IP Multicasting over ATM: The Multicube Approach

H. Schulzrinne, M. Smirnov, R. Roth, A. Wolisz
GMD FOKUS, Hardenbergplatz 2, Berlin 10623, Germany
Tel.: +49 30 25499200 Fax.: +49 30 25499202

Abstract. This paper gives practical information along with theoretical background on Multicube project design decisions for scalable and reliable multicast service facilitating multimedia interworking of real users over the European ATM pilot. IP multicast mapping is proposed to be performed with the use of a minimal address resolution server serving a special multicast LIS within the ATM LAN. The feasibility of this choice is proved partly by practical experiment partly via simulation. Data applications for CSCW scenarios depend on a reliable information transfer. The Multicube project selected Xy, an X application sharing tool, to study the reliable multicast services provided by the protocols MTP, RMP and SRM.

1. Introduction

How should applications use the multipoint functionality offered by ATM? Interoperability across internetworks and the use of existing applications suggests the use of IP multicast over ATM. This, however, requires additional mappings and address resolutions. The paper discusses alternatives and reasons behind the current decision for the Multicube project [1]. The objective of Multicube is to develop, test and validate an ATM-based multipoint infrastructure for the support of CSCW applications. It is a trial-oriented project with the participation of end users from the automotive and aerospace industry. The majority of their multicast applications are IP based.

IP multicast is maturing fast. It currently supports teleconferencing, teleteaching, virtual meetings. It is involving more complex CSCW applications spread around the globe, like distributed simulation and collaborative CAD. The underlying internetwork tends to migrate towards gigabit technologies deploying ATM local and wide area networks. At present, however, ATM connectivity is concentrated mainly within user premises. Consequently, these islands are treated as Logical IP Subnetworks (LIS). One of the main differences with regard to multicasting is in different addressing schemes in LANs with native broadcast capabilities and ATM networks without this service. The addressing mechanisms in the Internet and, generally, in IP subnets make use of common flat availability of any pair of stations within the IP subnet address space (all hubs, bridges, switches, etc. are invisible). This feature causes no serious problems for address resolution tasks within legacy IP subnets such as Ethernet. Address Resolution Protocol (ARP) queries are broadcast, an operation which is cheap and natural for these networks. On the contrary, in the ATM environment, switches play a more visible role. ATM service model is connection oriented, broadcast is not a natural operation and, therefore, address resolution grows to become one of the major problems for IP multicast over ATM. Hence, ATM is an example of the so called NBMA (Non-Broadcast Multiple Access). IP multicast uses a special multicast address, derived from a reserved Class D Internet address space. A host sends its multicast flow to this address and subscribes itself to this address at the nearest multicast router to receive the flow. Operation of joining the multicast group is receiver initiated. NBMA multicast is to be organised via explicit connection establishment from sender(s) to receivers. A special kind of point-to-multipoint connection is specified within the UNI 3.1 signalling for multicast. However, joining the multicast group in NBMA is possible only via the root of the point-to-multipoint connection, i.e. the root has to know all NSAP addresses for all receivers.

The Multicube design approach tries to offer a seamless internetworking across private and public domains in a heterogeneous environment. Its strategy allows a smooth migration of existing applications and prevents application development to be locked onto a particular technology. Current understanding of the main project challenge is to achieve several levels of interoperability: between automotive and aerospace users and network designers, between different applications (across hardware and software), between IP and ATM styles of addressing, resource reservation, session management, multicasting, and, finally, between signalling stacks of private and public ATM networks.

The paper concentrates on design alternatives and their evaluation aiming at making general feasibility recommendations in the following areas.

1. *Reliable multicast for X application sharing*. IP multicasting offers only an unreliable datagram delivery. Transport protocols add the reliability needed by application sharing. TCP provides reliability in the case of unicast communication, but cannot be extended for group communication. Multicube is exploring alternative proposals for reliable multicast transport protocols, comparing them with respect to performance and scalability. As the required reliability services depend much on the particular application, Multicube has chosen to concentrate on X application sharing in its experiments. This section of the paper gives an overview of the methods selected in the project and experiments planned.

2. *Mapping of IP multicast to ATM point-to-multipoint connections*. This effort is intended to propose and validate efficient mechanism for IP multicast applications over ATM LANs interconnected via public ATM network. We propose a minimal (as compared to G. Armitage's MARS [2]) multicast address resolution server (mARS). mARS supports multipoint-to-multipoint connections and leaf initiated join for UNI 3.1 and 4.0. Its extension supports different QoS levels for the same multicast group. Today’s architectural alternatives are examined with the use of simulation for different scenarios of address resolution and multicast in ATM LANs.

3. *Implementation issues*. Several alternatives for the implementation of IP multicast over ATM were examined including FORE SPANS signalling (with broadcast ARP), FORE UNI 3.0 signalling (with Classical IP over ATM ARP), Vendor Independent Network Control Entity (VINCE) by ARPA and NRL, and MARS using Q.port (Portable ATM Signalling Software) by Bellcore. Comparison of existing alternatives is made in terms of feasibility, practical availability of software components, their conformance to existing ITU-T and ATM Forum standards, interoperability and generality.

2. Reliable multicast for X application sharing

Besides audio/video services that can be run over an unreliable IP multicasting service, there are other CSCW applications depending on a reliable information transfer. So issues of reliable multicast become relevant to the Multicube project. The main focus for reliable multicast will lie on questions of scalability. Different solutions will be compared for efficiency in handling larger user groups distributed over a wide area network. Since reliability requirements depend much on the particular application under consideration, Multicube will concentrate on an appropriate application and study the behaviour of several reliable multicast protocols in their capabilities of providing support.

The Multicube demonstration example is an X application sharing tool that will allow to contrast TCP based unicast with the multicast service provided by the protocols MTP, RMP and SRM. The experiments will include mixed traffic scenarios with parallel real-time audio / video transmission, allowing to study integration with resource reservation.

2.1 X Application Sharing for Multicube

With X application sharing tools, single-user X Window applications can readily be introduced into group conferences. Under the X window system a client application can be run on a different system than the X server which controls display and input devices. Communication between X client and X server is achieved by message exchange using the X protocol. An X application sharing tool intercepts these messages and re-distributes them among several users.

The original X Window system, however, has not been designed for a group environment, hence the development of an application sharing tool becomes non-trivial ([3, 4]). The main difficulty arises from the fact that X messages may contain *atoms*, which are unique identifiers private to the X server. This prevents the simple replication of client requests to all servers, instead they need a translation to local settings. Dynamic conference behaviour like late joining or temporarily leaving participants is hard to accommodate in X, as the exact knowledge on current resources such as windows, pop-up menus, dialogue boxes, together with their actual attribute values is kept at the X server and is not available at the client-side. Simple strategies that archive all preceding messages for replay at the late-comer become prohibitive, because it would make join latency for the new participant a function of the previous duration of the conference.

2.2 Xy - an X Application Sharing Tool

Multicube selected for its experiments the X application sharing tool Xy [5]. The basic idea for Xy is to use an displayless X server as part of the distributed sharing component. This *Xy server* does not control I/O devices directly, but interacts with *Xy agents* local to the participant (see Figure 1).

Xy can fulfil the requirements on scalable X application sharing. It is fully transparent to X Clients and X servers: a client notices no difference when communicating with the Xy server to any other ordinary single-user X server. But also the X servers at the participant side need no modification, as the local Xy agent will emulate to them X client behaviour in a consistent way.The distribution of the sharing component opens up the possibility for exploiting multicast communication capabilities. An identical request message from an X client can be multicast via the X server to all Xy agents, and there the message is translated for the local X server. The local Xy agent also handles heterogeneity in system capabilities like pixel resolution, colour model, substitution of locally unavailable font families etc.

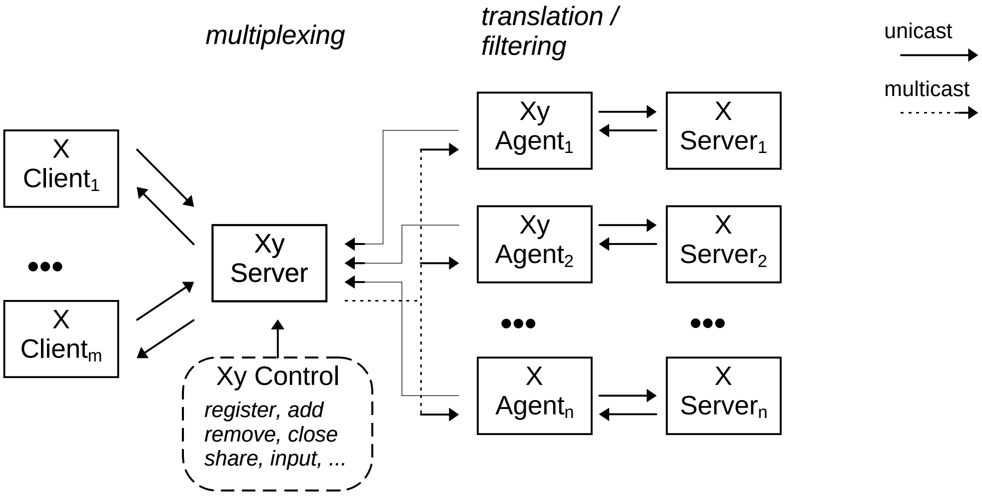

Fig. 1 Xy Architecture

Efficiency is guaranteed, since those requests that can already be responded by the Xy server need not be distributed over the network; and not all events need to be reported back; but some are already filtered out locally by the Xy agent. For instance only cursor movement from the current input token holder are transmitted, while corresponding events at other displays are discarded.

Accommodation of late-comers causes no problem to Xy, since all required information to instantiate the X server of a new participant is accessible in the Xy server. Xy even allows to close a conference temporarily and to resume it at a later stage in exactly the same state as before with no need for terminating running applications. This feature gives mobility to participants who can close a session on one machine and resume it on another.

2.3 Reliable Multicast Protocols

A large collection of proposals for reliable multicast protocols has been developed in recent years [6]. The service that they provide differs in adopted acknowledgement strategy, order preserving capabilities, supported atomicity and flow control.

Coordination between senders and receivers can be done explicitly by positive acknowledgement as it is done e.g. in TCP for the unicast case. Positive ACKs by all receivers for each packet sent, however, may easily become inefficient if not impossible for large groups. So trade-offs are needed to secure scalability with growth in group size. Negative acknowledgements are often better suited. Correct delivery is assumed and only lost or corrupted packets are reported back to the sender for retransmission.

In a multicast transmission, there can be several concurrent senders to a group, hence we can distinguish different degrees of sequence preservation for the receivers. Under source ordering, the packets are only ordered in sequence for each sending source, and these streams may be interleaved arbitrarily at each receiver; under total ordering, in contrast, each receiver gets all messages in exactly the same sequence. For the maintenance of a consistent state at all receivers, some applications require atomicity of packet transmission; i.e. a message is either received by all group members or by none at all.

Flow control prevents the sender from sending at a bandwidth higher than receivers can handle. Mechanisms for flow control may set the length of the sender window buffer, or be rate controlled, changing the frequency by which a sender can send out packets. The latter policy is especially appropriate for protocols using negative acknowledgements. For its experiments, Multicube has chosen three protocols that will be compared for supporting X application sharing sessions: MTP, RMP and SRM.

2.3.1 MTP - Multicast Transport Protocol

The Multicast Transport Protocol was defined in RFC 1301 [7]. It describes a protocol for reliable transport that utilizes the multicast capability of lower layer networking architectures. MTP is based on a token scheme and can provide atomicity and total ordering service. A group under MTP is known as a *web*. One member of the web is designated as *master* that controls the communication. It hands out send-tokens to *producers,* for sending data packets to the *consumers* in the web.

Transmission in MTP is rate based, the *heartbeat* is a time interval, during which only a fixed number of packets may be sent by a producer holding the send token. The *retention* value sets the number of heartbeats how long a sender must keep a packet buffered for possible retransmission. During that time, consumers can send NACKs in case they detect lost packets. The master sends out message acceptance records by which atomicity of packet reception in the group is controlled.

The variant of MTP that will be used in Multicube is the protocol named MTP-2 [8], which offers improvements in functionality and efficiency. MTP-2 adds recovery procedures for master failures and migration of the master side. Beside total ordering MTP-2 can also support weaker notions of message order. MTP-2 has been implemented in user mode and as a kernel driver for SunOS 5.

2.3.2 RMP - Reliable Multicast Protocol

Reliable Multicast Protocol (RMP) [9] provides a reliable multicast service on top of the unreliable multicast datagram of IP multicasting. It is based on a token ring technique and uses a combination of positive and negative acknowledgements (ACKs/NACKs) for error detection and delivery confirmation. With RMP the user can choose between different reliability guarantees which are selectable on a per packet basis. The QoS level ranges from unreliable, reliable, source ordered, totally ordered, and various levels of atomicity. The selected QoS level determines the ACK or NACK policy; an increase in the level of QoS has to be paid off with latency of packet delivery.

In RMP a token rotates in the group. The token holder is responsible to send out a positive ACK which contains information on the order and timestamps of received data packets so far, and then passes the token on. Based on the ACK message, members can detect lost packets and request retransmission by NACKs. A member only accepts the token, until it has received all previous messages. Thus, complete delivery of a packet can be detected after full rotation of the token.

RMP allows dynamic joining and leaving of group members. With special request packets, membership changes are signalled to the group members. RMP offers flow and congestion control based on a modified sliding window protocol similar to the algorithms used in TCP.

2.3.3 SRM (Scalable Reliable Multicast)

SRM (Scalable Reliable Multicast) [10] is not a fixed protocol but more of a framework in which reliable multicast services can be realized. The design of SRM is based on the idea of application level framing (ALF) and light-weight sessions. Under ALF functionality and flexibility is left to the application as much as possible.

Consequently, SRM assumes only a minimal delivery model for reliability, interpreted as eventual delivery of all data to all group members without enforcing any particular delivery order; the responsibility for stricter requirements is transferred to the particular application.

The SRM group model is closely modelled to the anonymous host group model of IP multicast allowing multiple senders and dynamic join/leave operations. Efficiency and scalability to very large groups is achieved by extending error recovery actions to all group members. Not only the sender alone, but any member that has correctly received a data packet can answer the retransmission request from a group member. By dynamically adjusting latency and response delay parameters, implosion of identical retransmission requests and repair packets is avoided, and thus a self-organising distributed error recovery system is installed.

The SRM framework has been prototyped in wb, a distributed whiteboard application, and has been extensively tested on a global scale with sessions ranging from a few to more than 1000 participants.

Given the characteristics of SRM, it qualifies as one of the prime candidates to be applied in Multicube. The adaptation of the SRM framework to X application sharing will be an important outcome of the experiments on X application sharing in Multicube.

3. Mapping of IP multicast to ATM point-to-multipoint connections

3.1 A Minimal Address Resolution Server

Convergence of IP and ATM service models [11] will require a full ‘IP service’ over ATM: multicast group management (IGMP [12, 13]), multicast flow distribution with receiver initiated joins, leaves, and QoS specifications. This convergence is supported by about 30 relevant IETF RFCs and drafts centered around Classical IP over ATM [14], vendor implementations, the MPOA work within the ATM Forum [15] and new features in UNI 4.0 draft following the IP service model. Table 1 gives an overview of the IP multicasting over ATM problem area.

A Minimal Address Resolution Server (mARS) was proposed in [16] for the MULTICUBE consortium partners to enable IP multicast experimentation over the European ATM pilot. Interconnection of the partners’ ATM islands could be achieved with the use of PVCs or with a specially designed type of call facilitating multipoint-to-multipoint connections in a public ATM network. Major mARS features are as follows:

- mARS is intended to resolve only IP Class D (multicast) addresses to NSAP addresses (in contrast to G. Armitage’s MARS that resolves both unicast and multicast addresses).
- mARS is to be used within the so-called multicast Logical IP Subnet (mLIS) while the regular unicast ARP is to be performed within Classical IP over ATM; like the IP

Table 1:

IP	ATM
Connectionless X	Connection-oriented
Only a single host needs to reply to Áll hosts’’query	Every host is to be queried directly and has to answer individually, endpoints registration is required
Receiver initiated join for the established MCT	Root initiated join only [UNI 3.1]
Total availability and knowledge of IP addressing	Need for ARP and ARP server
Various algorithms for the MCT construction	Use of multicast servers and/or meshes of VCs
Special means for QoS assurance (RSVP, etc.)	QoS assurance is a part of signalling

Table 2:

Name óf scheme	Address Resolution		Multicast	
	How	Int.	How	Int.
CLS	By Connectionless Server (CLS) in Switch Control Processor (SCP)	fa0	Cells duplication by switch hardware; point-to-multipoint connections	fa0
ClIP	By Classical IP over ATM ARP	qaa0	Cells duplication in host, mesh of point-to-point connections	qaa0
mARS	By mARS in mLIS	qaa2	Cells duplication by switch' hardware; point-to-multipoint connection	fa0

multicast which uses a separate address space within a connectionless environment, the mARS proposal suggests to separate LISs within a connection-oriented environment, where the above structuring of the address space is impossible.

- To avoid confusion with the existing Classical IP over ATM ARP server, mLIS could be configured as a private IP subnet within the MULTICUBE partner' premises; a single mLIS can serve a number of Classical LISs co-existing in a given ATM environment.
- The mARS implementation is to be based on the FORE *atmarp* source, extended to enable multiple NSAP addresses to be mapped to a single (Class D) IP address.
- mARS operation will be most beneficial for ATM networks with UNI 3.1/UNI 4.0 signalling, but also can be used together with the FORE IP.

Two issues should be distinguished: a/. *How is the address resolution for IP multicast over ATM done?* and b/. *How actually the multicast flow is distributed?* Table 2 shows these issues (with interfaces used - *Int*) for three schemes examined via simulation (section 3.2). Practically, nowadays the mARS scenario implies that each workstation, equipped with the ATM driver, should be configured to have *qaa0* interface for Classical IP, *fa0* - for FORE IP, and, say, *qaa2* - for mLIS. The *qaa0* and *qaa2* should know NSAP addresses for Classical IP over ATM ARP and mARS respectively. The *fa0* will use the FORE IP ARP. Table 2 shows two motivations for mARS: it decreases the SCP load by avoiding broadcast ARP and saves the bandwidth via utilisation of hardware cells duplication. Thus a workstation subscribes itself to a particular multicast group with mARS via *qaa2* interface and then receives the multicast flow via *fa0* (or any of *qaXX, faX* when UNI 3.1/UNI 4.0 will become available). This is possible because the ForeRunner SBA-200 ATM SBus adapter inside the workstation serves also as a router between different connected LISs.

3.2 Simulation results

The above schemes were simulated within the discrete event domain of the Ptolemy modelling tool [17]. All experiments assume that a local ATM network has a FORE ATM switch (say, the ForeRunner ASX-200) supporting the above mentioned assumptions and that the SCP's processing capacity (*C*) is shared among the following types of load: unicast connection establishment and maintenance, multicast connections establishment and maintenance, connectionless service (with lower priority). According to [18] $C = 100$ calls/s. The load is produced by FORE IP for a broadcast ARP. Independent parameters for each run of the simulation are: Unicast Call Load (*UCL*) and Multicast Call Load (*MCL*). Address Resolution Load in terms of Connectionless Load (for FORE IP ARP) or Classical ARP Load (for Classical IP over ATM) is defined by *UCL* and *MCL*. Independent parameters are measured in terms of percentage of *C*. Obviously, at any time $UCL + MCL$ should not exceed the total *C*.

Basically, two different cases should be considered: Classical IP ARP implemented at switch and Classical IP ARP implemented at other location (endpoint, router). In the latter case the SCP has the increase of its unicast load due to the necessity to establish on demand

and maintain PVCs from endpoints to the ARP server. When the Classical IP ARP server implementation occurs within the switch it also decreases the switch productivity. This paper neglects the difference between these two cases. We also assume that all endpoints are directly connected to the switch, that Classical IP over ATM follows the RFC 1577 and its timing defaults (for endpoint's ARP cache ageing for example).

Simulation results are summarised in Tables 3 to 6. Each simulation ran for 1000 simulated seconds which corresponds to simulation of 10000 to 80000 calls established for each of the considered variants. In all of these experiments the switching fabric performance was not considered - only SCP performance was taken into account in terms of minimal connection establishment delay (*SCPmin = 1/C*). The major output of the model was called a Multicast Connection Latency which consists of ARP delay and connection establishment delay (*CEdelay*). *CEdelay* is a function of *SCPmin (1 + UCL + MCL)*.

Actual instance of *CEdelay* (to model connection establishment to a host or to an ARP server) was considered randomly chosen out of the interval [*0.8·CEdelay, 3.6·CEdelay*]. These coefficients were measured in the GMD FOKUS IP over ATM environment; details on simulation tuning and organisation could be found in [20].

According to [18] the CLS service is considered to have a lower priority for an SCP in comparison with connections establishment. This situation is reflected in the model as well as hardware multicast (*CLS*), mesh of point-to-multipoint connections (*ClIP*), and mLIS usage only for multicast ARP queries (*mARS*).

Table 3:

Multicast connection latency [sec] as a function of unicast call load (UCL)

UCL =	*0.1*	*0.2*	*0.4*	*0.6*	*0.8*
mARS	0.104	0.099	0.168	0.264	0.380
ClIP	0.262	0.287	0.329	0.377	0.415
CLS	0.886	1.298	2.630	3.365	5.90

Table 4:

Multicast connection latency [sec] as a function of multicast call load (MCL)

MCL =	*0.1*	*0.2*	*0.4*	*0.6*	*0.8*
mARS	0.104	0.050	0.044	0.052	0.059
ClIP	0.262	0.291	0.328	0.372	0.415
CLS	0.886	0.708	0.697	0.708	0.739

Table 5:

Multicast connection latency [sec] as a function of a group size (M)

M =	*1*	*5*	*8*	*10*	*11*	*16*
mARS	0.066	0.067	0.067	0.067	0.066	0.066
ClIP	0.066	0.164	0.268	0.328	0.361	0.521
CLS	0.993	1.062	1.062	1.096	1.034	1.112

Table 6:

Multicast connection latency [sec] as a function of # of active ports (P) in a switch

P =	*10*	*12*	*16*	*24*
mARS	0.066	0.066	0.067	0.066
ClIP	0.325	0.329	0.325	0.329
CLS	0.661	0.798	1.049	1.635

Tables 1 and 2 assume that the load which is not varying in the experiment is UCL/MCL = 0.1 (i.e. 10% of the maximal switch load C); the number of multicast group members M = 10; the number of active switch ports P = 16. For Tables 3 and 4 UCL = MCL = 0.25 (25% of C each, which constitutes 50% of C in total).

Some results:

1. CLS case has the largest latency (due to a broadcast implementation and, mainly, due to a lower priority); it depends linearly on the number of active ports (P); and non-linearly on the unicast load. The CLS case is generally less effective than ClIP or mARS but,

nevertheless, it scales well in terms of MCL growth and the group size growth. The CLS is the only scheme where ARP and Call Establishment are not separated.

2. ClIP case with a mesh of point-to-point connections has the 2nd largest latency (however, 2 - 10 times less than that of CLS) which depends linearly on the number of receivers (each newcomer to the group requires both the ARP service and a separate point-to-point connection); it's the only scheme which depends on the MCL growth.

3. Minimal address resolution server shows the smallest latency and scales well in terms of all parameters: UCL, MCL, M, and P. This is due to combination of a separate mLIS and the hardware multicast.

4. Implementation Issues

The following implementation alternatives were considered within the project: FORE SPANS signalling, FORE UNI 3.0 signalling, Vendor Independent Network Control Entity (VINCE) by ARPA and NRL, and Q.port (Portable ATM Signalling Software) by Bellcore.

It's important to mention here that the multicast capable MARS [2] by G. Armitage is available as public domain software in a source code except the Q.port. Almost all of the project participants are using FORE equipment on their premises, therefore dependence on a different signalling was considered as a disadvantage. VINCE, available as public domain software in a source code, was considered as an alternative, especially for the reason of its vendor independence. However, the lack of experience with VINCE and its non-standard NSAP format has turned the project' choice in favour of decisions supported by FORE - UNI 3.0 signalling and SPANS signalling. Availability of FORE source code for needed utilities was also a factor.

Unfortunately, it turned out that current release of *ForeThought 3.0.2* software does not support the add_party primitive which is needed for a point-to-multipoint call. It will be supported with other UNI 3.1 features only in a next release. In parallel to waiting for this new release it was decided to start implementation of mARS.

So far, each ATM LAN will be configured to have at least two logically separated LISs - one for Classical IP over ATM with unicast address resolution, and another with multicast address resolution. To avoid a full mesh of point-to-point connections over ClIP it is recommended to use FORE IP with hardware multicast capability. ARP overhead shown by simulation is considered less harmful than completely inefficient bandwidth utilisation in case of mesh. This solution was implemented within the networking infrastructure of GMD FOKUS and demonstrated with the use of integrated multimedia conferencing tool `isc [19]`.

The lessons learned so far on implementation are as follows:

- Multicast applications can run successfully within the FORE IP logical subnetworks in parallel with the ClIP LIS, co-existing for the same physical structure.
- FORE ATM adapter software serves as a router (inside multihomed hosts) between different LISs (including Ethernet, several Classical and FORE IP LISs).
- FORE IP uses hardware cells duplication which makes it effective for the distribution of the multicast flow but not for the broadcast ARP, because the CLS becomes a bottleneck in case of high call load.
- based on simulation results, due to the separation of LISs, the mARS alternative could be considered as the best available choice in terms of efficiency, scalability, and implementation efforts.

References

1. MULTICUBE - *Efficient MULTIpoint to MULTIpoint broadband switched network services for distributed MULTImedia applications*, ACTS project AC012, home page at http://www.cselt.stet.it/sonah/MULTICUBE/index.html
2. Armitage, G. *Support for Multicast over UNI 3.0/3.1 based ATM Networks*, Internet draft <draft-ietf-ipatm-ipmc-12>, February, 1996, 83 pp.
3. Hussein Abdel Wahab, Kevin Jeffay: *Issues, Problems and Solutions in Sharing X Clients on Multiple Displays*. - Technical Report (TR92-042). UNC-Chapel Hill, December 1992; ftp://ftp.cs.odu.edu/pub/wahab/XTV/XTV.related.papers/XIssues.ps
4. G. Chung, K. Jeffay, H. Abdel-Wahab, *Dynamic Participation in Computer-based Conferencing System*. - Journal of Computer Communications, pp. 7-16, Vol. 17, No. 1, January 1994; ftp://ftp.cs.odu.edu/pub/wahab/XTV/ XTV.related.papers/ LateJoin.ps
5. Bormann, C, Ou, J, Gehrcke, H-C, Kerschat, T, and Seifert, N, MTP-2: *Towards Achieving the S.E.R.O. Properties for Multicast Transport* - In: ICCCN 94, San Fransico, Sept. 1994, http://www.prz.tu-berlin/~torsten/papers/sero.ps
6. Knight, Jon P.; *Multicast Transport Protocols.* - High Performance Networks and Distributed Systems, Archive URL http://hill.lut.ac.uk/DS-Archive/MTP.html
7. S. Armstrong, A. Freier, K. Marzullo *Multicast Transport Protocol*. IETF RFC 1301, Feb. 1992.
8. C. Bormann, G. Hoffmann: *Xmc and Xy - Scalable Window Sharing and Mobility*. - 8th Annual X Technical Conference, January 1994; http://www.prz.tu-berlin.de/docs/ Publications/TUB-PRZ-W-1187.ps
9. Whetten, B., Kaplan, S. and Montgomery, T., *A High Performance Totally Ordered Multicast Protocol* - In: Dagstuhl Distributed Systems Workshop, Sept. 1994, ftp://research.ivv.nasa.gov/pub/doc/RMP/RMP_dagstuhl.ps
10. Floyd, S., Jacobson, V., Liu, C., McCanne, S., and Zhang, L.: *A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing*, in ACM SIGCOMM 95; file://ftp.ee.lbl.gov/papers/srm.ps.Z
11. Crowcroft, J., Wang, Zh., Smith, A., Adams, J. *A Rough Comparison of the IETF and ATM Service Models*. - In IEEE Network, Nov./Dec. 1995 Vol. 9, No 6, pp. 12-16
12. Deering, S., *Host Extensions for IP Multicasting*, IETF, RFC 1112, Aug., 1989
13. Fenner, W. *Internet Group Management Protocol*, Version 2, Internet draft, <draft-ietf-idmr-igmp-v2-01>, September, 1995
14. Laubach, M., *Classical IP and ARP over ATM*, IETF, RFC 1577, January 1994, 17 p.
15. *Baseline Text for MPOA*, ATM Forum document [ATMF/95-0824r1]
16. Salgarelli, L., Schulzrinne, H., Roth, R., Smirnov, M. *Definition of services provided by the protocol architecture*, MULTICUBE/I/W23/GMD/ 001.02 [I.2301 Release2], 39 pp.
17. Buck, J., Goei, E., Huang, W.-J., Kamas, A., Lee, E. *PTOLEMY: Users Manual*, U.C.Berkeley, Department of EECS, Apr., 1994.
18. *ForeRunner ATM Switch User's Manual*. - MANU 0065 - Rev.A - BETA2 - January, 1996, FORE Systems, Inc.,
19. Schulzrinne, H., *Guide to isc 1.1* (Integrated Session Controller), http://www.fokus.gmd.de/step/isc/
20. Smirnov, M. *ARP Perfomance Study* , - Technical Report, GMD FOKUS, 1996.