realized by means of several point-to-multipoint connections. Its main advantages are:

- its simplicity,
- that it allows flexibility for VC management, since each ingress edge device can request the appropriate QoS for its point-to-multipoint VC or even use several VCs if heterogeneity is desired,
- its low latency, since no intermediate reassembly is needed,
- that ATM signalling can ensure optimal branching points.

On the other hand, its disadvantages are:

- that it produces a higher signalling load, especially if dynamic membership is taken into account,
- resource consumption is growing linearly with the number of sources.

11.2 Multicast Servers

A multicast server(MCS) is a device that accepts cells from multiple senders and sends them via a point-to-multipoint VC to a set of receivers. The MCS reassembles AAL-SDUs arriving on all the incoming VCs and queues them for transmission on the single outgoing point-to-multipoint VC. The reassembly is required because AAL5 does not support cell level demultiplexing of different AAL-SDUs. Although AAL3/4 does, it is not a good encapsulation method for IP packets, since it involves a much higher overhead than AAL5 encapsulation.

A side effect of using MCS is that ATM endpoints which are both, sender and receiver, receive a copy of packets sent by themselves, thereby creating circles if no action is being taken. To solve this problem the MCS could retransmit packets on individual VCs between itself and group members. This would, however, decrease the good scalability of the MCS approach.

Although an MCS could use different VC management strategies in order to allow for some level of heterogeneity support, it is not as flexible as the VC mesh approach, since data can no more be provided different QoS according to its sender.

The main advantages of using the MCS approach is the relatively low consumption of network resources and the lower signalling load in case of very dynamic sets with a large number of sources, since these dynamics must only be honored at the MCS's point-to-multipoint VC. Therefore the MCS approach scales constantly with the number of sources.

However, its main drawbacks are the comparably higher delay for packets due to the reassembly of packets at the MCS, and the fact that the MCS can potentially become a bottleneck and a central point of failure. Using multiple MCS can help to solve these last two problems.

It must also be noted that if a MCS makes no IP level processing for the incoming packets, thus only doing AAL5-SDU reassembly, it can only serve one multicast group at a time, since it has no way to distinguish among packets addressed to different groups. With a minimum amount of IP processing, a MCS can be used by several groups, thus improving its utilization [Arm96].

11.3 Intermediate Solutions

Both, the VC-Mesh and the MCS approach, have advantages and disadvantages. Therefore, there might be intermediate solutions which are able to combine some of these advantages while avoiding some disadvantages. So let us reconsider:

1. *When is the VC-Mesh approach useful ?*

   If there is only one or at least a very low number of sources, so that the signalling overhead necessary for membership changes and dynamic QoS is still reasonable. In this case, the number of

additional VCs (if there is more than one source) does not justify the use of a MCS yet. The VC-Mesh approach is also useful if the data has very low delay requirements, thus not allowing the for the additional processing delay introduced by a MCS.

2. *When is the MCS approach useful ?*

   If the number of sources becomes larger, there is a considerable saving in number of VCs when using the MCS approach. However, it must be taken care that the amount of traffic being forwarded by the MCS does not become too large in order to avoid the MCS becoming a bottleneck on the data path. Therefore, the use of a MCS depends on:
   - the number of sources,
   - the expected traffic from the sources, and
   - the current load of the MCS.

Regardless of the scheme being used for multipoint-to-multipoint communication over ATM, the options for inter-cluster communication, shortcut or hop-by-hop, must be taken into account. Let us consider the best-effort and QoS case separately.

Best-Effort Case

In the best-effort case, the VC-Mesh approach involves the use of n, the number of sources, point-to-multipoint VCs. In the MCS case, there would be n point-to-point VCs and one point-to-multipoint VC. Since the number of sources is not known in the best-effort case and must thus be assumed to be potentially large, the savings in number of VCs and signalling overhead argue for the MCS approach. The drawback is the added delay. However, for a best-effort service delay requirements should not be very strict. Possible approaches to improve the straightforward use of the MCS approach for best-effort transmissions in order to avoid congestion in the MCS are to:

- limit the number of sources a MCS can support, such that, if this limit is reached, another MCS could be used, or new sources could use the VC-Mesh approach, or
- measure the current traffic load of the MCS and allow according to this measurement new sources to either be supported by this MCS, by another MCS or by the VC-Mesh approach.

These decisions should certainly be transparent for a source, which just issues a MARS request and receives a reply with either the ATM addresses of the receivers or with the ATM address of the MCS in charge for that multicast group. However, the MARS and MCS components need to be extended to coordinate each other in order to make the MCS state information available to MARS. An algorithm that could be run at the MARS to take the decision between using VC-Mesh or MCS would be:

1. *Receive MARS_REQUEST message.*
2. *If there is no other source "active" for that group*
   - 2.1. *Use Mesh and register this source as "active"*
   - 2.2. *Go to step 1.*
3. *If MCS is "overloaded"*
   - 3.1. *Use VC-Mesh and register this source as "active"*
   - 3.2. *Go to step 1.*
4. *Register this source as "active" and use MCS.*
5. *Go to step 1.*

This is certainly a very simple algorithm which would need further improvement. For example, with this algorithm, the first source of a group will always use the VC-Mesh approach, regardless of how many sources will be become active for that group later on. An alternative would be to use the MCS approach already for the first source becoming active. The obvious drawback, however, would be the needless VC from the source to the MCS and the added delay if the source remains the only one for this group. A better approach might be to switch the first source from the VC-Mesh to the MCS mode by using the MARS_MIGRATE message sent from MARS to the cluster client.

With regard to the registration of a cluster client as an active source, the MARS_REQUEST message could be taken as an indication for that condition. Having said above that cluster clients will not have to be modified, an exception would be an explicit de-registration as an active source in order to not blockade resources uselessly and to allow MARS to take the best decision about using VC-Mesh or MCS mode for a specific MARS_REQUEST.

An example scenario of how such a mixed model of VC-Mesh and MCS approach could look like is illustrated in figure 19.

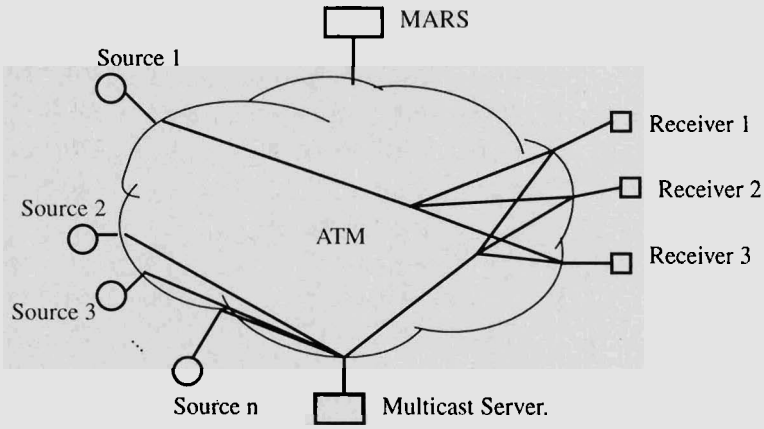


Figure 19: VC-Mesh and MCS: Mixed Model.

As already mentioned, there needs to be some coordination between MARS and a MCS in order let MARS know about the current load situation at the MCS. For this purpose, two new messages should be added to the MARS-MCS protocol.

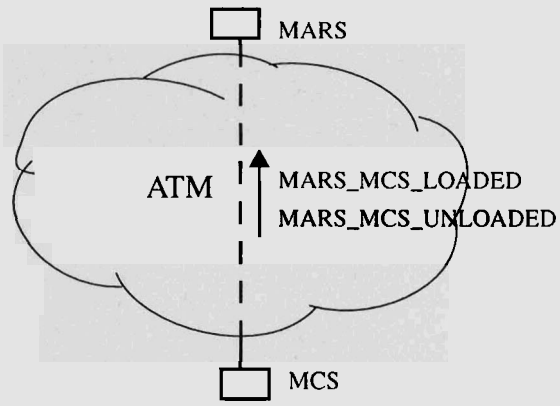


Figure 20: MCS load messages.

MARS_MCS_LOADED: sent by the MCS to MARS indicating that no more sources can be supported by the MCS currently.

MARS_MCS_UNLOADED: Sent by the MCS to MARS indicating that its load state permits more sources again (after a MCS_LOADED message was sent).

QoS case

On the basis of the above scheme, some extensions are necessary in order to allow for the transmission of QoS traffic. In the QoS case delay requirements certainly become more important and should be taken into account. Furthermore, the MCS should be modified in order to allow for VCs of different QoS. Otherwise, the point-to-multipoint VC would not have the requested QoS characteristics. Moreover, heterogeneity support would not be possible. In fact, the MCS should more or less perform like any other RSVP capable multicast router.

For QoS communications with very strict delay requirements, the use of shortcuts is, as already explained, an interesting option. A trade-off between shorter latency and added resource consumption must be made in case that there are several sources and receivers requesting QoS. It must be realized that with shortcut sources in different clusters the VC consumption of the VC-Mesh is becoming even larger than for VC-Mesh with hop-by-hop forwarding. However, a compromise could be to use the MCS approach but with shortcut point-to-multipoint VCs from the MCS to the receivers. Here, the delay would be longer than for a complete shortcut but shorter than for hop-by-hop forwarding, and the benefits of the MCS approach in terms of VC usage could be exploited. The different scenarios are depicted in figure 21.

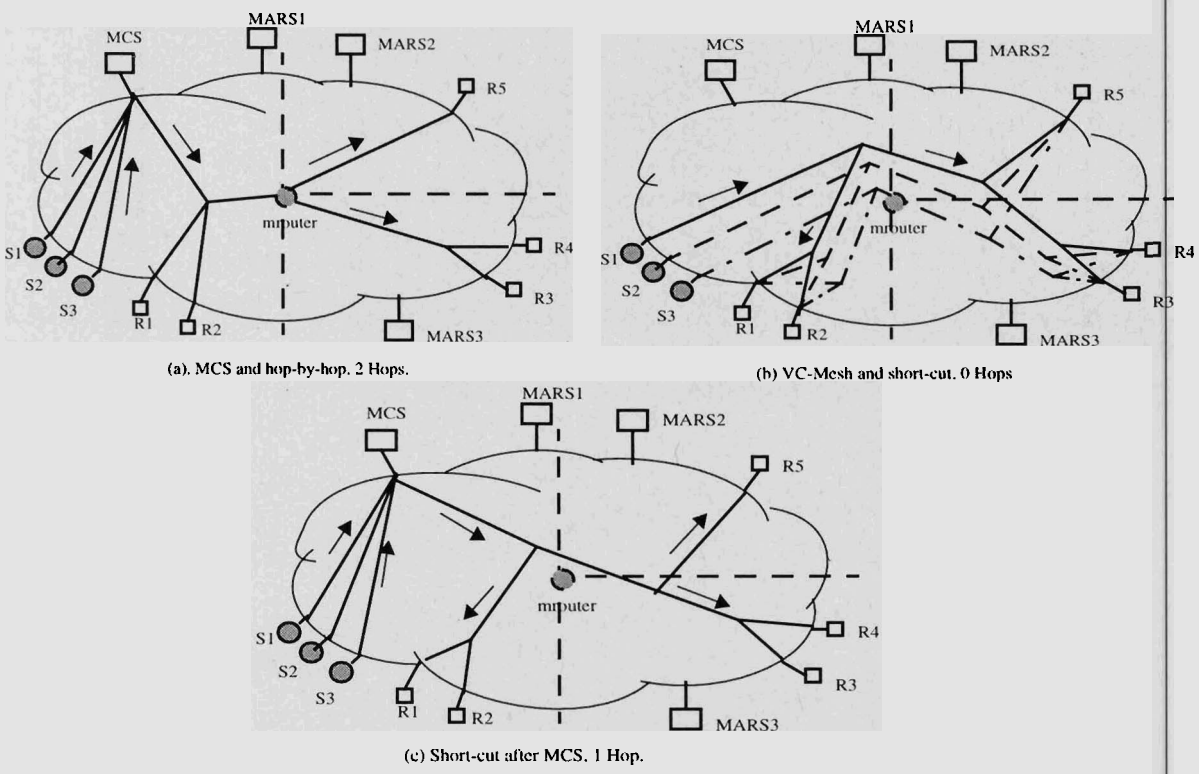


Figure 21: Short-cut and MCS/VC-Mesh.

In case of shortcutting from the MCS, the MCS would be the next hop in the data path for the sources in the cluster. The MCS should process the PATH messages by including its ATM address into them, so that shortcuts are rooted in it. The MCS would receive RESV messages from the receivers, like any other ingress edge device, and would merge the reservations appropriately in order to send out new RESV messages upstream to the virtual sources in the cluster.

12 Heterogeneity Management

In order to show the problems associated with heterogeneity support, let us look at the network in figure 22, and follow the different steps in which receivers request their reservations. For this discussion, it is not important whether MARS or EARTH or any other multicast address resolution method is used. In this case we assume that MARS is being used.

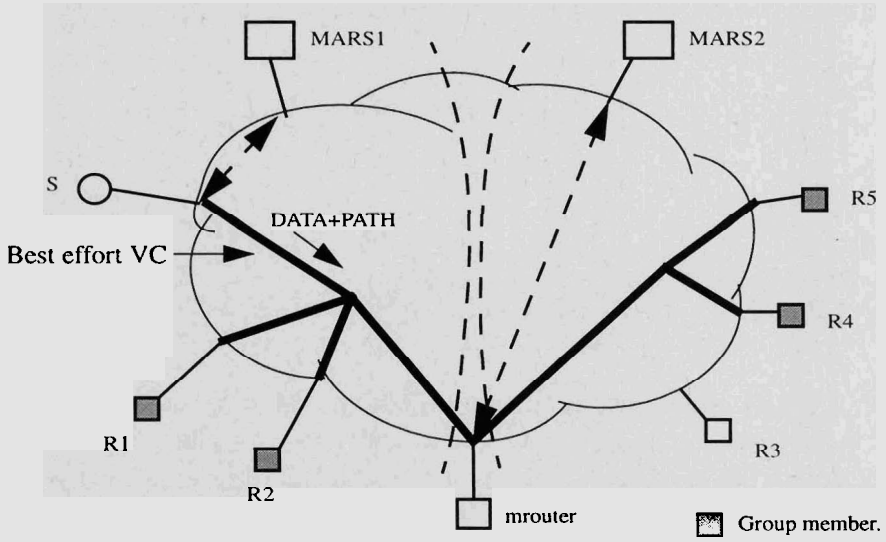


Figure 22: Group members in various LIS/Clusters.

In figure 22 we have a concatenation of VCs which serves as a best-effort distribution tree on which data without any reservations is being sent, as well as RSVP control messages. If R1 now sends a Resv message, S should set up a new reservation, but how should this be done exactly:

1. Setup a new QoS VC from S to R1 and then delete R1 from the best-effort tree.
2. Setup a new QoS VC as well, but do not delete R1 from the point-to-multipoint best-effort VC. In this case R1 would receive duplicate data. Nevertheless, it can be necessary not to delete the receiver from the best-effort VC in case S is a router. In this case, data from more than one source to the same group could be received and forwarded. If R1 makes a reservation for only one of them, it should still receive data from the other sources on a best-effort basis.

Let us assume that option 2 is chosen due to the reasons given above. This means that S now needs to duplicate data, one copy on the best-effort VC and the other on the QoS VC.

Let us suppose now that R2 issues a larger reservation than the reservation of R1. What are the reasonable options for VC management now ?

1. Setup a new point-to-point VC from S to R2:
   - S must triplicate the data it sends.
   - There is a point-to-multipoint best-effort VC, a point-to-point QoS VC to R1 and another one to R2.
2. Setup a point-to-multipoint VC from S to R1 and R2 with the biggest reservation (R2) and delete the VC to R1. Now, the resources allocated in the data path to R1 are larger than it had requested. Then,
   - What should R1 pay for ? (If it has to pay)
   - S only needs to duplicate the data.
   - We are saving resources in S and on the common path of R1 and R2, but we are wasting them when the path of the point-to-multipoint VC is separated into two subtrees.

The choice among all these different options in our example for managing heterogeneity should certainly also be made dependent on economic factors. This is assuming that the edge devices which have

to do the VC management belong to a different provider than the ATM network. In our example the provider operating the edge device S will certainly make its choice dependent on the prices of the provider of the ATM network, i.e., whether it is cheaper to use two point-to-point VC, a "large" one (for the reservation of R2) and a "small" one (for R1), or one point-to-multipoint VC with two branches, but with the resources of the "big" one.

For example, let us suppose the following pricing scheme for ATM VCs:

$$C(N, B, t) = \beta_1 + \beta_2 N + \beta_3 NBt$$

where

- C = Cost of the ATM connection
- N = Number of leaf nodes in the connection
- B = Bandwidth reserved for the ATM connection
- t = Duration of the connection

The model of the example tries to take into account the bandwidth requirements and the number of leaf nodes of the VC connection, by means of the terms with parameters β_2 and β_3. The more bandwidth is being requested or the more leaf nodes on the connection the more expensive the connection will be. On the other hand, it is also taken into account how long the communication lasts, using the term with parameter β_3 again.

This is of course only a very simplified pricing model which does neither take into account many other parameters that could also have influence on prices nor regards more elaborated relationships between those parameters. The purpose of this example is to show that the VC management strategy in order to support heterogeneity may depend also on the pricing scheme.

Let us suppose now, two reservation requests, with bandwidths B1 and B2 (with B1 > B2). The two possible options are: one point-to-multipoint VC with B1 resources reserved, or two point-to-point VCs with B1 and B2 reserved.

1. Option: 1 point-to-multipoint VC with B1.

$$C_1 = \beta_1 + 2\beta_2 + 2\beta_3 B_1 t$$

2. Option: 2 point-to-point VCs with B1 and B2.

$$C_2 = 2\beta_1 + 2\beta_2 + \beta_3 (B_1 + B_2) t$$

With these two options, the price of both options will be the same if:

$$t = \frac{\beta_1}{\beta_3 (B_1 - B_2)}$$

If the duration of the connection t is longer than the right hand side of the equation than option 2 will be cheaper. Otherwise, option 1 will be the better choice. Therefore, it is not possible to know a priori which choice is economically best, if the pricing model is not known (and the duration of the session is not known).

If the edge device is on the premises of the ATM network provider then the decisions for VC management with regard to heterogeneity support will rather be based on resource consumption than on a pric-

ing model. However, it is neither possible to determine generally which option saves more resources in the ATM network, because it is network topology dependent.

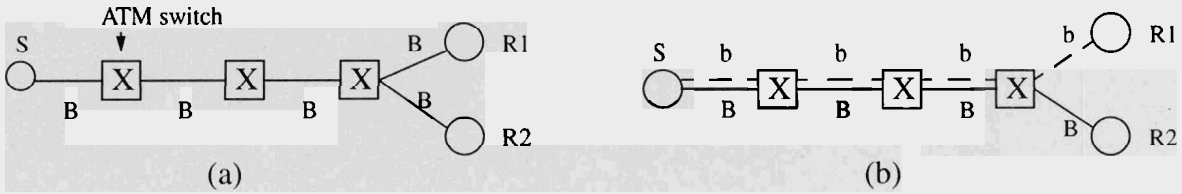


Figure 23: a) One point-to-multipoint VC. b) Two point-to-point VCs with.

In figure 23, more network resources are being wasted in the case (b) than in case (a), if we assume that b>1/4B. However, a counterexample is shown in figure 24, where using one point-to-multipoint VC wastes more resources than using two point-to-point VCs if we assume that B>4/3b.

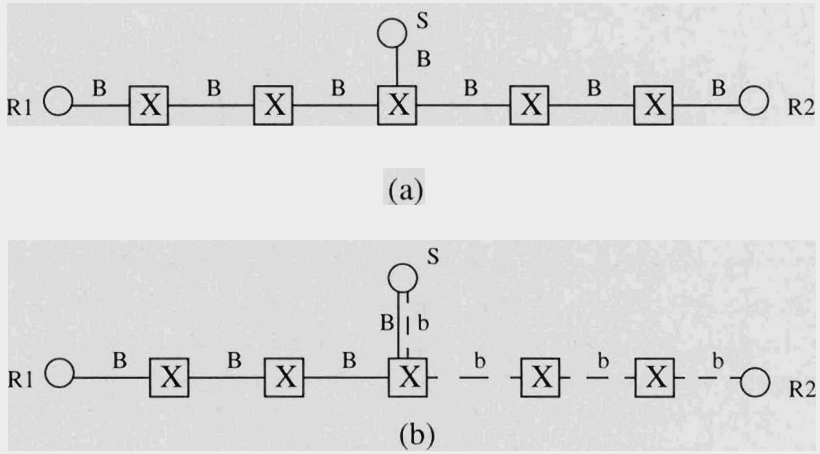


Figure 24: a) One point-to-multipoint VC. b) Two point-to-point VCs.

Both figures show that the best solution depends on the topology, not only of the ATM network but also of the IP network, and on the amount of resources being reserved. Therefore, it would be helpful to let the process that maps IP/RSVP onto ATM take part in the ATM-PNNI, in order to obtain information about the ATM network topology and thereby allow for resource-optimal flow-to-VC mapping strategies.

Apart from efficient use of network or financial resources, the decisions for VC management are also affected by:

- the possibility of supporting heterogeneity as exact as possible,
- the scheme's simplicity with regard to reservation changes or new reservations,
- the signalling load that is being produced especially at the edge device,
- the scalability of the scheme to large ATM networks.

The combination of IP Multicast and RSVP permits heterogeneous receivers, i.e., each receiver can request and receive a different QoS. However, ATM networks do currently not support that property at the point-to-multipoint VC level, all the branches of a VC offer the same QoS to the receivers.

Let us reconsider what is the value of heterogeneity support from the point of view of a user respectively a network:

1. From users' perspective, heterogeneity means that each user can request the QoS that is sufficient for him. Moreover, the user can be sure not to receive data flows with higher QoS than requested, which its local resources might not be able to handle.
2. From networks perspective, heterogeneity has the potential to avoid network resources wastage,

as it allows reserving only what is necessary for each receiver.

As already mentioned, using one point-to-multipoint VC for all QoS receivers does not allow heterogeneous reservations in the ATM network. At the other end of the spectrum a point-to-point VC to each of the receivers would allow for an exact support of heterogeneity from the ATM network user's point of view. Of course, such a solution is not scalable to a large and very heterogeneous set of receivers, because it is not really multicast but "multi-unicast". However, depending on the significance that is given to the heterogeneity property and the concrete situation it might still be a reasonable option.

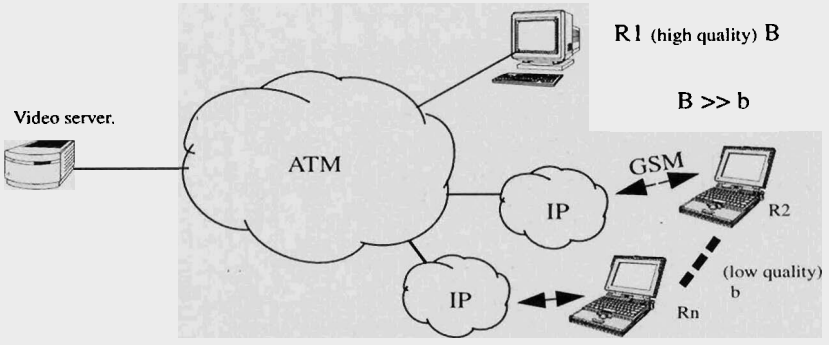


Figure 25: Video services example.

Consider for example the situation as depicted in figure 25, where we have a video multicast to a very heterogeneous set of receivers. There is only one source and several receivers, of which one requests a very high quality especially with respect to bandwidth, while the others either only want a lower quality transmission or can even not cope with a high bandwidth stream due to for example being connected by a radio channel for mobile receivers or the conventional phone link. If the video service is commercial then the receivers might, for example, pay the video service provider according to the quality of transmission they request, and the video service provider might in turn pay the ATM network operator for the ATM services according to parameters like number of VC setups, bandwidth, duration, etc.

In this situation, the content provider could choose to use one point-to-multipoint VC of bandwidth B. This solution is straightforward and involves low processing overhead for the source. The receivers R2,...,Rn would be delivered a better QoS than they requested and pay for. Thus the service provider wastes financial resources for receivers which might not even be able to cope with those higher QoS data flows.

An alternative could be to setup a point-to-point VC with bandwidth B to the R1, and one point-to-multipoint VC of b to R2,...,Rn. In this case, each receiver gets exactly what he/she requested, and the ATM network resources are being used more economically than before. The drawback of this approach is the wastage of resources at the subnet-sender and in the ATM network, since two copies of the same data must possibly be sent over some common links. With two different QoS levels supported, this overhead might not be problematic, but for the extreme case of having as many VCs as receivers, the virtual source to the ATM network is likely to become overloaded.

The optimal solution would certainly be the provision of "variegated VCs" [BCB+98] by the ATM network, which would allow for heterogeneous receivers in the same VC. But such facilities are not available with ATM's current signalling protocols and switching hardware.

From the example we can draw some conclusions:

1. With current ATM signalling there is no optimal solution valid for every situation and from every point of view to the problem of heterogeneity support.
2. For this optimal solution "variegated" VCs would be needed.
3. A certain degree of heterogeneity support by the ATM network can be achieved using VC management strategies which take into account:

- economic factors respectively resource consumption,
- the current load situation at the virtual source to the ATM network,
- number of receivers and the diversity of their QoS requests.

In the following a simple algorithmic framework for such a VC management strategy is given.

Simple VC Management Algorithm

With this algorithm, there will be as many VCs per flow as an ingress edge device can manage, supporting a controllable level of heterogeneity.

```
New reservation request R arrives (for a MC group which is already being
delivered QoS data)
STEP 1: Look for a VC whose QoS is 'similar' (but greater) to R's
STEP 2: IF (VC found)
                Add receiver to that VC
                RETURN
STEP 3: IF ('source not overloaded')
                Create new VC to that receiver with R's QoS
                [Reorganize()]
                RETURN
STEP 4: IF (ClaimBack(R's QoS))
                Create new VC to that receiver with R's QoS
                [Reorganize()]
                RETURN
        ELSE
                Reject R
                RETURN

Claimback(Q): tries to merge VCs together to regain at least Q
              resources and return TRUE on success and FALSE
              otherwise.
Reorganize(): tries to assign older reservations to the new one
              subject to the similarity definition and aiming at the
              release of resources.
```

Alternatively to rejecting R in the ELSE-branch of STEP 4 one could also redefine the similarity definition and go back to STEP 1 (if R is not bigger than all the existing reservations) hoping for an existing VC to be now similar enough to the new reservation request.

Depending on how we define similarity and source overload we obtain a spectrum of behavior between the homogeneous and full heterogeneity model. It is of course difficult to define the similarity concept between QoS requests, since those requests are multi-dimensional in nature and are thus potentially only partially ordered. So, probably, for the definition of similarity there are some simplifying assumptions needed, as e.g., the restriction onto some or even just one parameter like bandwidth.

If similarity is defined as an exact match then we obtain a full heterogeneity support model as a limiting case. Whereas the other limiting case where all requests are regarded as similar degenerates into a homogeneous QoS support model, where only one VC per RSVP session is allowed. The intermediate cases are certainly the most interesting ones giving more flexible heterogeneity support without necessarily admitting any new reservation request to obtain a new VC. Therefore, the definition of similarity allows to control the support of heterogeneity within the ATM network.

Another determinant with regard to the supported level of heterogeneity is the definition of "source overload". In general, if the source has to manage too many VCs for one RSVP session it will become overloaded. Thus there must be a condition at the source which limits the number of VCs per session. This condition can be indicated by different metrics, like e.g., internal queue lengths, processor load, or just the number of already existing VCs for all or the particular session. The exact setting of that condition is a local decision of the ingress edge-devices.

Two further components of the VC management algorithm framework are the functions `Claim-back(Q)` and `Reorganize()`, which essentially try to reduce the load at the virtual source at the cost of reducing the level of heterogeneity support. This could be done by e.g. merging two very similar VCs into only one, thus reducing the copies of data to be sent from the virtual source.

This algorithmic framework tries to offer a more flexible scheme for providing heterogeneity support by means of VC management. The exact behavior of a concrete algorithm can vary from a homogeneous to a full heterogeneity support model, depending on the definition of the similarity concept and the condition for source overload. The most interesting models are those in the middle of the spectrum subject to the assumption that heterogeneity support is regarded as an important service in future integrated services networks.

13 VC Management for Heterogeneous QoS Multicast Transmissions

As we have illustrated in the last section, there is a particularly hard problem with RSVP's support of heterogeneous reservations, since ATM only allows for a homogeneous QoS within a single VC. The focus of this section is on how this difference can be actually bridged to allow for an efficient support of RSVP over ATM with regard to that issue. 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 discuss whether heterogeneous QoS is possible and useful. In section 13.2, VC management strategies are discussed – we review related work, and present our own schemes. As argued in section 13.3, the currently defined RSVP traffic control interface is not capable to support NBMA (Non-Broadcast Multiple Access) networks and VC management strategies in particular.

13.1 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 [MJV96] wide-spread interest. Yet, if used widely and potentially even combined with object-oriented [ISO98] or thin-layered coding schemes (e.g., [WSS97]), 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:

- Change RSVP to disallow heterogeneous reservations, respectively force them to a homogeneous QoS. While not very attractive, this is somehow already the case nowadays because "excess traffic" is not dropped in routers but forwarded as best-effort. Yet, this might lead to overload further downstream and unpredictable overall QoS.
- Ignore the problem and use just one QoS within the ATM subnetwork. This approach can be seen as similar to the last one. 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.

13.2 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 is 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 just 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.

13.2.1 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 [BCB+98] the following models to support heterogeneous reservations over an ATM subnetwork:

Full Heterogeneity Model. In the full heterogeneity model (see Figure 26), point-to-multipoint VCs are provided for all requested QoS levels plus an additional point-to-multipoint VC for best effort receivers.

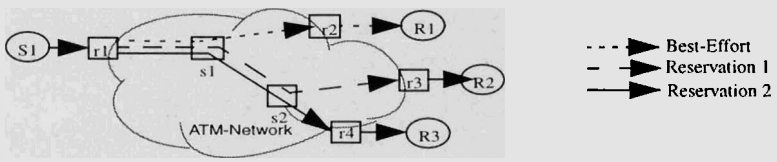


Figure 26: The Full Heterogeneity Model.

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.

Limited Heterogeneity Model. 1 In the limited heterogeneity model (see Figure 27), one point-to-multipoint VC is provided for QoS receivers while another point-to-multipoint VC is provided for best-effort receivers.

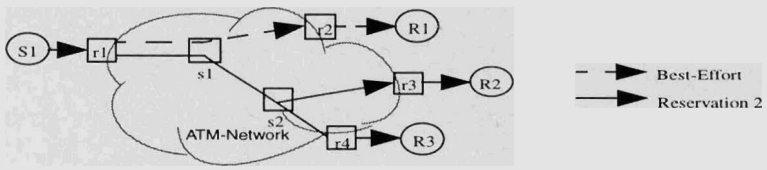


Figure 27: The Limited Heterogeneity Model.

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 [SCSW97]. 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 [CSS+97]). How to make these decisions in an efficient manner is exactly what we will deal with in the rest of this section.

13.2.2 Administrative Location of the Edge Device

In Figure 28 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.

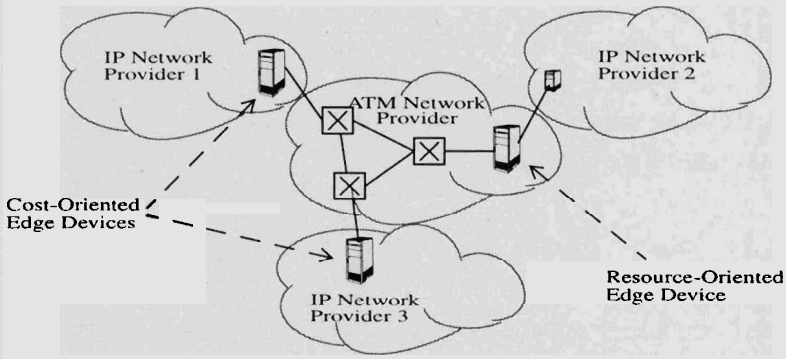


Figure 28: Different Types of Edge Devices.

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.

13.2.3 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.

13.2.3.1 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 ∈ S with QoS q;
c(S) = f(S, q(min S)) for S ⊆ R;

Call p = {R_1,...,R_n} a partition of R, if $R_1 \cup \ldots \cup R_n = R$ and ∀i,j: $R_i \cap R_j = \varnothing$.

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 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 $|S_p(N-k-1)|$ alternatives (per definition). In Table 6 (next page) 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 to 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 this search 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 you obtain $|S_p(15)|$ = 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 f (the price function) is subject to

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

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

The proof of Theorem 1 can be found in Appendix A of this section.

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 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 A 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 6.

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

Table 6: Growth of the Complete Partition Space and 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).

13.2.3.2 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.

13.2.4 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.

13.2.4.1 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 analog 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

[SCSW97] 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.

[SDMT97] L. Sagarelli, M. DeMarco, G. Meroni, and V. Trecordi. Efficient Transport of IP Flows Across ATM Networks. In *IEEE ATM '97 Workshop Proceedings*. IEEE, May 1997.

[Smi97] M. Smirnov. EARTH- EAsy IP multicast Routing THrough ATM clouds, March 1997. Internet Draft, work in progress.

[SPG97] S. Shenker, C. Partridge, and R. Guerin. Specification of Guaranteed Quality of Service, September 1997. RFC 2212.

[SWS97] J. Schmitt, L. Wolf, and R. Steinmetz. Interaction Approaches for Internet and ATM QoS Architectures, August 1997. 2nd Milestone of the IQATM-Project Phase 1.

[TA97] R. Talpade and M. Ammar. Multicast Server Architectures for MARS-based ATM multicasting, May 1997. RFC 2149.

[Wro97] J. Wroczlawski. Specification of the Controlled-Load Network Element Service, September 1997. RFC 2211.

[WSS97] L. Wu, R. Sharma, and B. Smith. Thin Streams: An Architecture for Multicasting Layered Video. In *Proc. of NOSSDAV '97*. IEEE, May 1997.

C. Liu, P. Sharma, and L. Wei. Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification, June 1997. RFC 2117.

[FCD98] 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.

[Fen97] W. Fenner. Internet Group Management Protocol, Version 2, November 1997. RFC 2236.

[FMR97] D. Farnacci, D. Meyer, and Y. Rekhter. Intra-LIS IP multicast among routers over ATM using Sparse Mode PIM, August 1997. Internet Draft, work in progress.

[GB98] M.W. Garrett and M. Borden. Interoperation of Controlled Load and Guaranteed Service with ATM, June 1998. Internet Draft, work in progress.

[GKW97] R. Guerin, D. Kandlur, and D. Williams. Extensions to the MARS model for Integrated Services, September 1997. Internet Draft, work in progress.

[GLO96] E. Gauthier, J.Y. LeBoudec, and Ph. Oechslin. SMART:A many-to-many multicast protocol for ATM . Technical Report Technical Report, LRC Lausanne, August 1996.

[ISO98] ISO/IEC JTC1/SC29/WG11: MPEG-4 Systems Final Committee Draft, March 1998.

[ITU94] ITU-T: Rec. Q.2931: B-ISDN User-Network Interface Layer 3 Specification for Basic Bearer/Caller Control, March 1994.

[JW97] J. Jamison and R. Wilder. vBNS: The Internet Fast Lane for Research and Education. *IEEE Communications Magazine*, 35(1), January 1997.

[KLS98] 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.

[Lau94] M. Laubach. Classical IP and ARP over ATM, January 1994. RFC 1577.

[LKPC98] J. Luciani, D. Katz, D. Piscitello, and B. Cole. NBMA Next Hop Resolution Protocol (NHRP), April 1998. RFC 2332.

[Mil95] W. Milliken. Integrated Services IP Multicasting over ATM, July 1995. Internet Draft, work in progress.

[MJV96] S. McCanne, V. Jacobson, and M. Vetterli. Receiver-driven Layered Multicast. In *Proc. of ACM SIGCOMM'96*, August 1996.

[Moy94] J. Moy. Multicast Extensions to OSPF, March 1994. RFC 1584.

[Sch98] J. Schmitt. Extended Traffic Control Interface for RSVP. Technical Report TR-KOM-1998-04, Darmstadt University of Technology, July 1998.

References

[Arm96] G. Armitage. Support for Multicast over UNI 3.1 based ATM Networks, November 1996. RFC 2022.

[Arm97a] G. Armitage. Issues affecting MARS Cluster Size, March 1997. RFC 2121.

[Arm97b] G. Armitage. VENUS - Very Extensive Non-Unicast Service, September 1997. RFC 2191.

[ATM95] ATM Forum Technical Commitee: User-Network-Interface (UNI) Specification 3.1, June 1995.

[ATM96a] ATM Forum Technical Commitee: Traffic Management (TM) Specification 4.0, April 1996.

[ATM96b] ATM Forum Technical Committee: User-Network-Interface (UNI) Signalling Specification 4.0, July 1996.

[ATM96c] ATM Forum Technical Committee: Private Network-Node Interface (PNNI) Signalling Specification, March 1996.

[BBC+98] D. Black, S. Blake, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services, May 1998. Internet Draft, work in progress.

[BCB+98] L. Berger, E. Crawley, S. Berson, F. Baker, M. Borden, and J. Krawczyk. A Framework for Integrated Services with RSVP over ATM, May 1998. Internet Draft, work in progress.

[BCS94] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: an Overview, June 1994. RFC 1633.

[BFGK96] A. Birman, V. Firoiu, R. Guerin, and D. Kandlur. Provisioning of RSVP-based Services over a Large ATM-Network. In *Proc. of IEEE Global Internet.* IEEE, November 1996.

[BV98] S. Berson and S. Vincent. Aggregation of Internet Integrated Services State, August 1998. Internet Draft, work in progress.

[BZ97] R. Braden and L. Zhang. RSVP Version 1 Message Processing Rules, September 1997. RFC 2209.

[BZB+97] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource Reservation Protocol (RSVP) - Version 1 Functional Specification, September 1997. RFC 2205.

[CSS+97] 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.

[EFH+97] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson,

14 Summary and Conclusion

In this report, the most significant problems related to the implementation of Integrated Services IP Multicast over ATM have been discussed and some schemes for resolving these problems have been suggested.

In particular, after reviewing the concepts of IP Multicast, RSVP and multicasting in ATM, we covered the problem areas of group membership management, VC management for data and control traffic, shortcutting over ATM, aggregation of RSVP/IP flows, dynamic QoS provision over ATM and management of heterogeneity over ATM. We proposed several solution approaches to these issues and contrasted them against existing work in this area by showing the different tradeoffs of the approaches.

For the issue of heterogeneity support over the ATM subnetwork we went into even more detailed descriptions of how actual VC management strategies could look like. We differentiated these strategies according to the fact whether the edge device is situated on the premises of the ATM network provider or not. That led us to different algorithms for each case. We showed how these algorithms could achieve a significant gain in either reduced costs or saved bandwidth when compared to simple schemes as proposed in the literature.

From the proposed schemes for the different problem areas in RSVP/IP Multicast over ATM we can draw the conclusion that they certainly do not simplify the solution of mapping the two architectures, but can become quite complicated. It is therefore always necessary to consider the particular environment in which the mapping is required. Only then it is possible to decide whether a certain optimization as,e.g., shortcuts is worthwhile the higher implementation and operational overhead of the mapping solution or not.

Appendix A – Proofs

Proof of Theorem 1:

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

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

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

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

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

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.

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 (Figure 31).

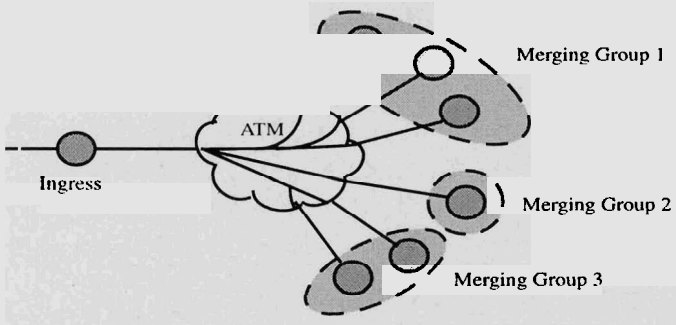


Figure 31: Merging Groups.

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. 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 will be discussed in a companion technical report [Sch98].

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`, but further study of that parameter is needed.

13.2.4.2 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).

13.3 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 ([BZB+97],[BZ97]) 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 [BZB+97], 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

waste a lot of bandwidth for the part of the point-to-multipoint VC that is unique to a receiver with low resource requirements.

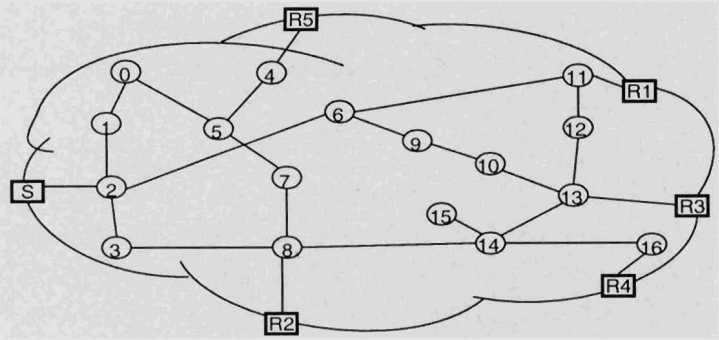


Figure 29: Example Network.

To show what results can be achieved with that simple algorithm consider the example network in Figure 29, which represents a model of the topology of the NSF backbone as of 1995 [JW97]. 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 not necessarily the case as the simple example in Figure 30 shows:

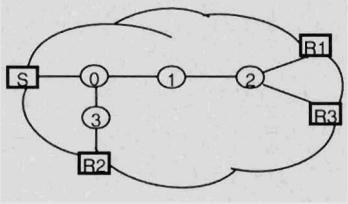


Figure 30: Example of an Unordered Optimal Partition.

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