Gateway Based Approach for Conducting Multiparty Multimedia Sessions over Heterogeneous Signaling Domains

Madhu Sudan and Nachum Shacham
Computer Science Laboratory, SRI International
{madhu,shacham}@csl.sri.com

Abstract

Emerging networking technologies tend to be introduced complete with their own protocols for resource management; for example, routing, resource reservation, and signaling. Each network offers a quality of service (QOS) interface, and applications take advantage of network signaling to demand and reserve predictive service. Whereas such interaction enhances the performance of applications that run on a specific network, it creates a challenge for applications that run over a heterogeneous infrastructure spanning multiple signaling domains. We present an approach for managing multiparty, multimedia sessions in a heterogeneous internetwork spanning multiple signaling domains and diverse link capacities. Participants utilize the native signaling on their respective domains and interact with participants on other domains through signaling gateways that bridge the domains and provide translation of signaling procedures and QOS semantics. Data streams are transmitted using a hierarchical representation, which allows participants to individually tune the quality of each stream they receive to their resources and interests. We present a particular design and implementation details for connecting ATM and IP signaling domains and discuss its extension to an arbitrary number of interconnected domains.

1 Introduction

The computer communication field has long been characterized by a rapid pace of introduction of new networking technologies bringing with them dramatic changes in transmission rates and functionality. Networks were characterized by heterogeneity in the lower levels of their architecture, namely, the physical media (e.g., fiber optics, wireless, twisted pair) or their switching techniques (packet or circuit switching). Internetwork protocols, most notably IP, provided the needed connectivity for applications that span multiple networks. The success of IP can be partially explained by its design that expects little from the underlying networks, by the fact that networks traditionally provided only basic transport service, and by the simplicity of the applications that were not equipped to utilize more advanced network services.

The recent emergence of new applications and networks has rendered the old network interconnection paradigm inadequate. Both applications and networks have evolved to the point where simple unreliable datagram service is no longer an adequate internetworking glue. Multimedia and real-time applications impose strict quality-of-service (QOS) requirements that must be supported by resource reservation or advanced buffer management by network switches. In anticipation of this class of applications, emerging network technologies are being introduced complete with protocols and user interfaces through which applications can negotiate QOS and have network resources reserved for their use. Typically, a *signaling protocol* is used by an application to communicate its QOS requirements to the network, and by the network to propagate this information to all the network entities (hosts, routers, switches) so that appropriate resources can be reserved. Thus, in addition to heterogeneity in the physical media and switching techniques, the current networks have introduced signaling heterogeneity as well. For instance, the Reservation Protocol (RSVP) [ZDE+93] and the User Network Interface (UNI) [For93] are the signaling protocols for the IP suite and the Asynchronous Transfer Mode (ATM) networks, respectively.

The diversity of signaling protocols restricts the interoperability among QOS-aware multimedia applications in a heterogeneous internetwork (a collection of networks using *different* signaling protocols). Communication between different *signaling domains*[1] in such an internetwork still remains a challenge. The traditional approach for achieving inter-operability is to impose *a uniform signaling* protocol throughout the internetwork irrespective of the underlying networking technology. For example, the existing approach in the current Internet is to use the IP protocol suite as the universal protocol. A considerable amount of research has been done in order to find efficient ways of implementing IP over different networking technologies, especially ATM [CSV96, Mil95, Lau94, PLM+95, Arm96]. With IP as the unifying framework, the only signaling protocol is RSVP [ZDE+93].

Although attractive, this approach is not trivial to implement. It involves address translation, translation of QOS parameters and translation of signaling messages between the various domains in order to facilitate communication between two end points separated by domain boundaries. This approach requires all the end points to use IP for all their communications and hence might lead to unnecessary translations from IP to ATM to IP, when direct ATM connectivity is possible between two end-points. It also prevents applications from taking advantage of more powerful native signaling when most of the communication is restricted to a single signaling domain.

We propose an alternative approach for achieving interoperability between applications that communicate across signaling domains. Our approach is based on performing *signaling translation* at the boundaries of two signaling domains. For intra-domain communication, the applications utilize the native signaling in their respective domains, and for inter-domain communication the interaction is through signaling *gateways* that bridge the domains and provide translation of signaling procedures and QOS semantics. This approach allows applications in different domains to communicate with each other *without* any changes to the existing signaling or resource reservation protocols. If communication involves end-points in the same domain, unnecessary translations are avoided.

[1] A domain here refers to a network where all the entities communicate using the *same* signaling protocol.

Teleconferencing applications provide a good example for the utility of the proposed approach. Numerous desktop teleconferencing applications provide similar service, session management and control, and the delivery of participants' video images. These applications, however, differ in the methods whereby their services are rendered. For example, RSVP-based and ATM-UNI 3.0 sessions are based on leaf- and source-initiated join, respectively. The two also differ in their connection type, namely, dynamic routing with soft state (RSVP) versus virtual circuit (ATM). The objective is to let such multiparty applications run unchanged on their respective domains and let them interoperate by bridging their differences in session setup, resource reservation, and data representation.

These examples demonstrate another facet of *Heterogeneous MultiCast* (HMC), which is a paradigm for operating multimedia multiparty sessions in environments where the participants differ in the constraints they impose on the session and receive different levels of service. The HMC concept, first suggested in [Sha92], originally addressed the issues of different transmission and reception capacities available to the session participants. Enhanced signaling and *hierarchical encoding* of the real-time streams enable participants to customize the fidelity of their reception to their individual needs and constraints. Hierarchical encoding is a contrast to a more traditional approach of *transcoding*. In the latter, inter-domain data is decoded and re-encoded at the domain boundaries to accommodate receivers with differing receiving capacities and networks with widely varying transmission capacities (like wireless and ATM networks). Layered streams overcome the problems associated with transcoding, namely lack of end-to-end security, increased delay in the internetwork and scalability. Hierarchical encoding also enables a receiver with resource constraints to receive a stream, albeit at lower quality (by subscribing to a few layers), instead of not receiving at all.

This paper describes the gateway based approach for realizing the HMC communication paradigm in a heterogeneous networked environment, consisting of IP and ATM networks. The concept of HMC and its implementation over two domains, namely IP and ATM, are discussed in Section 2. It also discusses the enhancements required in both ATM and RSVP signaling procedures for supporting HMC sessions. Signaling translation extends the HMC capability by enabling interoperability between signaling domains. As indicated above, the signaling gateway is the centerpiece for this architecture. The objectives of this gateway and its functions are discussed in Section 3. Section 4 presents the procedural aspects of the gateway and specifies the methods whereby HMC sessions are established and maintained across IP and ATM signaling domains. These concepts and procedures have been implemented and tested both in a laboratory environment and over a metropolitan-area internetwork. The implemented prototype is described in Section 5. In Section 6 we mention some issues and present a methodology for extending the architecture to encompass a large number of signaling domains that are interconnected in an arbitrary mesh. Section 7 contains concluding remarks.

Characteristic	ATM (UNI 3.0)	IP/RSVP
Orientation	Sender-driven	Receiver-driven
Support for QOS changes	Static QOS. Dynamic renegotiations are not allowed.	Dynamic QOS
Support for heterogeneous receivers	No. All receivers in a point-to-multipoint connection get the same QOS.	Yes
Group Reservations	Not supported	Not supported

Table 1: Comparison between ATM (UNI 3.0) and IP signaling (RSVP)

2 Heterogeneous Multicast

An HMC session consists of multiple sources, which offer real-time streams to a set of destinations. A source can offer more than one stream, and all the streams are assumed to be encoded hierarchically to facilitate viewing at multiple fidelity levels. Each destination signals to the network the streams it prefers to receive, and the desired fidelity level for each of the streams. Destinations can change their requests for streams anytime during the session, and the stream delivery to them changes accordingly.

The network switches attempt to deliver to each destination the requested streams. The HMC multidestination stream delivery must maintain efficient utilization of network resources, that is, a link carries a stream only if that stream is delivered to a destination downstream. Destinations' requests can conflict if there are insufficient resources to fully satisfy all of them. Such conflicts can be resolved fairly based on a number of criteria. We have found it useful to assign a monetary value to each request, that is, to let destinations bid for the stream they wish to receive. Bids represent a financial reward to a source if the requested stream is indeed delivered. Using this model, conflicts are resolved so as to maximize the total gain by the source [Sha95, SM94].

Teleconferencing applications with HMC services have been implemented and demonstrated over ATM and IP domains, which use different schemes for transmission and switching, namely, virtual connections (VCs) and datagrams, respectively. Both domains provide signaling with multicast support, but in both cases that support is not adequate for HMC. Table 1 summarizes the current capabilities of ATM and IP pertinent to HMC. For ATM we consider UNI 3.0, in which only a source can initiate a multicast connection, all branches of a multipoint VC receive the same QOS, and there is no provision for supporting a group of related multipoint VCs with different destination sets. UNI 4.0, which is still at a draft stage, includes mechanisms for a more flexible multicast, such as destination-initiated join and a group ID that does not vary across the network. Under IP multicast, in comparison, destinations control their connectivity, and request resources to be reserved through the RSVP signaling protocol. RSVP, however, does not presently include provisions for admissions of and reservations for groups of hierarchically organized streams.

In the rest of this discussion we describe implementations of HMC over ATM and IP and explain the enhancements in both domains' signaling procedures to support HMC.

2.1 HMC over ATM Network

Currently, ATM lacks support in two areas that are crucial for HMC sessions. First, ATM signaling does not allow heterogeneity among receivers in a multicast VC (all the receivers in a point-to-multipoint VC must get the same QOS). Second, ATM receivers cannot renegotiate VC QOS in mid-session. Given these constraints, we have chosen to implement HMC over ATM by transporting each layer of each stream over a separate multipoint VC. Thus, the fidelity at which a stream is received is determined by the number of layers delivered to that receiver. A receiver renegotiates its QOS by adding or deleting layers, that is, by connecting to or disconnecting from multipoint VCs representing the corresponding layers.

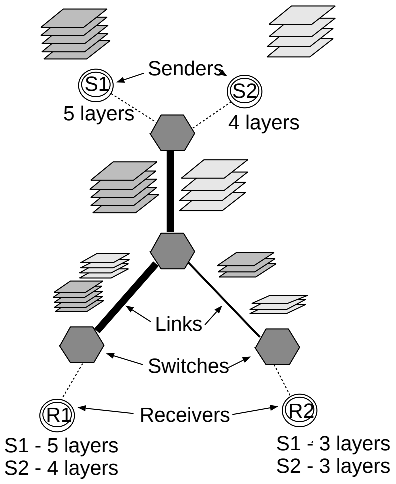
This approach was accomplished by the following enhancements to the ATM signaling[2] to support the transport of layered streams.

- Ability to open multiple VCs in a single round-trip of a signaling message, which shortens the time to establish a multi layer connection by or to a destination. Opening all VCs for the connection in a single signaling round-trip causes all those VCs to follow the same route, thereby facilitating admission and resource allocation decisions.
- All the VCs for a particular stream are related by a group identifier, which remains unchanged throughout the network and during the session. This group ID enables a destination to initiate a join to the session, and it is used by switches for admitting VCs according to their ranking in the stream hierarchy.
- Enhanced admission control algorithms for admitting connection requests from single and multiple participants. The details of the algorithms used can be found in [Sha95, SY96].

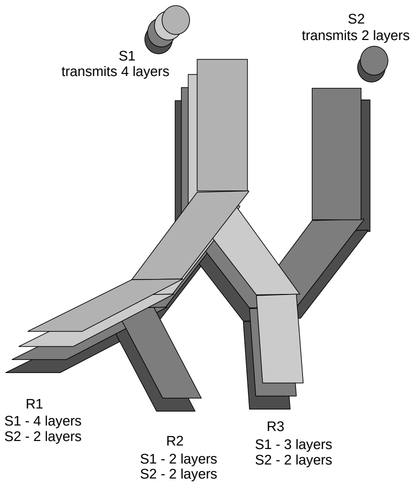
The main features of the HMC enhancements to the ATM signaling are summarized in Table 2.

Finally, in our HMC implementation, data is transported by using RTP [SCFJ96]. The RTP packets are encapsulated [FBFM95]as AAL protocol data units (PDUs). Data from the source to the receivers is carried by a distribution tree as shown in Figure 1(a). The distribution tree consists of as many point-to-multipoint VCs as the number of layers. Each stream (even if the streams are from the same source) has a separate distribution tree. If a certain participant wishes to receive a stream, it is added to the distribution tree by the network.

[2]The new UNI 4.0 signaling standard includes provisions that accomplish some of these goals (e.g.,leaf initiated join).



(a)



(b)

Figure 1: *HMC in ATM and IP domains. (a) The distribution tree from the source (S1 and S2) to the receivers (R1 and R2). The solid lines represent links connecting ATM switches. The number of layers carried on each link depend on the requirements of the receivers. (b) The multicast tree from the source (S1 and S2) to the receivers (R1, R2 and R3) depending on the number of layers desired by the receivers. Different shades represent different multicast addresses (and hence different layers).*

2.2 HMC over IP/RSVP Network

IP multicast [DC90] has received wide acceptance and is the standard method for multidestination distribution of messages and monolithic streams over the Internet. The simplest way to implement HMC over IP multicast is to assign different layers to different multicast addresses. According to this scheme, first proposed by Deering, destinations change their fidelity by joining or leaving multicast groups using IGMP messages. This approach is illustrated in Figure 1(b). In this scheme destinations are responsible for managing their connections and receiving layers according to their ranking in the hierarchy. Furthermore, under IP multicast all sources of the session transmit a particular layer on the same address and once a destination joins the session it receives all the corresponding streams. Consequently, to receive the second layer from a given source, a destination must receive the base layer from all sources. This, of course, adversely affects the bandwidth management aspects of HMC.

RSVP, as the Internet's signaling protocol, is used by destinations to select the sources from which they want to receive and to specify to the network switches the bandwidth needed for the corresponding streams [ZDE$^+$93]. However, RSVP currently lacks two aspects of support for layered streams, namely, (1) the ability to make *group reservations* and (2) the performing of *ordered* admission control. We suggest the following enhancements to IP signaling.

- A *group identifier* to relate all the multicast addresses carrying data for a layered stream
- Enhanced criteria for ordered admission control (described later)

Characteristic	ATM (for HMC)	IP/RSVP (for HMC)
Transport of layered stream	Each layer carried on different point-to-multipoint VC	Each layer carried on a separate multicast address
Receiver QOS	Aggregate QOS of all the received layers	Aggregate QOS of all the received layers
Orientation	Could be receiver or sender-driven	receiver-driven
Support for QOS changes	Dynamic QOS. Receivers can add or drop layers.	Dynamic QOS. Receivers can add or drop layers as well as change the QOS of individual layers.
Support for heterogeneous receivers	Yes. Different receivers can subscribe to different numbers of layers.	Yes

Table 2: Comparison between enhanced ATM and IP signaling

Table 2 summarizes the properties of the signaling in the IP domain to support layered streams. If a source transmits multiple streams, each stream is sent from a distinct UDP port. This allows the receivers to distinguish different streams from the same source. The IP address identifies the source, and the UDP port number identifies the stream sent by the source. Thus, this combination uniquely identifies any stream in the IP domain.

3 The Gateway: Objectives and Functionality

To enable transparent inter-domain communication the two domains are connected by a *session-level gateway* having two interfaces, one to the ATM and the other to the IP domain. The session participants on the ATM domain are aware of the ATM interface address of the gateway, and those on the IP domain know the IP interface address of the gateway. The gateway has the following objectives.

Data transmission: Inter-domain data is transmitted via the gateway to the participants in either domain. Hence the gateway acts as a *proxy source* (Figure 2) to a domain whenever the actual source is on a different domain.

Transparent operation: Although inter-domain communication involves the gateway, the receivers and senders are not aware of the domain boundary. They continue to follow the same procedures for joining and leaving a session on their respective networks, assuming that all the participants are in the same domain. The gateway makes the necessary translations to transparently facilitate inter-domain communication.

Efficient distribution: During the lifetime of a session, the receivers (on any domain) can dynamically adjust the number of layers received from the source. In the case of inter-domain traffic, the gateway tries to maintain a balance between the number of layers being transmitted from the actual source (l_t) and the number of layers being actually delivered (l_d) to the receivers in the other domain. For instance, in Figure 2, if the receivers R1 and R2 do not wish to receive the second layer, the gateway takes appropriate actions to prune the distribution tree from the source, in the ATM domain.

To achieve these objectives, the gateway must perform certain translations and maintain specific information about sessions. The gateway has the following functionalities.

Signaling translation: Signaling is used to create and establish an HMC session and to propagate QOS information in the internetwork. To enable interoperability, a signaling message (originating from *any* domain in the heterogeneous internetwork) should elicit the same behavior in both the domains. The gateway interprets these signaling messages from one domain and generates corresponding messages for the other domain. For example, the `PATH` and `RESV` and the `RSVP_ERROR` messages in the IP domain cause the gateway to generate the appropriate messages on the ATM domain and vice versa. The gateway also performs the translation of QOS semantics while generating the corresponding messages.

Data forwarding: Any inter-domain session, in turn, is composed of two intra-domain sessions (one on ATM and one on the IP network) with the gateway providing the needed interconnection. The gateway maintains appropriate mapping information between the sessions in the two domains and uses this to forward the inter-domain data from a sender in one domain to the correct set of receivers in the other domain.

Multiplexing/demultiplexing of data: Data crosses domain boundaries, through the gateway. If several senders in domain D1 are sending data to receivers in domain D2, the gateway acts as a *sink* for data coming from the sources in domain D1 and a *proxy source* for data going to receivers in domain D2 (see Figure 2). Since a *single* gateway represents *multiple* sources in domain D1 to receivers in D2, all the streams from different sources are multiplexed at the gateway. It provides enough demultiplexing information in the data such that the receivers can distinguish the actual senders when the data is received.

Data encapsulation and de-encapsulation: Data in the two domains arrives (and is transmitted) in different sizes and uses different transmission schemes. ATM data is transmitted in AAL PDUs and IP data is transmitted in the form of IP packets. In addition, the audio and video packets have encapsulation headers to enable the receivers to assemble and play them back correctly. The gateway receives data from respective domains and performs encapsulation or de-encapsulation such that it is suitable for transmission in the other domain.

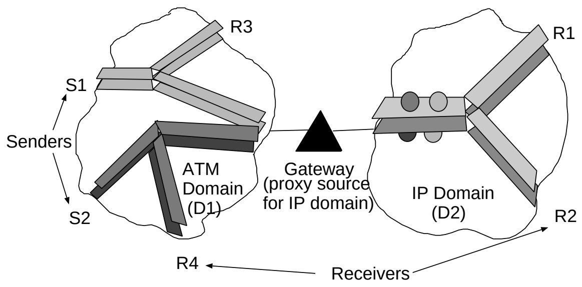


Figure 2: *Same layers of different streams (from S1 and S2) in the ATM domain, being multiplexed on to the same IP multicast address. Different shades in the ATM domain represent different VCs, and shades in the IP domain represent different multicast addresses.*

4 HMC Procedures for Inter-Domain Communication

In Section 2 we described the enhancements to ATM and IP signaling to enable HMC sessions in the individual domains. With such a signaling at our disposal, we now consider a joint session with participants scattered over ATM and IP domains. Here we describe the procedures followed by the participants to join or leave the session, and the actions taken by the gateway to enable interoperability.

4.1 Participating in a Session

To participate in a session, the participants should have (1) information about the session identity and (2) information about the addresses of participants so that they can request data from the source, invite each other to join the session, or do both.

4.1.1 Addressing

Since addressing schemes in IP and ATM domains are completely different, it does not make sense to provide receivers on the IP domains with the address of the source in the ATM domain and vice versa. Designing a common addressing scheme and mapping ATM (IP) addresses to IP (ATM) is a significant problem in itself, and a detailed discussion of a common addressing scheme is beyond the scope of this paper. We mention herein our approach to resolving this incompatibility among addresses. This approach is a subject of further research, but is based on the following.

All the participants are assigned *names* (or IDs), which are independent of the domain-specific addresses. The participants on either domain identify each other by their names instead of the addresses, when trying to initiate communication with each other. The names are finally mapped to addresses in the respective domains. This eliminates the need for propagating domain-specific addressing information to participants outside the domain. The gateway is configured to be aware of the mapping between the names and addresses of participants on either domain. Inter-domain requests for connection establishment, teardown and modification are received by the gateway, which translates the names (mentioned in the requests) into actual IP (or ATM) address and establishes communication between the participants.

All the participants in a session receive the following information in advance by an out-of-band mechanism: (1) start time of the session, (2) source IDs, (3) maximum number of layers in the transmission, (4) QOS for each layer, and (5) multicast addresses on which each layer is transmitted (for participants on IP only).

It is assumed that if there are k layers, they are ranked in the order $1...k$, where rank 1 is the lowest (base layer) and k is the highest. As explained below, the procedures for participating in an inter-domain session remain the same as an intra-domain session.

4.1.2 Joining a Session

Actions taken by the source: The source in the ATM domain waits for receivers to `connect` to it. The `connect` message from a receiver specifies the number of layers desired by the receiver. On receipt of such a request, the source tells the ATM network to create a multilayer (one for each layer), point-to-multipoint VC to the destination and sends an acknowledgment to the receiver. The `connect` could also be initiated by the source.

The source in the IP domain sends a `PATH` message (indicating the source traffic specifications) for each layer that it is transmitting on the corresponding multicast address (known in advance). A source generating multiple streams uses a different port number (UDP) for each stream. Thus, the combination of `<Source_IP_Address, Source_UDP_Port>` uniquely identifies the stream and the source.

Actions taken by the destinations: In the ATM domain, the receiver sends a `connect` message to the source specifying the number of layers desired and waits for an acknowledgment for the request. If there are not enough resources, the network reports the number of layers for which resources were reserved successfully. The receiver can then choose to leave the session (described later) or continue with the available number of layers. If the resources are available, the ATM network adds this receiver to the distribution tree from the source to the destination(s).

In the IP domain, any participant interested in receiving a particular layer joins the multicast group (corresponding to that layer) by sending an `IGMP_JOIN` message for that group address (known in advance). Thus, for receiving k layers, it joins all the multicast groups, for layers $1...k$, in that order. The receiver then waits for the `PATH` messages for the different layers. On receipt of the `PATH` messages, it sends `RESV` messages, one for each layer, that it wishes to receive, in the order of lowest- to highest-ranked layer (this restriction is not currently imposed by RSVP). The reservations are made using *fixed filter* style and specify the `<Source_IP_Address, IP_Multicast_Address, Source_UDP_Port>` for each layer. After sending the `RESV` messages, the receiver assumes[3] that the request succeeded, unless it receives a `RESV_ERROR` message. On receipt of such a message, it stops sending the refresh[4] `RESV` messages for all the layers higher than the layer for which `RESV_ERROR` was received. Then it leaves the multicast groups for which no `RESV` messages are being sent. Note that this assumes no changes in the admission control procedures in the IP domain (used by RSVP). Destinations are responsible for ordering the `RESV` requests.

4.1.3 Changing QOS in a Session

It is to be noted that each layer has a certain QOS associated with it. Hence, the receiver QOS is the sum[5] QOS of the individual layers received. Receiver QOS can be changed in two ways - either by changing the QOS of a received layer or by changing the number of layers received.

Because of restrictions imposed by the ATM standard[6] it is not possible to change the QOS of an individual layer (transported over a point-to-multipoint VC). Hence, in the ATM domain, the receivers change their QOS by sending a `modify` request, which specifies the layer to be added or dropped. If a layer is to be added to the set of received layers, this request results in the creation of a new point-to-multipoint VC from the source (if one does not exist) or causes the receiver to be added to an existing point-to-multipoint VC in the network. If a layer is to be dropped from the set of received layers, the corresponding branch of the point-to-multipoint VC (leading to this receiver) is deleted.

In the IP domain, the receivers can change either the QOS of an individual layer or change the number of layers received. To change the number of layers received, the receivers join or leave the corresponding multicast group. To change the QOS of an individual layer, the receivers start sending `RESV` messages which indicate the desired QOS. Soft state in RSVP helps to provide dynamic QOS.

4.1.4 Leaving a Session

In the ATM domain, a receiver or sender can initiate a `disconnect` to leave a session. This releases all the resources associated with all the layers that were being delivered to the receiver. Specifically, it closes all the point-to-multipoint VCs, in the distribution tree, which were being used to deliver the received layers. In the IP domain, a receiver sends a `RESV_TEAR` message and then leaves the multicast groups corresponding to the layers.

Table 3 summarizes the actions by the sources and destinations.

[3] Since RSVP does not have any confirmation mechanism

[4] RSVP maintains soft state and hence sends refresh messages periodically

[5] We consider only bandwidth to be the measure of QOS for the purpose of this document. If there are other parameters to specify QOS, the receiver QOS may not be the sum but some *aggregate* of the individual QOS.

[6] We assume here UNI 3.x

Operations	ATM (HMC signaling)		IP/RSVP (HMC signaling)	
	Source	Receiver	Source	Receiver
Join	Wait for receiver to `connect`	Send a `connect` message specifying the number of layers desired	Send a `PATH` message for each layer	Join the desired multicast groups. Send a `RESV` message for each group.
QOS modifications	-	Send a `modify` request. It indicates the layers to be added or dropped.	-	Join or Leave the multicast groups to increase or decrease the number of layers. Send modified `RESV` messages to change the QOS of individual layers.
Leave	Send a `disconnect` message	Send a `disconnect` message.	Stop sending `PATH` messages	Send a `RESV_TEAR` for all the groups. Leave the multicast groups.

Table 3: Actions taken by the source and destinations in both domains.

4.2 Gateway Procedures

To maintain *transparent* inter-domain connectivity, the gateway must represent the actual sources to the receivers whenever the sources and destinations are separated by a domain boundary. Thus, it represents an IP source to the receivers in the ATM domain and vice versa. Note that all the receivers of a particular stream are organized into a distribution tree (see Section 2) in both domains. The distribution trees in the ATM domain consists of a number of VCs (one for each layer) and are distinct (no two streams share the same VC, and hence the same distribution tree), whereas they overlap in the IP domain (since a particular layer from all the streams is transmitted on the same multicast address). Figure 3(a) shows two IP sources, each transmitting a single layered stream to multiple receivers in the ATM domain. The gateway is a *dummy receiver*[7] for the IP domain and is the root of the distribution tree in the ATM domain. Since there are two streams in the IP domain, there are two distribution trees in the ATM domain. All the receivers in the ATM domain who wish to receive a stream become a part of the corresponding distribution tree. In this case, the gateway acts as a *proxy source* to the receivers in the ATM domain. Figure 3(b) shows two ATM sources sending data streams (one per source) to the receivers in the IP domain. In this case, the gateway is a part of both the distribution trees in the ATM domain and acts as a proxy source for the receivers in the IP domain.

While acting as a *proxy source* to either domain, the gateway must perform all the functions of an actual

[7]We call it "dummy" since the gateway does not run any application that needs the data

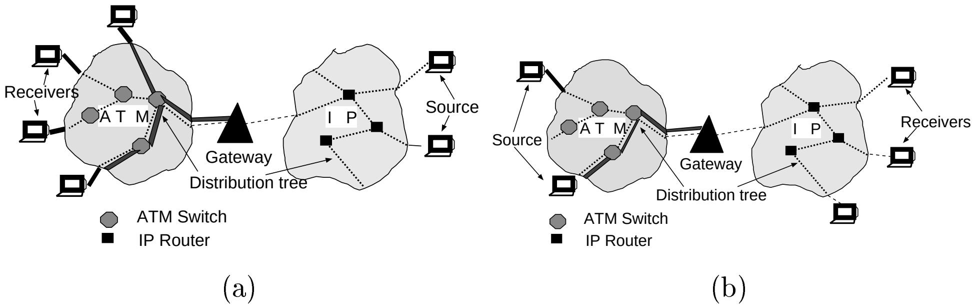


Figure 3: *Flow of data (for one layer) (a) From the IP domain to the ATM domain. (b) From the ATM domain to the IP domain. The figures show the distribution trees in both domains.*

sender in the respective domains. This allows the inter-domain communication to proceed as if the participants were all on the same domain. In the following, we describe information maintained by the gateway and the actions taken in order to carry out the functions required to achieve interoperability.

4.2.1 Signaling Translation

Signaling messages are translated at the gateway when data flows in either direction (IP to ATM or ATM to IP).

Source on IP domain and receivers on ATM domain:

The gateway acts as a receiver in the IP domain and a sender in the ATM domain. For each stream in the IP domain, a multipoint VC is created in the ATM domain. The gateway completes the following sequence of events:

- To receive all the layers from any IP source, it sends out `IGMP_JOIN` requests for all the multicast addresses corresponding to the different layers.
- It receives the `PATH` message for different layers from the IP source. The information about the *traffic specifications* of the IP source is made available to the receivers in the ATM domain by some out-of-band mechanism.
- It waits for the `connect` message[8] from a receiver in the ATM domain. The `connect` message specifies the number of layers the receiver wishes to receive.
- On receiving a `connect` message, it sends out a *separate* `RESV` message for *each* of the layers that the receivers wish to receive (and for which a `PATH` message has been received). The messages are sent in order, from lowest-ranked layer to the highest-ranked layer. If the gateway receives

[8]Note that the `connect` message from the first receiver helps in creating the distribution tree in the ATM domain. All successive receivers in the ATM domain are added to this distribution tree by the network. The gateway does not necessarily have to know about all the receivers.

a `RESV_ERROR` message, it sends a `modify` request to the ATM network to tear down the VCs corresponding to the layers for which reservations failed in the IP domain.

- If there are no errors, it implies that the reservations were successful. Subsequently, the gateway records the mapping between the sender and the receivers in the *IP-to-ATM translation table* (described later). The transmitted stream (and hence the sender) is identified by the `<source_IP_address, source_UDP_port>` (received as a part of the `PATH` message). The receivers are identified by the ID of the ATM distribution tree, which is obtained after the gateway receives the first `connect` message from the ATM domain. This table is used for data forwarding.

- Once the session is established, the gateway tries to maintain a balance between the layers received from the IP domain and the layers being delivered to the receivers in the ATM domain. If there are no receivers in the ATM domain receiving layer i, then it sends a `RESV_TEAR` for this layer on the IP domain to release the reservations.

Source on ATM domain and receivers on IP domain:

The gateway completes the following sequence of events:

- It receives the information about the *traffic specifications* of the source in the ATM domain by an out-of-band mechanism and sends out separate `PATH` messages (one for each layer) to the receivers on the IP domain. The messages are sent in increasing order of the ranking of the layers. This step is repeated [9] for all the streams sent from the ATM domain to IP domain. To the IP domain, the gateway acts as the source of these streams. So that the receivers can distinguish the various streams, the `PATH` messages for different streams are sent from a different UDP port.

- It waits for an IP receiver to send a `RESV` message indicating its interest in a particular layer. Since `RESV` messages may be lost or misordered, it waits for a *refresh period* to determine the lowest i layers (in sequence) for which the receiver could have made a request.

- At the end of the refresh period, it sends a `connect` (indicating the requested number of layers) message to all the identified sources in the ATM domain.

- If a reservation request is rejected by ATM for a given layer, the gateway sends a `RESV_ERROR` for that layer and all others of higher ranking for that stream. The `RESV_ERROR` message propagates downstream in the distribution tree in the IP domain.

- If there are no errors, the gateway records the mapping between the ATM source (ID of the distribution tree) and the IP receivers (UDP port and the multicast address of the layers) in the *ATM-to-IP translation table* (described later). This is used for data forwarding.

[9] If x streams are sent from the ATM domain to IP domain, the gateway sends $x * l$ `PATH` messages, where l is the number of layers in each stream.

- It tries to maintain a balance between the layers received in the ATM domain and the layers delivered in the IP domain. If there are no receivers in the IP domain receiving layer i, then it sends a `modify` in the ATM domain to tear down the corresponding VCs.

For data going from the ATM domain to the IP domain, all the packets for layer i are multiplexed by the gateway on to the same multicast address. Different UDP port numbers for each stream allow the receivers to differentiate among the streams.

The gateway must also translate the QOS parameters in addition to translating the signaling messages. Currently, bandwidth is assumed to be the only measure of QOS and hence the translation is relatively simple.

4.2.2 Multiplexing and Demultiplexing

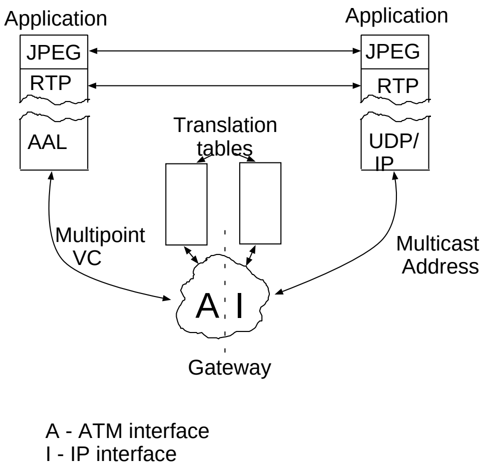
For data going from ATM domain to IP domain, the gateway acts as a sink for all the streams coming from the ATM domain and a source for all the streams going to the IP domain. It multiplexes a particular layer from all the streams to the same IP multicast address. For data going from the IP domain to the ATM domain, packets belonging to a particular layer from *all* the streams are received on the same multicast address. Hence it demultiplexes the data to identify the actual senders in the IP domain before transmitted it on the ATM domain. Any source generating a layered stream uses the *same* UDP port number to send all layers of the stream. An individual layer belonging to a particular stream can be distinguished by a combination of (1) `IP_Multicast_Address`, which identifies the layer received and (2) `<Source_IP_Address, UDP_Port>`, which identifies the stream and the source.

4.2.3 Data Forwarding

To forward the data streams from the senders to the correct set of receivers, the gateway must identify the sender, the stream that the sender is sending (since a sender might be sending multiple streams), and the correct set of receivers in the other domain. To achieve this functionality, the gateway uses the following tables, which are set up as part of the signaling process (as described in Section 4.2.1).

Layer-to-multicast address mapping: This table is used by the gateway to determine which layer corresponds to which IP multicast address. It is set up at the configuration time. The number of entries correspond to the maximum number of layers transmitted by any source. Each entry contains (1) a layer number, and (2) the multicast address of the layer.

IP-to-ATM translation table: This table is used in forwarding data from the IP domain to the ATM domain. Since the gateway is the root of the distribution tree in the ATM domain, it suffices to know the identity of the distribution tree rather than knowing all the individual recipients of the stream in the ATM domain. The number of entries in this table corresponds to the number of active



(a)

	A T M	IP/RSVP
Stream Identification	VC Number. Separate VC, Separate Streams	Source_IP_Address, Source_UDP_Port
Layer Identification	VC Number. Separate VC, Separate Layers	IP_Multicast_Address

(b)

Figure 4: *(a) Translation using the gateway and (b) source and layer identification in each domain.*

streams in the IP domain. Each entry contains (1) IP stream identification (source IP address, source UDP port number) and (2) ATM destination identification (ID of the ATM distribution tree).

When the gateway receives data from an IP source, it identifies the IP source and the stream, using the demultiplexing mechanism explained earlier, looks for the ATM distribution tree to which the stream is to be forwarded in the table, and sends data over the relevant distribution tree (which translates to a group of VCs).

ATM-to-IP translation table: This table is used to forward data from the ATM domain to the IP domain. The streams (and the senders) are in the ATM domain can be determined through the identity of the distribution tree (group of VCs). Receivers on the IP domain are (collectively) identified by the IP multicast address corresponding to each layer. Note that the gateway acts as a proxy source (representing multiple ATM sources) to the IP domain, and hence it uses different UDP ports for transmitting each stream. The number of entries in the table correspond to the number of active streams in the ATM domain. Each entry contains (1) ATM stream identification (ID of the ATM distribution tree) and (2) sending UDP port (one port per ATM stream).

When the gateway receives data from an ATM source, it identifies the stream (since ATM is connection oriented, we do not need any demultiplexing mechanisms; the VC number determines the source, stream, and layer), looks up the corresponding entry in the table to determine the sending port (UDP), and sends data on the multicast address (determined from the layer-to-multicast address map) and the port number (see Figure 4).

4.2.4 Encapsulation and De-encapsulation

All the audio and video data is encapsulated end-to-end in RTP [SCFJ96] format. The video packets are in the format as described in [FBFM95], and the audio packets use the format as suggested in [Sch96]. The packet size is chosen not to exceed the MTU (Maximum Transmission Unit) on the path. Each RTP packet is sent over the ATM domain as an AAL PDU and over the IP domain as a single UDP/IP packet.

4.3 Discussion

We have described the procedures that are followed by the sources and destinations in both the ATM and the IP domains to transmit and receive layered streams. Because of transmission of layered streams, the sources and destinations must observe some additional constraints, and the network must provide proper support. In discussing the rationale behind observing those constraints at the sources and destinations and at the network, we describe what could go wrong and what kind of preventive and corrective actions must be taken.

4.3.1 IP Domain

It was mentioned that the receiver sends `RESV` messages for multiple layers in the order of lowest to highest. Problems will develop if this constraint is not observed. Assume that a typical receiver receives `PATH` messages for layers 1...5 in the order 1, 4 and 5 (2 and 3 get lost) and sends `RESV` messages in the same order. The `PATH` messages for layers 2 and 3 arrive later. It is possible that the reservations for 1, 4 and 5 succeed, but those for 2 and 3 are denied because resources are not available. The receiver would now receive layers 1, 4 and 5 and would not be able to construct the layered stream because of the missing layers (we assume that the stream can be reconstructed only if layer $i-1$ is received before layer i).

Now let us assume that the receiver does send out the `RESV` messages in order, but the network does not guarantee that resources would be reserved in order. It is possible for the `RESV` messages to get misordered and to arrive at the routers out of sequence. Assume that the `RESV` messages were sent for layers 1...5 and that they arrive in the order 1, 4, 5, 2 and 3 at some intermediate router. The router makes reservations for layers 1, 4 and 5, but rejects the reservation request for layers 2 and 3. This causes the same problem as that described in the previous paragraph.

Hence, the receivers, as well as the network, must *order* the `RESV` requests for different layers for the successful reception of a layered stream at the receiver. In addition to making reservations, the routers also *forward* the request upstream. It is to be noted that forwarding the requests for each layer independently (out of order) would result in problems similar to those already explained.

If the routers take the following actions it will improve performance beyond what is achievable by destination-only ordering:

1. A router *admits* a reservation request (indicated by the tuple (`Source_IP_Address`, `IP_Multicast_Address`, `Source_UDP_Port`) if (1) resources for all the layers of lower rank from the same source have been reserved and (2) resources are sufficient to admit the current reservation requests.

   Failure in the first condition implies that either the requests for admission of lower layers were denied because of unavailable resources, or the `RESV` messages for the lower layers were lost in the network. Hence, the RSVP router waits for a *refresh period*[10]. At the end of the refresh interval

   - All the requests for which the first condition is satisfied are processed for resource reservation (resources are reserved for the lowest m layers that can be admitted).
   - If the first condition is still false, the current reservation request is ignored (it might be a duplicate request floating around in the network).

   If the first condition is true, but the second condition is false, the router sends back an `RSVP_ERROR` message downstream, which indicates that *only* the current request was denied.

2. A reservation for layer i is *forwarded* only if all the layers ($\leq i$) have been admitted by the router.

An IP network provides a *best effort* service, and hence the signaling messages might get lost or misordered. If a signaling message gets lost, it is likely that the message would arrive later, since the messages are sent at every refresh interval. Hence, the loss of a message can be interpreted as misordering, and the conditions we have outlined indicate the actions to be taken.

4.3.2 ATM Domain

An ATM network provides a connection-oriented service, and hence the problems like misordering (which are present in IP) go away. However, the hierarchical representation of a stream requires certain support from the network and also imposes some restrictions.

The reservation requests for all the layers are carried in a single message (unlike in IP) and hence the network does not have to implement mechanisms for dealing with out-of-order requests. This makes the admission control process a lot easier. However, the network should observe the following restriction: A layer i cannot be admitted unless layer $i-1$ has been admitted. A detailed analysis of admission control criteria can be found in [SY96].

[10] The `RESV` messages are sent once every refresh period.

It is preferable that all the layers of a stream follow the same path in the network. If the layers arrive at the destination via different paths, synchronization problems will occur at the receiver. In addition, unnecessary delay might be introduced because of the varying delays of the various paths over which the layers are transmitted.

To provide receivers with the maximum QOS, the network should be able to calculate the *maximum bandwidth path* to each destination. This calculation helps the sources to allocate bandwidths[11] to the layers so as to maximize the number of layers received by a destination. This results in the best possible fidelity in the reception.

5 Implementation

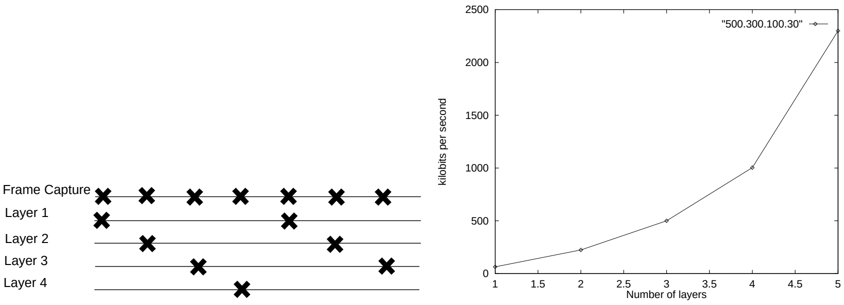
We have implemented a research prototype of the gateway and the HMC systems running on both the ATM and IP domains in our laboratory. This section describes our setup in both the domains. The focus was on using the gateway to interoperate between the tools written for individual domains.

Hierarchical encoding was emulated by varying the JPEG quantization factor for each captured frame and through temporal resolution by layer. Each layer is actually a JPEG encoded image at a different quantization factor. Figure 5(a) shows the transmission of data over all layers. Higher quantization factors implies lower amount of data and bandwidth requirement and poorer resolution of the image. The base layer (layer 1 in this case) has the highest quantization factor, producing the blockiest looking image. The quantization factors decrease with each subsequent layer and the resolution of the image is improved. Figure 5(b) shows the variation in aggregate bit rates for a sample set of quantization factors for different layers. The rates pertain to a 320 x 240 size image, using parallax boards for video capture and display, with quantization factors of 500,300,100 and 30 applied to the first, second, third and fourth layer of video, respectively.

We were not able to acquire system level boards that could do hierarchical encoding/decoding of real-time data at a reasonably fast rate. The software implementations (to the best of our knowledge) of hierarchical encoding/decoding techniques tend to be very computationally intensive, resulting in very low frame rates, thus making them unsuitable for teleconferencing applications. The purpose of hierarchical encoding/decoding service is to generate data with varying bit rates. Although our technique is not truly hierarchical, it utilizes the existing hardware in such a manner that hierarchical emulation is noticeable and thus it provides a temporary solution to test the concept.

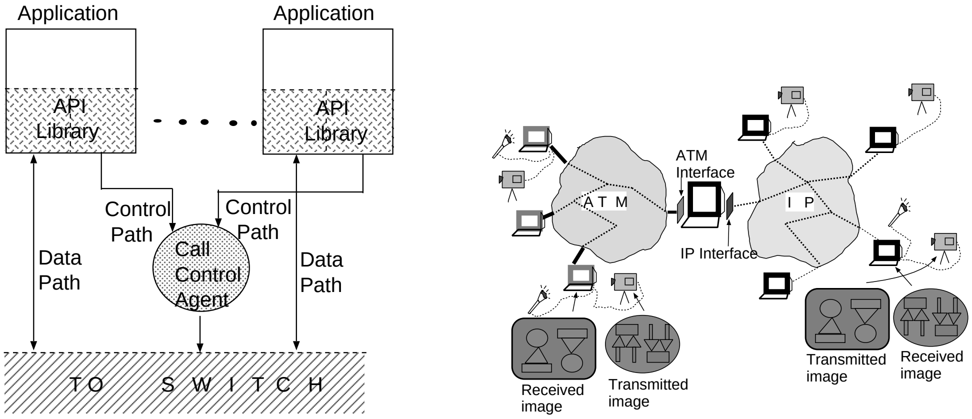
We developed our own audio/video conferencing tool to run on native ATM using the enhanced ATM signaling we have described. The tool provides the participants the capabilities to join or leave the session or modify the number of layers received, during mid-session. The switches in the ATM

[11] Note that this assumes the received bandwidth to be the only measure of the received QOS. If delay is the measure of quality (lower delay implies higher quality), the network must choose the minimum delay path instead.



(a) (b)

Figure 5: *(a) Hierarchical Data Transmission (b) Variation in bit rates for 320 x 240 size video images*



(a) (b)

Figure 6: *(a) Host architecture in the ATM domain and (b) Network testbed used for conducting an HMC session over IP and ATM Domains.*

network were enhanced to implement the HMC signaling (which is different from the standard ATM signaling [For93]) as explained in Section 2. Figure 6(a) shows the host architecture in the ATM domain. The applications use the API to make the open, close, and modify calls. The API is divided into a library and a Call Control Agent (CCA). The CCA implements the enhanced ATM signaling and handles the control messages (establishment, teardown of connections) from the API library. There is one CCA running per host. All the applications running on the host use the same CCA. Note that only the control messages go though the CCA. All the data transport takes place directly without the CCA as shown in Figure 6(a).

In the IP domain, we used the publicly available audio (VAT) and video (VIC) tools, and we used a multicast network to conduct HMC sessions. The tools were modified to receive data on multiple multicast addresses (and hence receive multiple layers). The audio and video tools in both the domains communicated using the gateway (see Figure 6(b)).

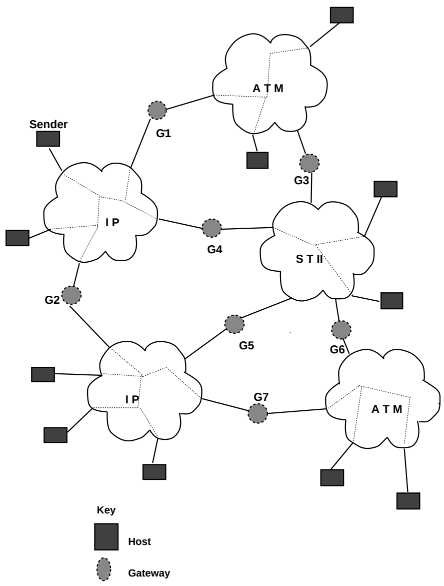


Figure 7: *An Internetwork consisting of an arbitrary number of domains connected by gateways.*

6 Issues

The design that has been described takes into account a single IP domain connected to a single ATM domain. If multiple domains are connected by gateways (see Figure 7) several other problems arise. We discuss how to extend the current architecture to multiple domains and some other issues that need to be addressed.

Extending the architecture to multiple domains: Figure 7 shows an example internetwork consisting of a number of domains connected by gateways with sources and receivers scattered across different signaling domains.

In such a situation, the gateways must be organized into a tree (spanning multiple domains) before data transmission can begin. The tree has the following properties: (1) It connects the sender to all the receivers in different domains and (2) It is dynamically reconfigurable depending on the changing participant population.

Since the tree spans multiple domains, all the gateways communicate using a *Meta Signaling Protocol* (MSP). We call it "Meta" signaling protocol, since it tells the gateways *when* and *how* to use the the native signaling in their respective domains. Because of the heterogeneity of domains, it is possible that two gateways (for instance, gateways G1 and G6 in Figure 7) in the internetwork do not understand each other, since they are connected to two different domains. However, G1 and G6 can communicate via G3, since G3 can understand the signaling protocol in both domains. The protocol details are a subject of further research.

The following issues are also related to conducting a session and are under further study.

Session control: To advertise a session in the internetwork, a session directory service is needed, which can be used by the participants in different domains to find information about a session such as the offered traffic by the sources and the participant population. The current implementation uses an out-of-band mechanism to deliver this information.

Route computation: To construct a tree of gateways in the internetwork, an appropriate *route* must be selected to the destinations. Route selection must be done based on certain metrics, such as load on the gateways, available resources in the possible paths, and hop count. The gateways must communicate among themselves and exchange appropriate information that would help them to select a route.

QOS mapping between the different domains: QOS is specified by different parameters in different domains. Hence, we need appropriate mapping and translation between these parameters so that the semantics of the actual QOS desired by the receiver or sender is preserved. This paper assumes that received bandwidth is the only parameter for determining the QOS of a receiver. If other parameters (delay variation, loss) are considered, then efficient QOS translation becomes more complex. When there are multiple domains between the sender and the receiver, as illustrated in Figure 7, an end-to-end flow would comprise individual flows across each domain. To establish these individual flows, the QOS parameters for individual domains must be derived from the end-to-end parameters. For instance, the delay parameter for an individual domain would be a fraction of the end-to-end delay parameter.

Scalability: In order to improve scalability of the gateway based approach, it is preferable to have multiple gateways between two domains such that each gateway handles only a fraction of the total inter-domain traffic. We are investigating into the criteria of selecting a particular gateway and various ways of load sharing between the gateways.

7 Conclusion

This paper proposed a design for conducting multimedia sessions in a heterogeneous internetwork using hierarchically encoded data. Although we have discussed in detail only two domains, the idea is extendible to more than two domains (as described in Section 6). The concepts that are the key to such a design are

- Use of the **native signaling** capabilities of the individual domains for intra-domain interaction and use of the gateway for inter-domain interaction.

- Use of the gateway to achieve interoperability. It performs *signaling translation* and acts as a *proxy source* to represent the actual source in a different domain. It also maintains *efficient distribution of data* by maintaining a *balance* between the flows in different domains.

The design allows applications in different domains to take advantage of their native signaling as well as to inter-operate with similar applications in other domains. In addition, the use of hierarchical data causes the communication in either domain to be unaffected by the resource constraints of the other domain. For instance, if the IP domain cannot support data transmission and reception at rates comparable to the ATM network, then receivers on the IP domain can subscribe to lesser number of layers **without** affecting the reception of the receivers in the ATM domain who can receive streams at their full quality. Networks are required to provide some additional support for the the transport of layered streams as compared to traditional monolithic streams. ATM signaling and IP signaling (RSVP) are still evolving, and we suggested the enhancements to be made to both the signaling domains to transport a layered stream. The requirements of the gateway and the procedures to achieve desired translation between both domains were described.

The current implementation proves the feasibility of a gateway based approach for realizing the HMC communication paradigm in a heterogeneous internetwork. This work is a part of an ongoing research and certain aspects (as mentioned in Section 6) need further attention. These issues will be discussed and analyzed in a forthcoming paper.

8 Acknowledgments

This work was supported partly by the U.S Naval Surface Warfare Center under contract number N00174-92-C-0129 and partly by the U.S Army Communications-Electronics Command under contract number DAAB07-90-D-A037. We would like to acknowledge Don Hoffman and Michael Speer of Sun Microsystems for their fruitful discussions about translation procedures and modifying the audio and video tools (VIC and VAT) for testing purposes. We would also like to thank Michael Brown and Danny Lee of SRI International, who were involved in the development and testing of the HMC software.

References

[Arm96] Grenville Armitage. Suport for Multicast over UNI 3.0/3.1 based ATM Networks. *Work in Progress*, February 1996.

[CSV96] R.G Cole, D. H. Shur, and C. Villamizar. IP over ATM: A Framework document. *Work in Progress*, February 1996.

[DC90] Stephen E. Deering and David R. Cheriton. Multicast Routing in Datagram Internetworks and Extended LANs. *ACM Transactions on Computer Systems*, 8(2):85–110, May 1990.

[FBFM95] W. Fenner, L. Berc, R. Fredrick, and S. McCane. RTP Encapsulation of JPEG-compressed video. *Work in progress*, November 1995.

[For93] ATM Forum. *ATM User Network Interface Specification, Version 3.0.* Prentice Hall, 1993.

[Lau94] M. Laubach. Classical ip and arp over atm. *RFC 1577*, Jan 1994.

[Mil95] Walter Milliken. IP Multicasting over ATM: System Artitecture Issues. *Work in Progress*, July 1995.

[PLM+95] M. Perez, F. Liaw, A Mankin, E. Hoffman, D. Gossman, and A. Malis. ATM Signaling Support for IP over ATM . *RFC 1755*, February 1995.

[SCFJ96] H. Schulzrinne, S. Casner, R. Fredrick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. *RFC 1889*, January 1996.

[Sch96] Henning Schulzrinne. RTP Profiles for Audio and Video with Minimal Control. *RFC 1890*, January 1996.

[Sha92] Nachum Shacham. Multipoint Communication by Hierarchically Encoded Data. In *Proceedings of IEEE INFOCOM*, pages 2107–2114, 1992.

[Sha95] Nachum Shacham. Preemption Based Admission Control in Multimedia Multiparty Communications. In *Proceedings of IEEE INFOCOM*, pages 827–834, 1995.

[SLB94] Nachum Shacham, Danny Lee, and Michael Brown. Real Time Traffic Distribution in Heterogeneous Environments. Technical report, SRI International, April 1994.

[SM94] Nachum Shacham and James S. Meditch. An Algorithm for Optimal Multicast of Multimedia Streams. In *Proceedings of IEEE INFOCOM*, 1994.

[SY96] Nachum Shacham and Hidetoshi Yokota. Admission Control Algorithms for Multicast Sessions with Multiple Streams. In *Network and Operating Systems Support for Digital Audio and Video*, 1996.

[ZDE+93] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala. RSVP: A New Resource Reservation Protocol. *IEEE Network Magazine*, 7(5), September 1993.