

A Streaming Architecture for Next Generation Internet

Ashutosh Dutta

Telcordia Technologies, 445 South Street, Morristown, NJ 07960

Henning Schulzrinne

Computer Science Department, Columbia University, New York, NY 10027

Abstract—

With the influx of multi-media streaming content over the Internet, building a streaming infrastructure that would support flexible Next Generation Internet application in a scalable way on heterogenous access technology is most desirable. This paper presents a streaming architecture which provides some innovative methods and technique that would help build an IP based radio/TV network. This would provide tools for E-commerce and would benefit the broadcasting stations, local affiliates, ISPs, and the end users. Additionally this paper highlights the mechanism behind some of the components associated with this architecture such as local content, global content, program management with local control, payment model, security, advertisement insertion, seamless mobility within a domain.

I. INTRODUCTION

Streaming real-time multimedia content over the Internet is gaining momentum in the communications, entertainment, music and interactive game industries. Streaming application include IP telephony, broadcasting multimedia content and multi-party conferences, collaborations and multiplayer games. These applications are far more demanding in terms of bandwidth, latency and reliability than traditional data applications and are thus ideal drivers for the Next Generation Internet. In addition, they would require multicast support. The Internet Engineering Task Force (IETF) has developed a suite of protocols including RTP, RTSP, SAP, SDP and SIP to deliver and control real-time multimedia traffic. Real-time streaming traffic is typically carried over RTP/UDP/IP. SAP and SDP announce and describe, respectively, multimedia sessions.

Many of the existing streaming systems (e.g., Real Networks or Windows Media) do not scale to large audiences, particularly for high bit rates. They also lack in providing user flexibility and are restricted to either

conferencing or broadcast modes. The proposed system is scalable and provides flexible streaming services for the NGI. In addition, we plan to demonstrate how such services can move beyond the traditional PC and workstation to networked devices that are as easy to use as today's analog radio and TV, yet far more flexible. This could serve as a potential architecture for future Mobile E-commerce (M-commerce).

The rest of the paper is organized as follows, Section 2 briefly describes the architecture, section 3 details the design and functionalities of different components associated with MarconiNet. Section 4 deals with the mobility for streaming multimedia, section 5 points out several options of how this can be supported in the core network in the absence of multicast support. Section 6 provides some details of the testbed, section 7 talks about the related work in this field. Section 8 concludes the paper.

II. ARCHITECTURE

Integrated Streaming Network describes many of the details of an architecture for an IP based next-generation network for streaming audio and video [1]. It uses Internet Protocol suite to provide robust communication over heterogeneous access technologies such as cable modem, ADSL, PPP, IEEE 802.11. Distributed server based approach of MarconiNet offers a great amount of scalability, flexibility of radio services, better reach and quality of service for the audio/video stream carried over IP. This streaming network will have various throughput, delay, error rates and include wired and wireless links.

This novel distributed platform offers several important features. It could provide flexible radio services to the local Internet Radio/TV clients which otherwise cannot be supported by traditional AM/FM receivers or local TV sets. It offers the flexibility to the clients to tune to any radio/TV station in the world. It offers capability of hierarchical searching in terms of categories, means

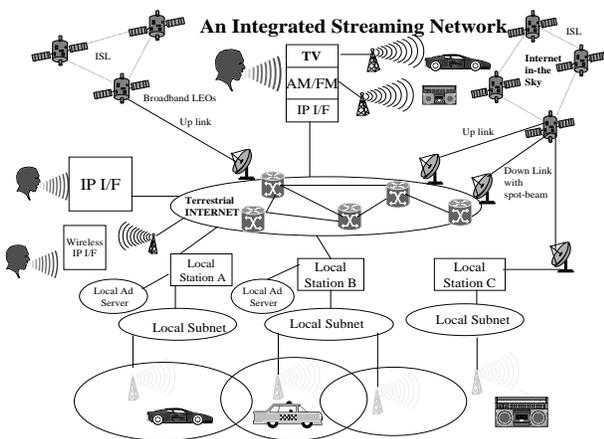


Fig. 1. Integrated Streaming Network

of insertion of the local advertisement during the commercial break. This will meet the challenges of bringing quality audio/video broadcast to the people in remote site, and to the wireless clients who are mobile. By introducing the Radio Antenna Servers for local domains which act as local stations, it can provide a scalable distribution of the streaming content so that many people can listen to a particular radio station in the world without possible degradation of audio quality and gives the ability to the local listeners in a single domain to switch between the local program and global program. It also provides the ability to the local listeners to listen to the local advertisement during the commercial break while tuned to any global program or listen to the some segment of the global program while tuned to the local program. Another remarkable feature of this architecture (often known as MarconiNet) is that it extends the capability to anybody in the world to be a potential broadcaster. The pricing model associated with different entities of the architecture would make it more attractive in terms of commercial usefulness.

This architecture is transport independent, works over wired and wireless links, and takes care of the mobility, thus providing a continuity to the listener of a particular radio station as it moves from one cell to another, one subnet to another, one domain to another.

There are four main functional components in this main architecture, namely Radio Station Client (RSC), Radio Antenna Server (RAS) or local station, Advertisement/media server and Internet Multimedia Client (IMC). Alternatively media server can be just a media database attached to the local station. The proposed MarconiNet broadcasting architecture consists of two tiered (hierarchically-scoped) IP multicast sessions.

This model assumes that there is multicast connectivity throughout the network, but if there is a lack of multicast connectivity at certain parts of the network, then there are some possible application layer solutions [5] that can be deployed which are described later on in the section. Satellites can also form the core of the network thus providing natural multicast capabilities within the core of the network. Figure 2 shows the functional components.

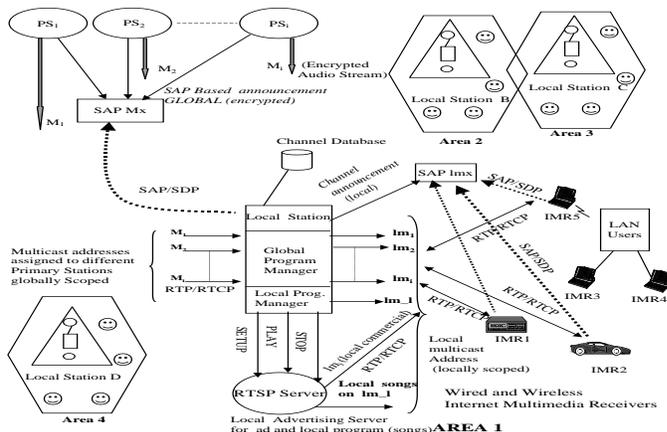


Fig. 2. Functional Components

The higher level of the two (global multicast) exists between the broadcasting radio stations (RSCs) and the Radio Antenna Servers (RASs) which are also called local stations. At the lower level of the hierarchy, a local multicast session is created for each broadcasting station between the server and the listening clients (IMC) which can be privately scoped.

III. PROTOCOL DESIGN AND FUNCTIONALITIES

This architecture uses distributed nature of Client-server architecture, and is based on the Internet protocol suite. The protocols which are used here are SAP, SDP, RTSP, RTP, TCP, UDP, IP and IP Multicast. Each source is assumed to be sending RTP [12] traffic along with the control traffic RTCP. A protocol flow is described below describing the interaction between different entities of the proposed system.

A. Channel Announcement

Main radio/TV stations all over the world like to broadcast their programs potentially to the global audience. Any main radio station RSC_i keeps sending its programs live on a unique multicast address M_i globally scoped and encrypted using RTP/UDP (where M_i

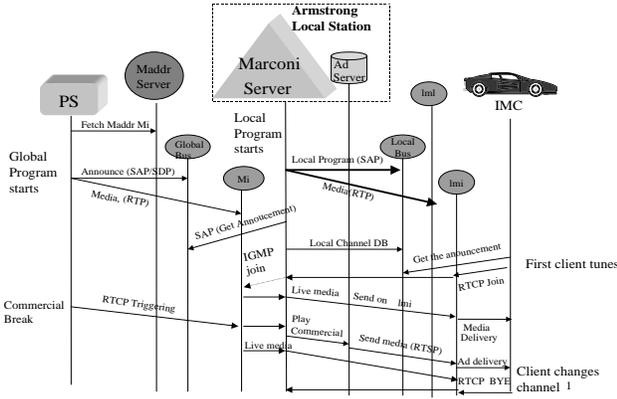


Fig. 3. Protocol Flow for MarconiNet

is different for different station). These radio stations send their session announcement using subset of SDP parameters to a global multicast address M_x which is also encrypted. This common global multicast address contains a list of the programs aired by the main radio stations (RSCs) all over the world. Modifying SDP or using a variant of SDP gives us more details about the programs being broadcast. To that effect we have designed a java based interface called JS DR which is more customized and provides a better hierarchical searching functionality compared to traditional SDR.

B. Channel Management

Each RAS gets a global encryption key which it uses to listen to the global common multicast address M_x to get the listing of the channels, and its contents. It decides to broadcast a part of the list to the local domain, and hence creates a local announcement database. The subset of channel descriptions announced by each radio station provides sufficient data for building a local channel database. This local announcement database contains the list of the supported channels, each with their appropriate attributes such as name of the program, duration, type of content, place of origin etc. This database would be a subset of the list that they get from the global multicast address. Local station sends this program-index to a locally scoped common multicast address lm_x for announcement using SAP [8]. The announcement on lm_x is not encrypted since the local stations would like all its clients to see what is being relayed by it. RAS also maintains a pair of multicast addresses for each channel. It keeps the mapping of the globally scoped multicast address on which the radio station sends its program and the locally scoped multicast address where it gets

relayed. RAS receives the audio stream on the global multicast address M_i and redirects it onto the local multicast address lm_i for the IRCs. Local programs are sent on a specific locally scoped multicast address lm_i .

The Local Stations (RAS) use the global encryption key to be able to point at the global common multicast address to find out as to what is playing globally. These local stations maintain a database of which programs they are planning to relay to the local clients. Each RAS announces the contents of this local announcement database on a common multicast address in the local domain for the IMCs to choose.

C. Channel Tuning

Internet Multimedia clients listen to the locally-scoped common multicast address lm_x to be aware of what is being available using any tool based on SAP and SDP. According to the SAP specification, the antenna server will update the announcement information every several minutes or so. Just pointing to a particular channel would provide the details of the program which is available.

D. Global Program Management

Local station manager receives the stream on M_i and sends it out on locally scoped multicast address lm_i for the client. This could be encrypted or not encrypted depending upon what kind of payment model is being used. The client keeps on sending RTCP packets to the management server as long as it receives the audio streams on a particular address. These reports can be used for many different statistical purposes like billing, audio quality feedback and also membership information for a particular multicast group. When the management server receives the audio/video stream from a radio station client on a specified multicast address using RTP/UDP, it also exchanges RTCP signals with the radio station periodically.

E. Local Program Management

Managing the local channel is bit different. After looking at the program preview the client makes a similar request to the local channel. For each local station there is only one locally scoped multicast address lm_i where the audio/video streams from the local programs are sent. Local Program Manager maintains a schedule of what has to be played and announces it to the local common multicast address lm_x using SAP but sends the audio streams on the locally scoped multicast address

lm_i . At times when it would like to play some songs or any other ads it would request the local RTSP [17] server to play onto this address using SETUP and PLAY command. During some part of local programs the local station may want to relay some radio station's news broadcast. Thus for that particular period the local manager decides to join this specific global multicast group, when it receives the RTP packets from the main radio station and generates the RTP/RTCP packets for the local listeners on the local multicast address lm_i . At times this local program Manager can store the global news and play it at a later time via the RTSP server.

F. Local Advertisement Insertion

It is assumed that each global radio station knows the starting time and duration for commercial break ahead of time or has the control over the time for break. Along with RTP packets these stations keep on sending RTCP packets to the multicast address where the stations are listening to receive the stream. Through the RTCP report the radio station triggers the signal to the local stations for commercial break. On receiving the signal for commercial break, the management server at the local station requests the local RTSP server to start playing the local advertisement from a storage media to a specific locally scoped multicast address lm_i corresponding to that station. It uses a set of RTSP commands like SETUP, PLAY and STOP. During this time, management server stops forwarding the RTP stream from M_i to lm_i in the local domain. The local advertisement runs for a specific time which the management server is made aware of from the RTCP reports. Management server sends a STOP signal to the RTSP server so that it stops playing on that particular multicast address, and the management server resumes redirecting the audio streams from the global multicast address to the locally scoped multicast address. These RTSP control signals are sent over TCP connection. Since the commercial break time for several radio stations may overlap it could so happen that RTSP server could be playing several different local advertisements on different locally scoped multicast addresses, each one meant for different radio station. RTCP packet is of type APP (application defined) conforming to RTP/RTCP RFC. Since RTCP packets are being transmitted over UDP, there might be a possibility of packet loss or incorrect order. To address this problems, RTCP packets are retransmitted (e.g. one packet 4 seconds prior to the commercial, next one - 3 seconds, next one - 2 seconds, etc. with corresponding value in

the field which indicates the time remaining till the start of the commercial). In addition, to allow Marconi to distinguish between re-transmissions and new commercials, RTCP packets have a sequence number. All retransmissions have the same sequence number. Each new commercial has a sequence number one greater than the previous one.

G. Channel Monitor

For each local channel being diverted, additional RTCP signaling channel is created with different port. Each listener periodically sends RTCP SDES packets to notify the Local Station (RAS) "who's listening to what". RAS maps each listener to the desired channel, which in turn increases the number of listeners for that particular channel. The listener-to-channel mapping is destroyed via RTCP BYE packets or via RTCP time-out feature. This also decreases the number of listeners for the associated channel. Channel monitor provides a snapshot view of how many clients are tuned to a particular channel.

H. Payment Model

This architecture can support many kinds of payment models. Local stations (Radio Antenna Server) collect the fees from the local commercial companies so as to send the advertisement during the commercial break while relaying the global stations. Besides relaying the programs from the common global stations, the local stations also relay some pay-per-listen programs, in which case it pays to the global station depending upon how many listeners are listening to a particular program. This can be determined from the RTCP reports generated from the Internet Clients. Every local station also broadcasts its local program to the local clients with segments of news or some other premium programs relayed from the global stations. Two kinds of payment model can be designed, there may be users who would like to listen to the program without being interrupted by the advertisement and some would not really mind the local ads.

I. Logging Mechanism

The logging mechanism has a very close ties with E-commerce application, and is a good way of generating revenue. The purpose of the logging mechanism is to facilitate for advertisers the process of determining when and on which channels to place their commercials in order to maximize returns on investment. The higher level

of advertising efficiency will drive more advertisers toward advertising on Marconi. This, in turn, will allow the Marconi to pursue an increasing share of Internet radio advertising market by charging different prices for different segments of airtime, based on demand.

This reporting mechanism enables Marconi (i.e. RAS) server operator(s) to track user listening trends and can provide to potential/existing advertisers detailed reports with users' listening statistics. This concept relies heavily on the RTCP-triggering model that is supported by Marconi. Even though the primary purpose of RTCP is to notify Marconi when users join a local multicast group to start listening on an audio channel, RTCP is also ideally suited for allowing Marconi to gather user-specific and channel-specific listening information.

J. Security

The proposed MarconiNet architecture presently offers four levels of encryption overall. Global announcement encryption, global multicast stream encryption, local audit encryption and user authentication. By using global announcement encryption we can separate global announcements from the local announcements. The local IMCs should not be able to get access to the global announcements and only be able to view the local announcements. By using a global encryption key during the announcement by the global radio stations, it does not allow the local Internet Multimedia clients to find out about the global channels and thus gives the control over to the radio stations to announce only a subset of these channels to the local clients. Security model for global multicast stream should effectively prevent IMCs, as well as the non-paid RASs, from receiving the broadcast content. Thus each radio station (RSC) must maintain a secret key and encrypt all outgoing content so that only ciphertext stream is transmitted. The basic strategy is to generate a symmetric encryption key at the station and securely distribute this key to a particular RAS upon its payment.

Global Multicast stream encryption can be extended to local section as a second level hierarchy. Some of the pay-per-listen programs are announced to the local multi-cast addresses in any domain using encryption key, so that proper charging methodology can be in place for the IRCs. Some sort of encryption can be applied for the audit data of RAS, so as to preserve the sensitive information such as the secret keys (of RSCs, and pay-per-listen channels), user accounts, and payment data. Advertising companies can be authenticated so that un-

authorized companies cannot try to play with the local Advertisement Insertion system. The security is taken care of as follows. Each local station generates its own PubKey / PrivKey pair. Each PS generates a SEK and begins transmitting encrypted audio content. This SEK needs to be distributed to the participating local stations in a secure way so that other local stations that did not pay cannot obtain this key. Public key technology is employed for this purpose. The local station submits its public key to the primary station alongwith its payment or via CAs. Local station receives an Integer ID from the station which is later used to index the SEK distribution list. The PS collects the public keys from the local stations and add these to its SEK distribution list upon their payment.

IV. MOBILITY FOR STREAMING MULTIMEDIA

Continuity of streaming multimedia or the client being able to listen to the same radio station or watch the same Radio/TV station, when it is moving from one cell to another in a wireless environment is of prime importance for a scalable streaming system. Some of issues have been discussed in [2]. It would be desirable to have minimum loss of packets when the client is moving. There needs to be a co-ordination between the adjacent servers, so as to trigger the stream as soon as the client moves into a new cell or trigger it even before the client moves in by some proactive mechanism. Figure 4 illustrates such a scenario. This would take care of several scenarios such as moving between the cells belonging to the same subnet, and moving between the subnets within a domain. In this particular case join and leave messages are taking place at the application layer unlike in traditional cases where it takes place in networking layer using IGMP. Currently we are investigating if application layer triggering is any way better than IGMP based triggering for quicker surfing.

V. NON-MULTICAST ENABLED NETWORK

Our previous model in section 2 assumes that we have multicast enabled network from end-to-end. In practice this has not been the case yet. Although it is easier to implement multicast in the Intranet or within an autonomous system, multicast support between the autonomous systems is still not easily available. Our model discussed above works perfectly when all the routers in the Internet will be multicast enabled. In order to extend the above functionality to a network where multicast is not supported there has to be some

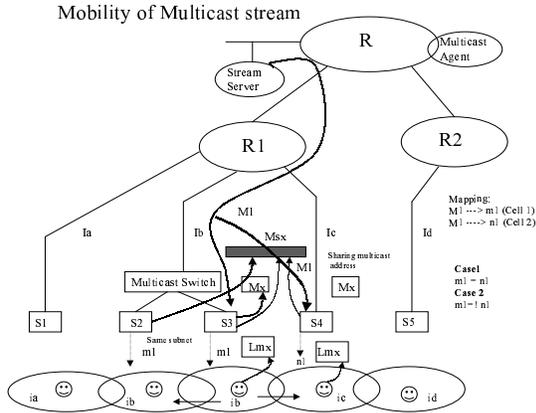


Fig. 4. Mobility Scenario in MarconiNet

changes in the design approach. There are several solutions based on user level or network level application available [5] to connect these multicast enabled islands. These multicast applications can also be distributed via broadband LEO satellite systems as part of several spot beams.

VI. MARCONINET TESTBED

Current MarconiNet test bed consists of several subnets connected to the enterprise router. Most of the systems which are used in the test bed are either Sparc systems or Intel Pentium based PCs running Solaris and Linux respectively. Each subnet is considered to be a LAN. Here we can compare a LAN with one autonomous system which could be an enterprise or campus.

Each LAN has one server which operates as a Radio/TV Station and broadcasts audio/video to a specific multicast address. There is at least one system acting as local station which takes care of relaying the programs to the clients which are in the same LAN and takes care of the management functions. Internet Multimedia Clients in the current test bed include a variety of systems over a heterogeneous links such as wired workstation, wireless laptops, and we are extending to include dial-up users over PPP link. Since these clients have heterogeneous connectivity speed, the quality of audio stream that each type of clients receive may be different. This is a further area of study. There is a radio station client in each subnet which broadcasts audio stream on a particular multicast address. Since at the moment this router does not support multicast, so we have a server running multicast routing daemon in each subnet which establishes a multicast tunneling between

the subnets, and thus enables the multicast communication between the subnets. Alternatively we could put a server with multiple interfaces which will act as a router thus obviating the need for a multicast tunneling. But this shows a perfect example where how this architecture can be realized in the absence of when there is no multicast support in the routers in the network. Current implementation supports relaying the audio stream from the global stations while inserting the local advertisement during the commercial break. Relaying some part of the news program during the local program is being implemented. Current implementation also supports customized SDR, and publicly available tools like rat [14], and vic [15] have been incorporated so that program can be played from a file or via these tools, like when an individual would like to speak, can use rat kind of tool, otherwise it can play the song from a database. Similarly a video can be played using vic tool.

VII. RELATED WORK

As demand for live and on-demand streaming (both audio and video) over the Internet increases, there are many ongoing projects trying to find out the best way to provide quality audio and video stream to the end user over the internet without possible congestion in the network and without possible overload on the server. Most of the products and streaming framework which are available to day are mostly suited for Intranet application. IP/TV from Cisco, Netshow from Microsoft are some of these products which use IP Multicast and provide good quality of service within an enterprise over the Intranet. Real Audio and Real Video from Real Networks provide the most popular software for streaming audio and video over the Internet.

Currently there are many organizations offering live radio services to the end users over the Internet. Some of these provide audio streaming over unicast and some provide using Mbone. Broadcast.com, World Radio Network, and National Public Radio are some of the organizations which offer links to the radio stations which are otherwise local. But scalability is a big problem in these cases, since there can be only a limited number of simultaneous listeners without possible loss of good quality, besides, these do not offer any flexible services to the users. None of the related works offers flexible radio services to the users such as listening to local advertisement during a commercial break while tuning to a global station, a choice of tuning to local or global channel, pay-per-listen program, payment model

between the global station and local affiliates, delegating control functions on the local stations.

Most recently Akamai and Network Appliance have developed a new protocol called ICAP (Internet Content Adaptation Protocol) which enables communication between edge content devices (e.g., web caches and Internet content delivery servers) and application servers that modify content. But this does not support multicast yet, and has not included mobility of the end clients into its consideration.

Also iBEAM's infrastructure and its product Active-cast are very similar to our proposed framework for streaming application over the Internet, but it lacks in taking care of user mobility, uses the GEO satellites for content distribution, does not provide flexible methods of advertisement insertion, mixing of local and global content.

Thus our proposed architecture offers many of these missing flexible components in addition to providing inherent scalability and control over local management server.

VIII. CONCLUSION AND OPEN ISSUES

This paper presented a novel architecture for scalable and flexible multimedia stream distribution over the Internet using the existing Internet based protocols and underlying multicast technology. Distributed nature of server placement, hierarchical multicast scoping approach, local management control on the local servers, novel streaming methods, support for mobility would make it possible to come up with this flexible and scalable next generation streaming network viable for E-commerce. This architecture when fully implemented will offer a variety of services which are not otherwise available in the traditional streaming systems.

REFERENCES

- [1] A. Dutta, H. Schulzrinne and Yechiam Yemini, "MarconiNet - An Architecture for Internet Radio and TV Networks" in *Proc. of NOSSDAV 1999*
- [2] M. Flament et. al, "An approach to 4th Generation Wireless Infrastructures- Scenarios and Key Research Issues". VTC 99, May 1999
- [3] H. Schulzrinne, "Re-engineering the telephone system," in *Proc. of IEEE Singapore International Conference on Networks (SICON)*, (Singapore), Apr. 1997.
- [4] H. Schulzrinne and J. Rosenberg, "Internet telephony: Architecture and protocols - an IETF perspective," *Computer Networks and ISDN Systems*, vol. 31, pp. 237-255, Feb. 1999.
- [5] R. Finlayson, "The UDP multicast tunneling protocol," Internet Draft, Internet Engineering Task Force, Feb. 1998. Work in progress.
- [6] J. Brassil, S. Garg, and H. Schulzrinne, "Program insertion in real-time IP multicasts," *ACM Computer Communication Review*, vol. 29, pp. -, ? 1999.
- [7] M. Handley and V. Jacobson, "SDP: session description protocol," Request for Comments (Proposed Standard) 2327, Internet Engineering Task Force, Apr. 1998.
- [8] M. Handley, "SAP: Session announcement protocol," Internet Draft, Internet Engineering Task Force, Nov. 1996. Work in progress.
- [9] A. Ballardie, "Scalable multicast key distribution," Request for Comments (Experimental) 1949, Internet Engineering Task Force, May 1996.
- [10] H. Harney and C. Muckenhirn, "Group key management protocol (GKMP) architecture," Request for Comments (Experimental) 2094, Internet Engineering Task Force, July 1997.
- [11] S. Mitra, "Iolus: A framework for scalable secure multicasting," *ACM Computer Communication Review*, vol. 27, pp. 277-288, Oct. 1997. ACM SIGCOMM'97, Sept. 1997.
- [12] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: a transport protocol for real-time applications," Request for Comments (Proposed Standard) 1889, Internet Engineering Task Force, Jan. 1996.
- [13] J. Rosenberg and H. Schulzrinne, "Timer reconsideration for enhanced RTP scalability," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, (San Francisco, California), March/April 1998.
- [14] "UCL RAT <http://www-mice.cs.ucl.ac.uk/multimedia/software/rat>"
- [15] "LBL video conferencing tool <http://www-nrg.ee.lbl.gov/vic>"
- [16] T. Friedman and D. Towsley, "Multicast session membership size estimation," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, (New York), Mar. 1999.
- [17] H. Schulzrinne, A. Rao, and R. Lanphier, "Real time streaming protocol (RTSP)," Request for Comments (Proposed Standard) 2326, Internet Engineering Task Force, Apr. 1998.
- [18] W. Fenner, "Internet group management protocol, version 2," Request for Comments (Proposed Standard) 2236, Internet Engineering Task Force, Nov. 1997.